

5ESS[®] Switch AUDITS MANUAL

Document: 235-600-400
Issue Date: October 1999
Issue Number: 8.01B

Legal Notice

Copyright ©1999 Lucent Technologies, Inc. All Rights Reserved.

This electronic information product (IP) is protected by the copyright and trade secret laws of the United States and other countries. The complete document may not be reproduced, distributed, or altered in any fashion. Selected sections may be copied or printed with the utilities provided by the viewer software as set forth in the contract between the copyright owner and the licensee to facilitate use by the licensee, but further distribution of the data is prohibited.

For permission to reproduce or distribute, call:

1-800-645-6759 (From inside the continental United States)

1-317-322-6847 (From outside the continental United States).

Notice

Every effort was made to ensure that the information in this IP was complete and accurate at the time of publication. However, information is subject to change.

This IP describes certain hardware, software, features, and capabilities of Lucent Technologies products. This IP is for information purposes; therefore, caution is advised that this IP may differ from any configuration currently installed.

Lucent Technologies is the successor to the business and assets of AT&T Network Systems business unit.

Mandatory Customer Information

Interference Information: Part 15 of FCC Rules - Refer to the 5ESS[®] Switch Product Specification IP.

Trademarks

5ESS is a registered trademark of Lucent Technologies.

ANSI is a registered trademark of American National Standards Institute.

AUTOPLEX is a registered trademark of Lucent Technologies.

COMMON LANGUAGE is a registered trademark and CLEI, CLLI, CLCI, and CLFI are trademarks of Bell Communications Research, Inc.

Motorola is a registered trademark of Motorola, Inc.

SLC is a registered trademark of Lucent Technologies.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Warranty

Warranty information applicable to the 5ESS[®] switch may be obtained from the Lucent Technologies Account Management organization. Customer-modified hardware and/or software is not covered by this warranty.

Ordering Information

This IP is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana.

The order number for this IP is 235-600-400. To order, call:

1-888-LUCENT8 (1-888-582-3688) or fax to 1-800-566-9568 (From inside the continental United States)

1-317-322-6847 or fax to 1-317-322-6699 (From outside the continental United States).

Support Telephone Numbers

Information Product Support Telephone Number: To report errors or ask nontechnical questions about this or other IPs produced by Lucent Technologies, call 1-800-645-6759.

Technical Support Telephone Numbers: For initial technical assistance, call the North American Regional Technical Assistance Center (NARTAC) at 1-800-225-RTAC (1-800-225-7822). For further assistance, call the Customer Technical Assistance Management (CTAM) Center:

1-800-225-4672 (From inside the continental United States)

1-630-224-4672 (From outside the continental United States).

The CTAM center is staffed 24 hours a day, 7 days a week.

Acknowledgment

Developed by Lucent Technologies Customer Training and Information Products.

1. INTRODUCTION

1.1 PURPOSE

The 5ESS[®] Switch Audit Descriptions manual describes each software generated audit. An audit is a program that verifies the software memory for consistency of data. It performs this task by checking the database for discrepancies such as lost hardware data and broken links between data tables. When an audit finds a problem, it immediately initiates a process to correct it. If an audit cannot correct the problem, it is either because the problem is extensive and higher level audits are necessary, or because the data has an inconsistency that must be manually repaired using a recent change or the office database editor.

1.2 UPDATE INFORMATION

This document is being updated to include new and modified audit descriptions.

1.2.1 SUPPORTED SOFTWARE RELEASES

In accordance with the 5ESS[®] Switch Software Support Plan, the 5E11 software release will be rated Discontinued Availability (DA) on November 13, 1999. The information supporting 5E11 and earlier is being removed from all documentation. The actual removal of the information will occur over time instead of concurrently.

If you are supporting offices using a software release prior to 5E12 and have a need for the information that is being removed, it is recommended that you retain the associated pages as they are removed from the paper documents or the earlier copy of the CDRM.

1.2.2 TERMINOLOGY

This 5ESS[®] switch document may contain references to the 5ESS[®] switch, the 5ESS-2000 switch, and the 5ESS AnyMedia Switch. The official name of the product has been changed back to the 5ESS[®] switch. The documentation will not be totally reissued to change these references. Instead, the changes will be made over time, as technical changes to the document are required. In the interim, assume that any reference to the 5ESS-2000 switch or the 5ESS AnyMedia Switch is also applicable to the 5ESS[®] switch. It should be noted that this name change may not have been carried forward into software-influenced items such as input and output messages, master control center screens, and recent change/verify screens.

National ISDN is an evolving platform in which new features will continue to be introduced for new revenue opportunities, improved operational efficiencies, and for support of specific applications. NI 1, NI 2, and NI 3 represent specific features as documented in Bellcore SRs 1937, 2120, and 2457. The industry is migrating to an additional terminology to more specifically denote the availability of National ISDN features: NI 95, NI 96, etc. A feature is included in a specific version (such as, NI 96) if it is available by the switch vendors by the first quarter of the year.

As a result of the World Telecommunications Standardization Conference held March 1-12, 1993, the International Telegraph and Telephone Consultative Committee (CCITT), no longer exists as an organization under the International Telecommunication Union (ITU). According to the ITU, the CCITT is now referred to as the International Telecommunication Union - Telecommunication Standardization Sector (ITU-TS).

For new and revised Recommendations issued by the ITU-TS, the term "CCITT Recommendation X.xxx" will be replaced by the "ITU-T Recommendation X.xxx" designation. For a transition period from 1993 to 1997, if the Recommendation had a previous CCITT designation, the new name will include "(formerly CCITT Recommendation X.xxx)". Names of existing CCITT Recommendations will not change unless revised.

As of March 18, 1999, Bellcore officially changed its name to Telcordia Technologies. Not all pages of this document are being reissued to reflect this change; instead, the pages will be reissued over time, as technical and other

changes are required. Customers on standing order for this document may see that, on previous-issue pages, the Bellcore name is still exclusively used.

Customers receiving new orders for this document will see the Telcordia Technologies name used as appropriate throughout the document, and the Bellcore name used only to identify items that were produced under the Bellcore name. Exceptions may exist in software-influenced elements such as input/output messages, master control center screens, and recent change/verify screens. These elements will not be changed in this document until such time as they are changed in the software code. Document updates will not be made specifically to remove historical references to Bellcore.

1.3 ORGANIZATION

This document describes 5ESS[®] switch application operating system audits for all software releases from 5E10 and later. This document supplements standard Lucent Technology software documentation and assumes that its readers have taken basic training about the 5ESS[®] switch. The document contains background information about audit objectives and philosophy. It presents specific information about the audit system code, and it includes the error code comment blocks from the system code.

The discussion of audits in this document begins with background information introducing general concepts about the Audit (AU) subsystem. It progresses through descriptions of the audits and information about how to use them. The document is divided into seven main parts.

- **Introduction** introduces the 5ESS[®] switch Audits Document and describes its contents briefly. It discusses terminology and documentation conventions.
- **Software Overview** contains background information about the 5ESS[®] switch software necessary for understanding audits.
- **Audits Overview** describes the software concepts specific to the audit system including modes of audit operation and audit control.
- **Using Audits** presents details on manually controlling audits and interpreting audit output. Several case histories trace through receive-only printer (ROP) output to find the sources of errors.
- **Audit Drivers** explains the utility programs called by various audits to perform similar verifications.
- **Audit Descriptions** discusses each audit separately. The audits are documented alphabetically. Most audit descriptions have four parts: an introduction, a list of the data structures with which the audit is associated, a description of the audit routine, and an explanation of how the audit handles the errors it encounters.
- **Glossary** lists terms used in this manual and their definitions

1.4 DOCUMENTATION CONVENTIONS

The Audits Document contains audit descriptions for all software releases. When there are major changes to an audit the audit description is **reissued**. Alphabetic suffixes are used to distinguish reissued audit descriptions. The original audit description has a -A appended to the name of the audit and the new version of the audit description has a -B appended to the end of the audit name.

In the Audit Drivers and the Audit Descriptions sections, software release information is included under the driver and audit names in the upper right-hand corners of the first pages of the descriptions. The software release information includes the **base software release** (the software release in which the audit description applies.) Usually the range is "**and later**," but in some cases, such as the BD audit, the range is specific.

1.5 USER FEEDBACK

The producers of this manual are constantly striving to improve its quality and usability. For comments and to advise us of any errors, you can write to us at the following address:

Lucent Technologies
Documentation Services
2400 Reynolda Road
Winston-Salem, NC 27106

Please include the issue number and/or date of the document, your complete mailing address, and telephone number. We will answer all correspondence within 30 days, informing you of the disposition of your comments.

You may also call our Documentation HOTLINE if you need an immediate answer to a documentation question. This HOTLINE is not intended to eliminate the use of the comment cards, but rather to enhance the comment process. The HOTLINE number is 1-800-645-6759 and is available from 7:30 a.m. to 6:00 p.m. Eastern Standard time. Outside of those hours, the line is served by an answering machine. You can leave a message on the answering machine, and someone will return your call the following business day.

Document users who have access to electronic mail facilities may send comments via electronic mail to **hotline5@wrddo.lucent.com**. When sending comments via electronic mail please make sure that the document title, document number, document issue number and/or document date are included in the mail along with the sender's name, phone number, and address.

1.6 DISTRIBUTION

This document is distributed by the Customer Information Center in Indianapolis, Indiana. Most service providers should place orders through their documentation coordinator. Some may allow customers to order directly from the Customer Information Center; however, the majority do not. Service providers that use documentation coordinators to manage their orders receive a significant discount. If you do not know the name/number of your documentation coordinator, you may call 1-888-582-3688.

Customers not represented by a documentation coordinator and Lucent Technologies employees can order the documentation for the 5ESS[®] switch directly from the Customer Information Center. Proper billing information must be provided. These orders may be mailed to:

Lucent Technologies
Customer Information Center
Order Entry
2855 N. Franklin Road
Indianapolis, IN 46219

Orders may be called in on 1-888-LUCENT-8 (1-888-582-3688) within the continental United States and +1-317-322-6416 from outside the continental United States. Orders may be faxed in on 1-317-322-6484.

1.7 TECHNICAL ASSISTANCE

Technical assistance for the 5ESS[®] switch can be obtained by calling the North American Regional Technical Assistance Center (NARTAC) at 1-800-225-RTAC (1-800-225-7822). This telephone number is monitored 24 hours a day, 7 days a week. During regular business hours, your call will be answered by your local NARTAC. Outside of normal business hours, all calls will be answered at a centralized technical assistance center where service-affecting problems will be dispatched immediately to your local NARTAC. All other problems will be referred to your local NARTAC on the next regular business day.

1.7.1 REFERENCES

The following documents will assist switch personnel in understanding and interpreting audit reports.

235-600-22x, Translations and Dynamic Data Reference Manual: Contains data structure layouts for dynamic and static data types.

235-600-23x, Dynamic Data Manual: Contains data structure layouts for dynamic and static unprotected random access memory (URAM) relations.

235-600-31x, ECD/SG Data Base Manual: Includes information on how to use recent change transactions to modify switch static office dependent data.

235-600-400, Audits Manual: Contains an overview of application audits, execution modes, environments and so on. It also contains descriptions of each application audit and their associated error codes.

235-600-510, Software Analysis Guide: Contains information on how to analyze ROP.

235-600-700, Input Message Manual: Includes audit input message information.

235-600-750, Output Message Manual: Includes audit output message information.

An "x" or "x"s in the last three positions of a release specific document number indicate the digits that change from release to release. Refer to 235-001-001, Documentation Description and Ordering Guide for the document number associated with each software release. Additional references are given as needed, within individual message descriptions.

This document also contains information concerning the Compact Digital Exchange (CDX) and the Very Compact Digital Exchange (VCDX), which are switching systems based on the 5ESS[®] switch. This document is a standard 5ESS[®] switch document that is also applicable to the CDX and VCDX switching systems. Information applicable only to CDX or VCDX may be found in the following manuals:

235-120-010, Compact Digital Exchange (CDX) Reference Guide

235-120-020, Compact Digital Exchange (CDX) User's Guide

235-120-120, Very Compact Digital Exchange (VCDX) User's Guide

2. SOFTWARE OVERVIEW

2.1 SOFTWARE ORGANIZATION

An audit is a means of verifying data. In the 5ESS[®] switching system, audits verify software data. The audit system makes up an important part of the 5ESS[®] switch software. It represents about ten percent of the lines of code in the system and is powerful enough to trigger full system initializations, which disconnect stable calls. This section introduces the audit system and its relationship to the 5ESS[®] switch software.

The 5ESS[®] switch software is distributed between the administrative module and the switching module processors. As a distributed system, the 5ESS[®] switch has a structured hierarchy of subsystems to enhance its efficiency. These subsystems are depicted in Figure 2-1 ; they ensure that the system is adaptable to changing technology and relatively easy to understand. The 5ESS[®] switch software is coded in a high-level language, C, to enhance program development and readability. Because the system is distributed, the number of interfaces with the UNIX[®] RTR operating system in the administrative module is limited. Common functions manage resource allocation and other services.

The features and design philosophy of the switching system as a whole are reflected by the audit system. The function of the audit system is to ensure reliable service by detecting and correcting errors, and recovering lost resources, which increases the efficiency of the system. The functions of checking for unavailable resources are performed by a subsystem, which increases the adaptability of the system to changes in technology.

The five major system areas that constitute the software of the 5ESS[®] switch are: the operating system, call processing, administration, database, and maintenance areas. Audits verify data in all five software areas. The audit system is a system integrity feature in the maintenance area, as described in the "Maintenance Software" section.

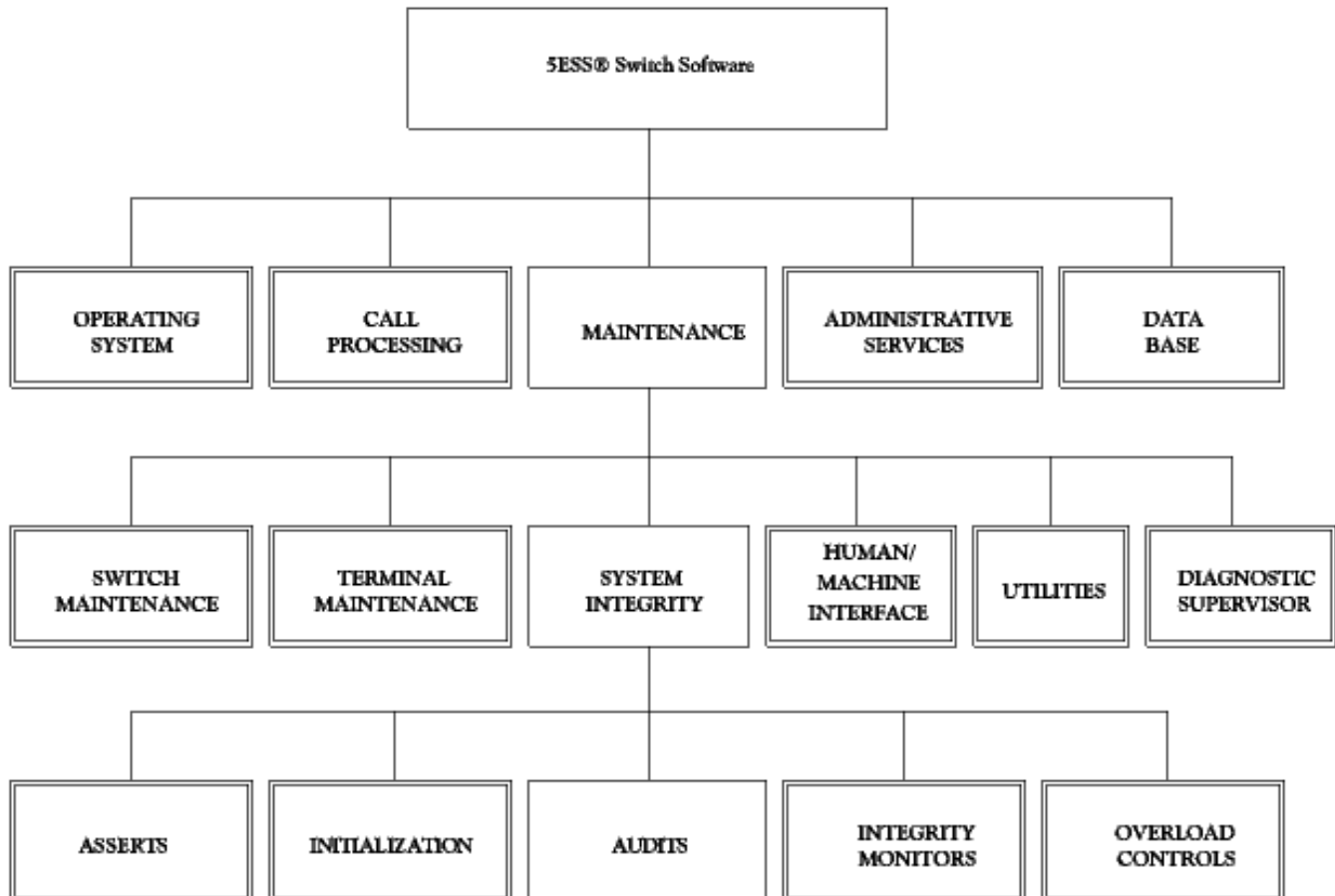


Figure 2-1 5ESS Switch Software Organization

2.1.1 MAINTENANCE SOFTWARE

The largest system area is the maintenance area. It consists of six major subsystems: switch maintenance, terminal maintenance, human/machine interface, system integrity, utilities, and diagnostic supervisor.

- Switch maintenance detects hardware errors related to the administrative module and peripheral units such as the message switch, the time-multiplexed switch, and the switching modules.
- The terminal maintenance subsystem tests the lines, trunks, and signaling paths both automatically and by user request.
- The human/machine interface subsystem comprises the software by which most user interaction with the system occurs, primarily through the master control center (MCC).
- The system integrity subsystem ensures that the 5ESS® switch continues functioning despite the occurrences of hardware and software faults.
- The utilities subsystem helps verify application programs and resolve software problems.
- The diagnostic supervisor subsystem organizes requests for diagnostics in the distributed 5ESS® switch environments. The goal of all these subsystems is to guarantee the reliability of the switching system. They guarantee that the initial start-up of an office goes smoothly, and allow the switching system to be repaired, controlled, and monitored effectively.

2.1.2 SYSTEM INTEGRITY

There are five system integrity features in the maintenance area: asserts (defensive checks), integrity monitors, overload controls, initializations, and audits. One system integrity feature may invoke another depending on the severity of the faults detected. These system integrity features are briefly discussed here.

Asserts (Defensive Checks): Asserts, or defensive checks, are short algorithms placed in the code of many programs (including audit programs) to check the validity of data. They are quick checks to ensure that the data used is within a specified range, and that duplicate copies of data are consistent with each other. More information about asserts is given in the 5ESS[®] Switch Asserts Manual (235-600-500).

Integrity Monitors: The system integrity monitors have a global view of the switching system and ensure that all parts of it are processing calls efficiently. The integrity monitors use sanity timers and various software checks to perform this task. If a portion of the system is no longer performing within the specified tolerances, the integrity monitors will invoke the appropriate level of initialization.

Overload Controls: Overload controls limit the extent of service degradations that would otherwise result when call processing demands exceed the capability of the system. This system integrity feature includes essential service protection, which allows calls to essential lines (such as police and the fire department) to be processed.

Initializations: Initializations ensure the congruence of hardware status and software data. Various levels of initialization are possible, depending on the nature of the error. Audits, asserts, integrity monitor tests, and hardware interrupts cause initializations.

The "Audit System" section briefly introduces the fifth system integrity feature, audits.

2.1.3 AUDIT SYSTEM

The audit system examines the changes made to the data during the call processing, maintenance, administration, operating system, and data base management functions of the switching system. The audits check this dynamic data against static (write-protected), recent-changeable data, or against other dynamic data. The audit system runs at a low priority to minimize interference with operational software, unless critical resources are jeopardized. In these cases, audits run in a mode that affects the operational software until the errors are corrected. All errors found are reported. Usually, however, the audits do not noticeably interfere with the operational software. And if they find no errors, they run silently.

Each audit is a short program that verifies the data in a specified data structure(s), or the linkages among certain data structures. This is important to system integrity as a whole to ensure that errors do not propagate through the system, and to ensure that resources are not lost to call processing activities.

Audits run automatically but may be invoked manually by means of input messages. Along with their routine operating cycle, audits are invoked by asserts, single process purges, and initializations. When they detect errors, audits recover lost resources and, if necessary, invoke appropriate initializations.

2.2 SOFTWARE CONCEPTS

The software concepts used in this document are explained in two subsections, "Data Structures" and "Processes." The "Data Structures" subsection is important to audit users because the discussions in the rest of this document are in terms of data structures. The "Processes" subsection defines what the general term "process" means to the audit system, and how it is used. Items covered here include:

- Classes of data
- Relations and tuples, including their structures, types, and components
- Arrays and linked lists
- Linkages
- Components of processes
- Types of processes.

2.2.1 DATA STRUCTURES

To understand the individual audit descriptions contained in this document, it is necessary to be familiar with the data structures on which the audits operate. This section describes basic data structure concepts and their application to the 5ESS[®] switching system. It is organized in three parts. The first part, "Classes of Data," describes the distinction between data managed by the database manager and private data, and the storage structures used by each of these two classes of data. The second part, "5ESS[®] Switch Data Structures," describes the makeup of data structures within the 5ESS[®] switch system. The third part, "Linkage," describes linkages among the 5ESS[®] switch data structures. The glossary contains concise definitions of the terminology used in the discussion of the data structures.

2.2.2 CLASSES OF DATA

There are two classes of data in the 5ESS[®] switch system: data managed by the database manager program (DBM), and *private* data managed by application programs independently of the DBM.

2.2.2.1 DATA MANAGED BY THE DBM

The data managed by the DBM is stored as a *relational database*. In a relational database, all data is organized in two-dimensional tables called *relations*. Relational databases give the application programs a relatively simple view of the data. Details of the physical storage structure of data and access algorithms to it become transparent. These advantages are offset by two disadvantages: relational databases tend to be slow, and they require a great deal of storage. For these reasons, the 5ESS[®] switch system uses a modified relational database. The application view is relational, with several extensions, but the actual implementation has elements of hierarchical and network databases (database types used in earlier electronic switching systems). Hierarchical and network databases are characterized by *physical linkage* (records contain the addresses of other records); purely relational databases do not use physical linkage. The elements of these types of databases used in the 5ESS[®] switching system include support for linear searching of ordered lists of tuples (a tuple is a row in a relation) and networking, a way of linking tuples.

2.2.2.2 PRIVATE DATA

Private data managed directly by an application program takes whatever structure is defined by the particular program. A frequently used structure is an *array*, which consists of *elements* of the same size. The elements may be simple variables, arrays (the C language supports arrays of arrays), or more complex *C-language structures*. A C-language structure is a collection of variables, possibly of different types, grouped together under a single name for convenient handling. The variables in a structure are called *members* (not to be confused with members in the context of owner-member linkage).

2.2.3 5ESS[®] SWITCH DATA STRUCTURES

The data structures used most frequently in the 5ESS[®] switching system are relations and arrays.

2.2.3.1 RELATIONS AND TUPLES

A relation is a two-dimensional table. The columns of a relation are called *attributes*; the rows are called *tuples*. The set of possible values for an attribute is called the *domain* of the attribute. Relations have these properties:

- No two tuples are identical.
- The ordering of tuples is insignificant.
- All values of a tuple are atomic data items (that is, they cannot be split further without loss of information).

The parts of a relation are shown in Figure 2-2. Each tuple of a relation has the same length and structure. The first tuple is called a *dummy tuple* and is not used. All the words of the dummy tuple except the logical key (see below) are set to zero; thus, by examining the dummy tuple for nonzero words, the system detects *wild writes*. The next N tuples (array indices 1 to N) are called *head cells*. Head cells are normally used as owners for lists of resources such as time slots and circuits. The owned resources exist in other relations. The remaining tuples are the *data tuples*. These are the tuples requested and manipulated by call processes.

TUPLE

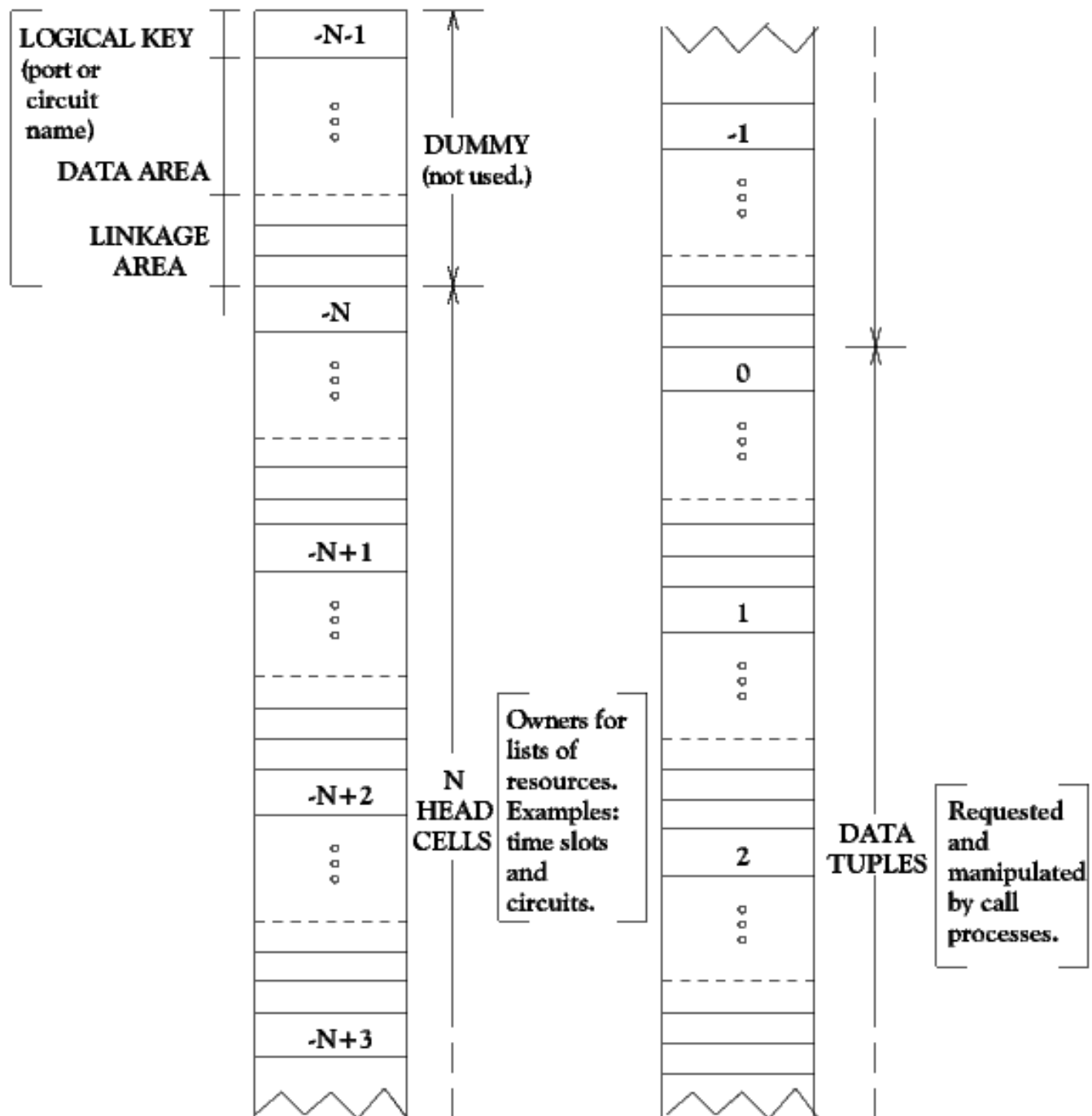


Figure 2-2 5ESS Switch Relation Structure

A tuple consists of a number of atomic data *items*. An item is a word of memory within a tuple, a portion of a word of memory within a tuple, or a number of contiguous words of memory within a tuple. A number of contiguous items in a tuple is called an *area*. A tuple is sometimes called a *block* (in other contexts, a block is a contiguous area of memory).

Each tuple has three components. The first component is the *logical key*. For ports, the logical key is the port name. For circuits, it is the circuit name. For all other tuples, it is a number obtained by this equation:

Logical Key = Physical Key - (Number of Head Cells + 1).

Example: -3 = 6 - (8 + 1)

For example, a tuple with physical key 6 in a relation with 8 head cells has a logical key of -3. The *physical key* is the array index or element label. It is the key stored in memory to implement linkage. The logical key is the user key; it is used in data access primitives. The equation relating the physical and logical keys yields negative logical keys for dummy tuples and head cells, and nonnegative logical keys for data tuples.

The other components of a tuple are the data and linkage areas. The data area contains the information manipulated by the application; it may contain linkages if the tuple is linked to several different data structures. The linkage area contains links to other tuples. The logical keys and linkage areas are referred to as *syntactic* data; the data areas are referred to as *semantic* data.

2.2.3.2 ARRAYS

An array is a number of *elements* all the same type and size. As an example of an array application, suppose that the message "SYNTAX ERROR" must be stored. Figure 2-3 shows how an array of characters stores this message. Each element of the array contains the ASCII value for one of the message characters. Because the array has only one row, it is called a *one-dimensional array* (or *sequential list*). The elements are labeled 0 to 11 from left to right. The label associated with an element is called its *index*. Thus, assuming that the array is called "MESSAGE," the letter T is addressed as MESSAGE[3].

0	1	2	3	4	5	6	7	8	9	10	11
S	Y	N	T	A	X		E	R	R	O	R

Figure 2-3 One-Dimensional Array

Suppose that the information in Table 2-1 must be stored. Figure 2-4 shows how an array of integers stores the information. Because the array has more than one row, it is called a *two-dimensional array*. The first row contains the part numbers and the second row contains the quantities in stock. Both the rows and columns have indices. Thus, assuming that the array is called "STOCK," the number 757 is addressed as STOCK[1][3]. (Using brackets around each index is a C language convention. Other languages may require references of the form STOCK[1,3].) To look up the quantity in stock for a particular part number, the first row is scanned from column index 0 until the desired part number is found. The column index of the part number is noted and then used to address the corresponding quantity in the second row. This is where the term relational database comes from: all the elements in a row are related to each other. The columns indicate how they are related. The example in Table 2-1 shows that part number 89 has a related quantity in stock of 757.

Table 2-1 Example of Table Requiring Two-Dimensional Array

Part Number	Quantity in Stock
3	83
22	7
45	56
89	757
105	0
690	6

	0	1	2	3	4	5
0	3	22	45	89	105	690
1	83	7	56	757	0	6

Figure 2-4 Two-Dimensional Array

0	1	2	3	4	5
13	22	45	89	105	690

PART

	0	1	2	3	4	5	6	7	8	9	10
0	B	O	L	T							
1	N	U	T								
2	F	L	A	T		W	A	S	H	E	R
3	L	O	C	K		W	A	S	H	E	R
4	S	P	A	C	E	R					
5	C	O	T	T	E	R		P	I	N	

NAME

0	1	2	3	4	5
0.06	0.04	0.01	0.02	0.05	0.04

PRICE

Figure 2-5 Multiple Arrays Used to Store Table with Different Data Types

As a final example, consider Table 2-2. Because the three table columns contain different types of data (the first column contains integers, the second characters, and the third real numbers), three separate arrays must be used.

Table 2-2 Example of Table Requiring Multiple Arrays

Part Number	Name	Price

13	Bolt	0.06
22	Nut	0.04
45	Flat washer	0.01
89	Lock washer	0.02
105	Spacer	0.05
690	Cotter pin	0.04

Figure 2-4 shows one scheme for storing the information. PART is a one-dimensional array of integers that contains the part numbers. NAME is a two-dimensional array of characters that contains the part names. PRICE is a one-dimensional array of real numbers that contains the part prices. Note that a single index can access an entire row of Table 2-2. For example, the part number for spacers is found at PART[4], the name begins at NAME[4][0], and the price is found at PRICE[4].

In the previous examples, *packed* arrays were used. The term packed means that there are no gaps between items. For example, if a one-dimensional array can hold 50 items yet only contains 25 items, these items occupy indices 0 through 24. By using arrays that are not packed, information access can be made quicker and easier. Suppose that the part numbers in Table 2-2 all fall in the range 1 to 49. The name of, say, part number 15 is stored beginning at NAME[15][0] and the price is stored at PRICE[15]. The PART array may be eliminated altogether. Instead of having to scan the PART array to determine which index to use to access the NAME and PRICE arrays, the part number itself may be used as the index. The arrays would contain gaps corresponding to unassigned part numbers. Despite the extra storage space required by unpacked arrays, they are advantageous because new parts can be added easily. To add a new part to packed arrays, the arrays must be expanded and the items must be shifted to make room for the new part. The unpacked scheme is useful and practical when the indexing item falls within a relatively small range of values and when the percentage of the values actually assigned is relatively large.

2.2.3.3 RELATIONS VERSUS ARRAYS

As explained in the "Arrays" section, tables (and thus relations) may be stored in arrays. The audit descriptions later in this volume contain references to both relations and tuples and to arrays and elements. However, the concepts are not entirely interchangeable. The concepts of relations and tuples are purely mathematical; they do not imply any specific implementation in terms of basic data structures. One possible implementation, however, is storing relations as arrays. If this scheme is adopted, relations are analogous to arrays. For one-dimensional arrays, tuples are analogous to elements.

The 5ESS[®] switch system stores only the data managed by the data base manager program (DBM) in relations. Other *private* data is managed by application programs independently of the DBM and takes whatever form is appropriate to the particular application programs. For example, call records are stored as arrays of elements; each element is a C-language structure. The distinction between data managed by the DBM and privately managed data explains why some audit descriptions refer to relations and tuples, and others refer to arrays and elements.

2.2.3.4 LINKED LISTS

A list is a collection of items of the same type arranged in a linear sequence. A list may be represented as a one-dimensional array. However, this approach has two disadvantages. First, additions and deletions to the list are awkward. To add an item at index position N within the list, all items having indices equal to or greater than N must be shifted one position to make room for the added element. Similarly, deletions require shifting to close the resulting gaps. For long lists, this shifting is time-consuming. Second, arrays are declared with fixed bounds. A list must therefore be declared with a bound large enough to hold the maximum number of items anticipated. If the list is considerably shorter than the maximum length for most of the time that it is used, storage space is wasted. The use of linked lists avoids these disadvantages and obtains greater flexibility.

In a *singly linked list*, each individual element in the list (called a *node*) contains data and a pointer to the next position. Thus, the nodes are linked by pointers. This is illustrated in Figure 2-6. A *list pointer* points to the first node in the list. The pointer of the last node in the list contains a zero (or another distinctive value) to denote the last node in the list. Note that the nodes do not necessarily occupy successive memory addresses. The list may be scanned, or *traversed*, by examining the list pointer to find the first node and then tracing along the node pointers.

Adding and deleting nodes is achieved by simply manipulating pointers; no shifting of data is required. To delete node 3 in Figure 2-6 the pointer in node 2 is changed to point to node 4. To add a new node between nodes 3 and 4, the pointer in node 3 is changed to point to the new node and the pointer in the new node is set to point to node 4. Allocation of storage using linked lists is also more flexible and efficient. A linked list of available nodes can maintain an *idle list*. When other lists composed of the same type of node need to be expanded, nodes may be drawn from this idle list. Similarly, when a list no longer needs one or more nodes, the nodes may be relinked to the idle list. This scheme manages resources and keeps the amount of unused memory to a minimum.

Linked lists have two disadvantages. First, a linked list requires more space than an equivalent list created with a one-dimensional array (assuming the array is completely full) because space must be allocated for the pointers. Second, node access is slower with linked lists. Access to node N in a list created with a one-dimensional array is achieved by simply referencing LIST[N-1]. Access to node N in a linked list requires starting at the first node and tracing along the pointers until node N is reached.

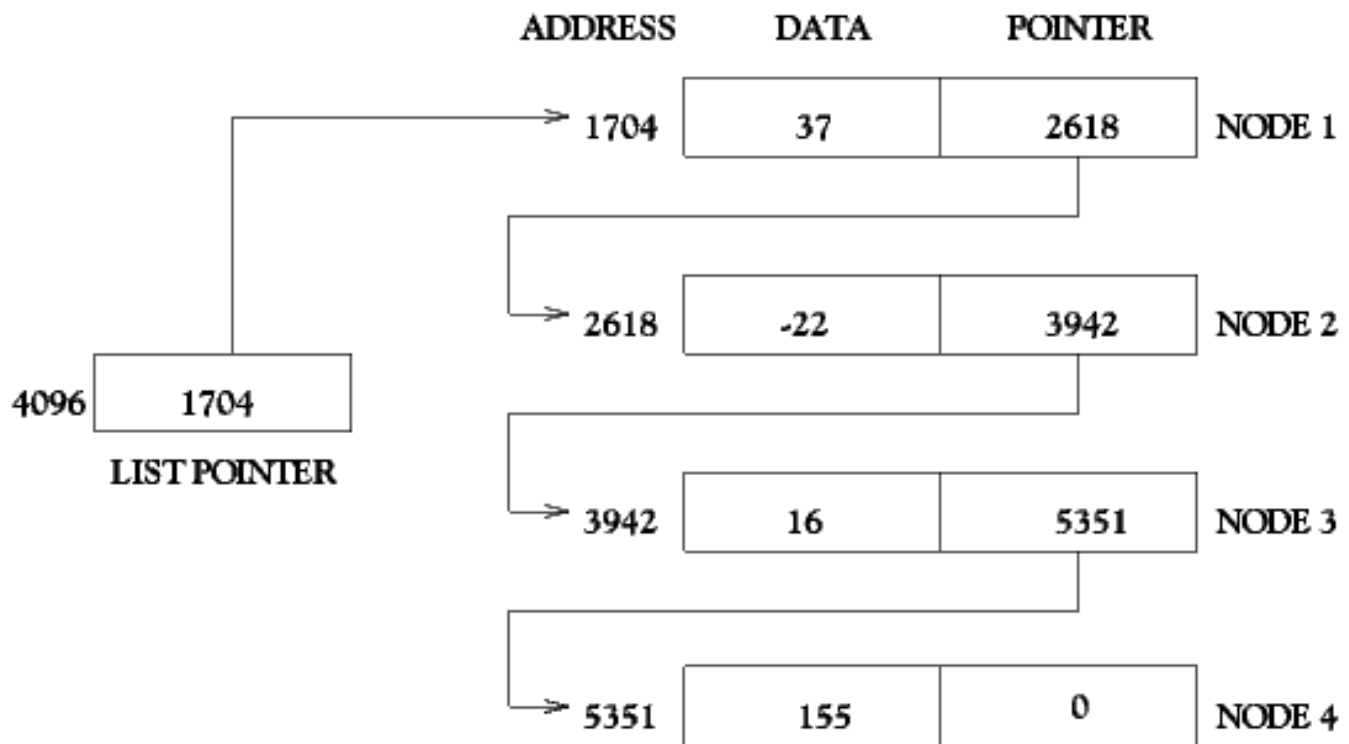


Figure 2-6 Singly Linked List

These disadvantages are relieved by using *circular lists* and *doubly linked lists*. In a circular list, the last node contains a pointer to the first node. The list in Figure 2-6 is made circular by changing the pointer in node 4 from 0 to 1704. In addition, circular lists maintain pointers to the current node. In a doubly linked list, each node contains both a pointer to the previous node (the *predecessor*) and a pointer to the next node (the *successor*). This allows the node to be traversed in either direction. Both circular and doubly linked lists simplify node access.

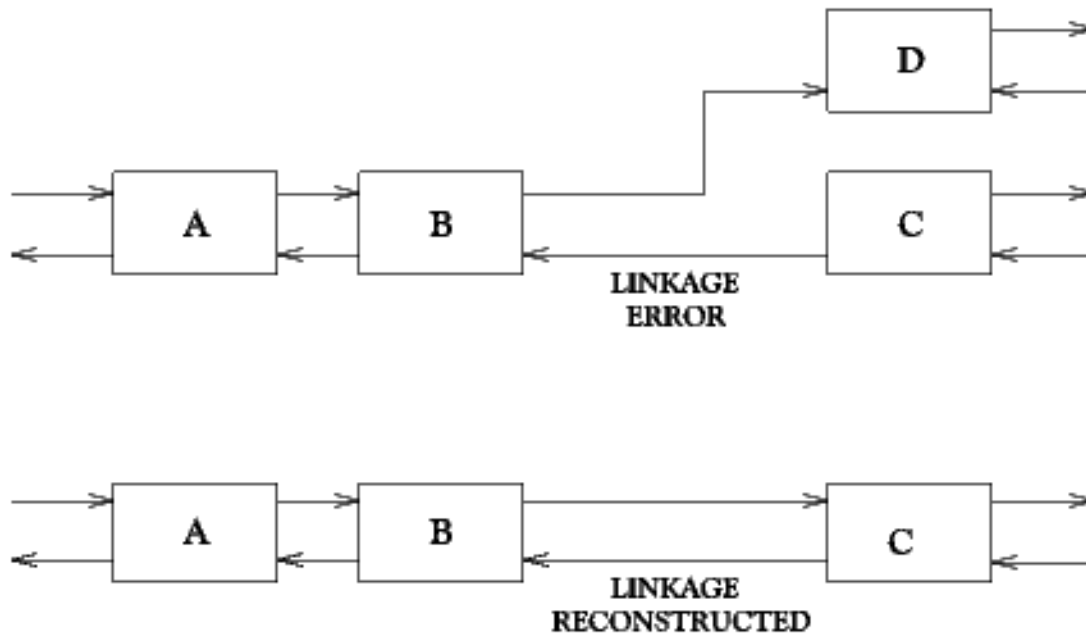


Figure 2-7 Linkage Reconstruction by Audits

With doubly linked lists, linkage errors may be detected by the application code, which checks the linkage defensively by means of asserts; and by audits, which traverse the list specifically looking for inconsistencies. Linkage errors in just one of the links of a doubly linked list may be recovered by reconstructing the list (Figure 2-7). If the first and last elements (A and C) in the list are known, an audit may traverse the list from both ends and rebuild the list. The audit ensures that B links to C. List reconstruction is important particularly when the list contains critical system information.

Figure 2-8 summarizes the types of linked lists.

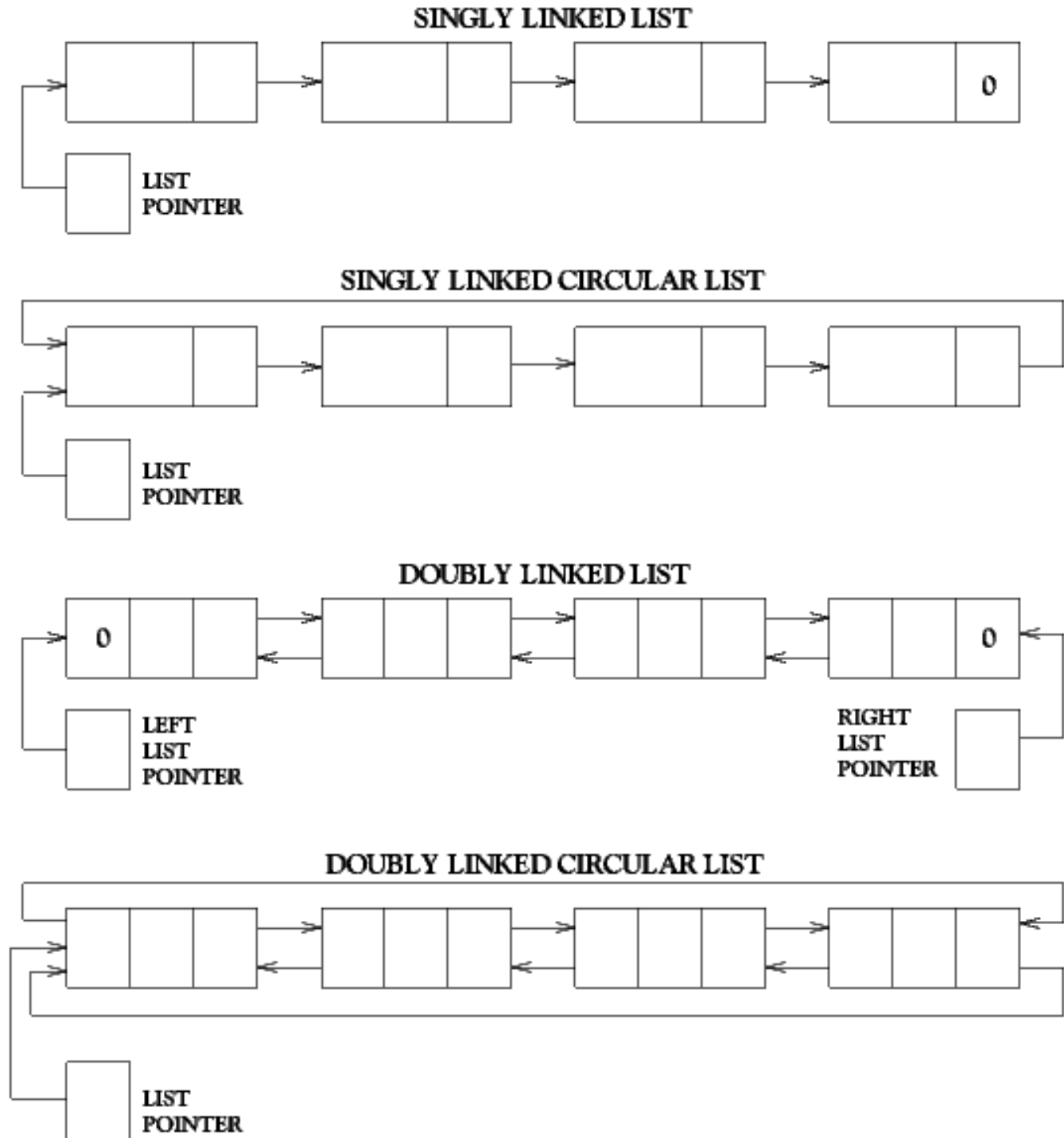


Figure 2-8 Types of Linked List

2.2.3.5 STACKS

A stack is a list that may have nodes added or removed only at one end (Figure 2-9). A stack is analogous to a pile of trays in a cafeteria. When a tray is added to the stack, springs below the pile are depressed so the top of the stack remains at the same place. When a tray is removed from the stack, the pile rises so the next tray is at the top position. Because the last item to be added to a stack must be the first removed, stacks are also called last-in,

first-out (LIFO) lists. The operation of adding a node to the stack is called *pushing*; the operation of removing a node is called *popping*.

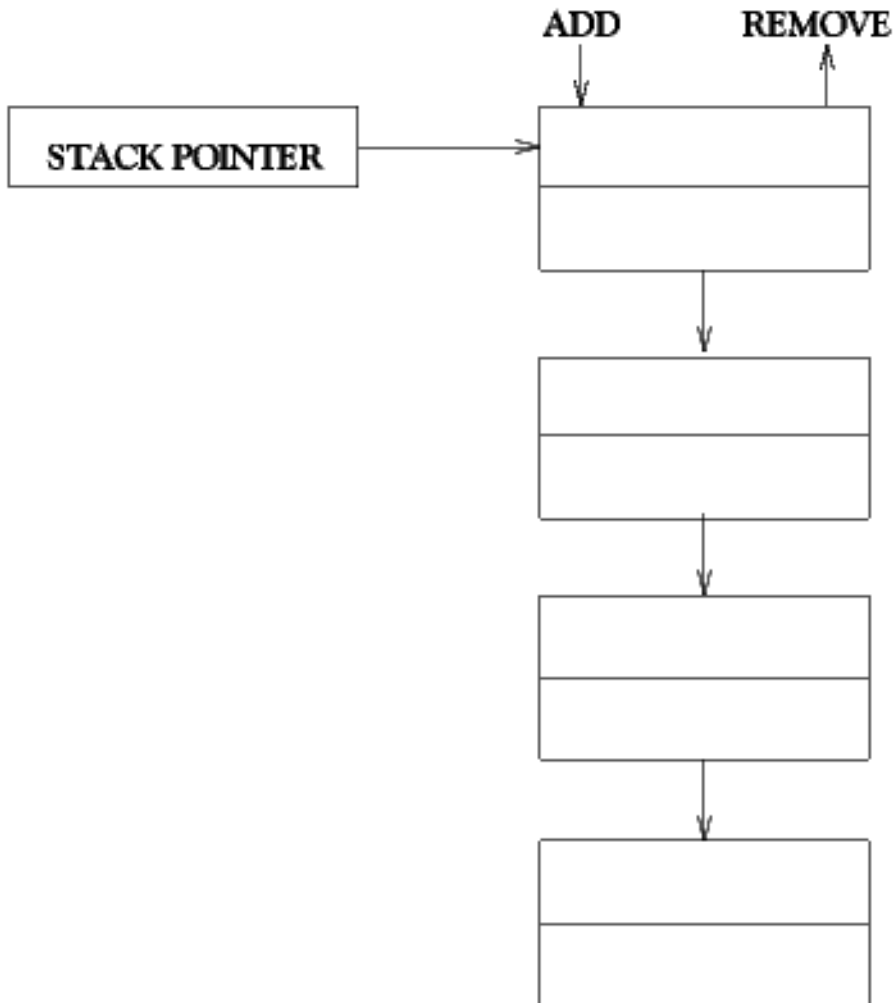


Figure 2-9 Stack (LIFO List)

All C language functions use stacks to store function parameters, local variables, and the return address. In addition, special recovery stacks are maintained by the operating system for distributed switching (OSDS) and system integrity.

2.2.3.6 QUEUES

A queue is a list that may have nodes added only at one end and removed only at the other (Figure 2-10). An analogy is the line at a grocery store. Because the first item that is added to a queue must be the first to be removed, queues are also called first-in, first-out (FIFO) lists.

The OSDS uses queues to store process control blocks (in this context, block is synonymous with tuple) that are in the READY state.

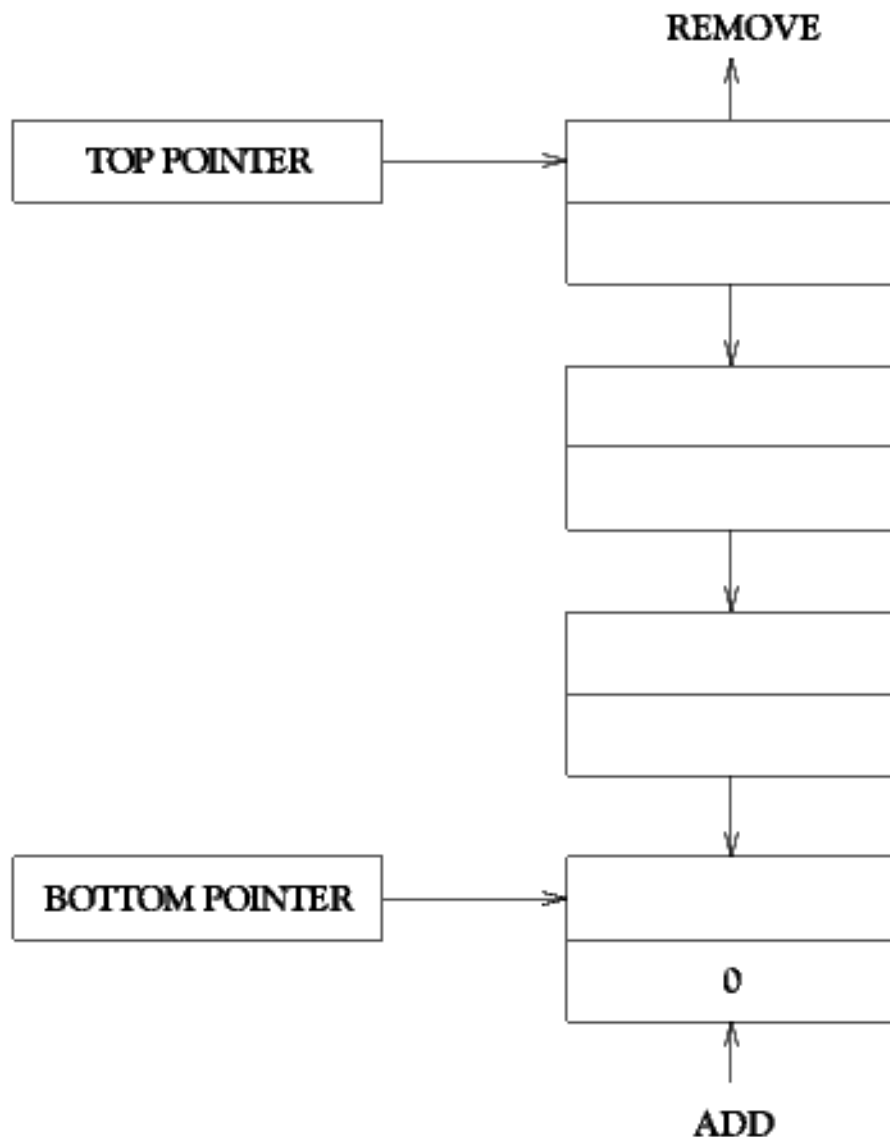


Figure 2-10 Queue (FIFO List)

2.2.3.7 HEAPS

A heap is a list that is similar to a stack in that it stores data in consecutive locations in memory and may have nodes added or removed only at one end (Figure 2-11). Unlike a stack, in which the nodes are arranged linearly and each node points to the next available stack location, a heap is shaped like a binary tree and each node points to the first location in the heap.

Heaps are used in 5ESS[®] switch software to facilitate process timing (see the TCB audit description).

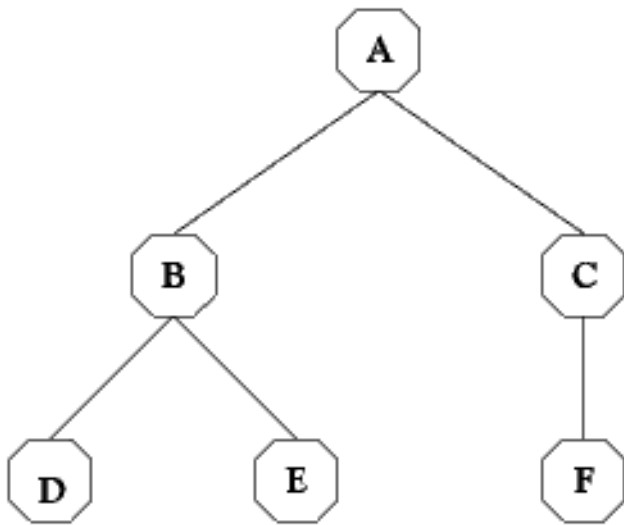


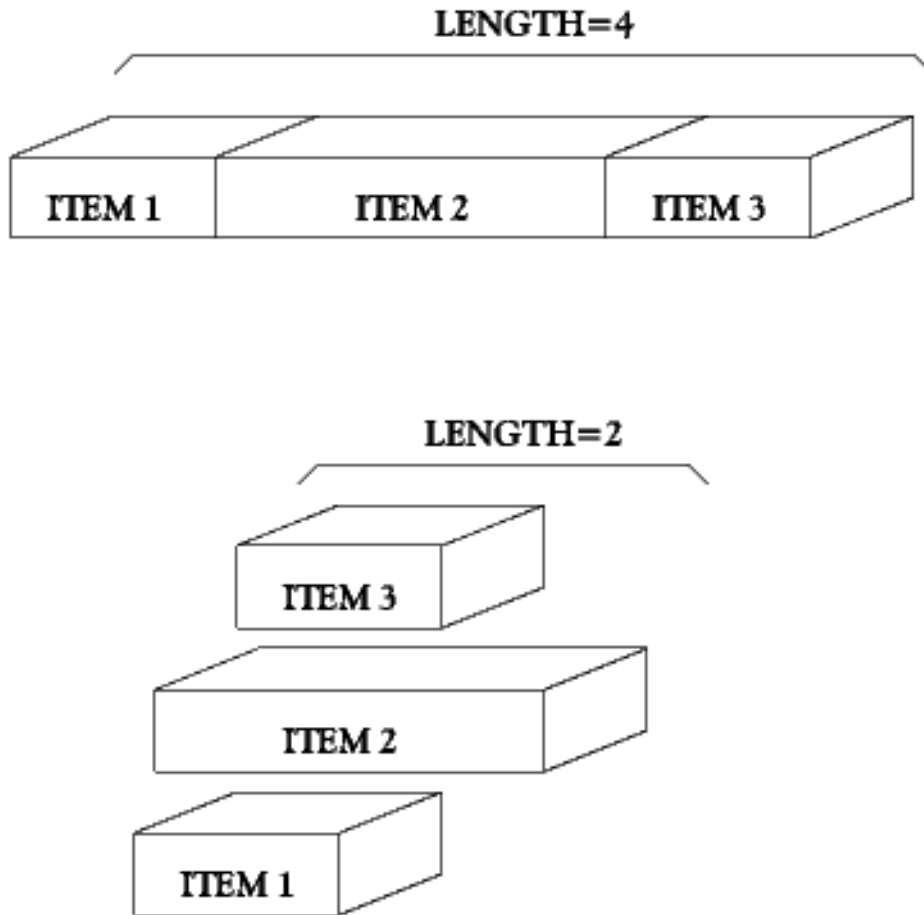
Figure 2-11 Heap

2.2.3.8 UNIONS AND STRUCTURES

Two other types of data structures occur in the 5ESS[®] switch software: *unions* and *structures*. Structures store data sequentially so it is well-organized and easily accessed. Unions are a type of structure used for mutually exclusive data; because unions use the same memory locations, they save space. Figure 2-12 illustrates structures and unions. Structures link together their items whereas unions overwrite the memory locations of their items.

Characters are one byte long; shorts, another type of data, are two bytes long. The size of the structure of items 1, 2, and 3 (two characters plus one short) is four bytes. The size of a union is the length of its longest item; therefore the length of the union of items 1, 2, and 3 is the length of item 2: two bytes.

Only a few relations (such as the process control block relation in OSDS) use unions.



ITEM 1: LENGTH=1
 ITEM 2: LENGTH=2
 ITEM 3: LENGTH=3

Figure 2-12 Structure and Union

2.2.4 MESSAGES

Processes communicate by sending *messages* to each other. Each message contains two parts: header information and data (Figure 2-13). The header information has a fixed length; it contains administrative information including the source and destination of the message, its priority, its type, and the length of the data section. The usage field is for transferring messages between buffers (see the CMIM audit description). The data sections vary in size depending on their functions. The data sections may include calling party information (for call processing processes), circuit names and error histories (for maintenance processes), and customer information (for administrative services processes).

Intraprocessor messages are handled by OSDS. Messages between processors or within the same processor in different environments are handled by the operating system and the communication package kernel process (CMKP). CMKP stores messages in message buffers; an audit checks the message buffers to ensure that communication proceeds efficiently.

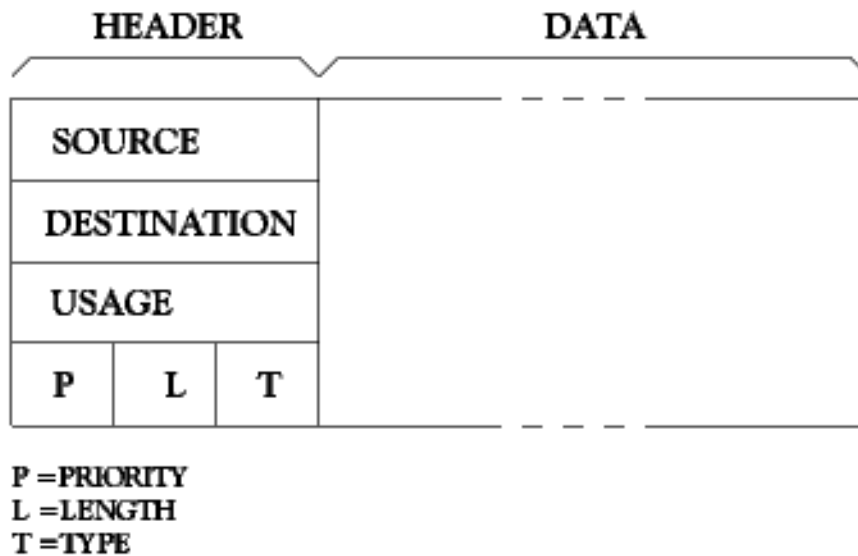


Figure 2-13 Message Layout

2.2.4.1 STATIC VERSUS DYNAMIC

There are two types of data within the system: *static* and *dynamic*. Static data is write-protected data; it may only be changed by explicit requests from user interfaces such as the recent change/verify system. Dynamic data refers to nonwrite-protected data. Audits operate primarily on dynamic data structures.

2.2.5 LINKAGE

The database design and call processing programs often require tuples to be linked to other tuples. For example, idle call control block tuples are linked to form a list of idle tuples. When a telephone call originates, tuples from the idle list are linked to implement the call. In essence, a *link* is made by assigning an item in one tuple as a pointer to another tuple. Figure 2-14 illustrates this process. If the system requires tuple D from the idle list to implement a call, it assigns tuple C to point to tuple D. Tuple D breaks its linkage to the idle list and points to tuple B (Figure 2-14, part B). In Figure 2-14, part C the new linkage is completely formed. For generically fixed relations, the pointer item is simply the physical key of the tuple to be linked. If the tuple to be linked can exist in one of several relations, the relation is implicitly specified by the position of the link item in the linkage area.

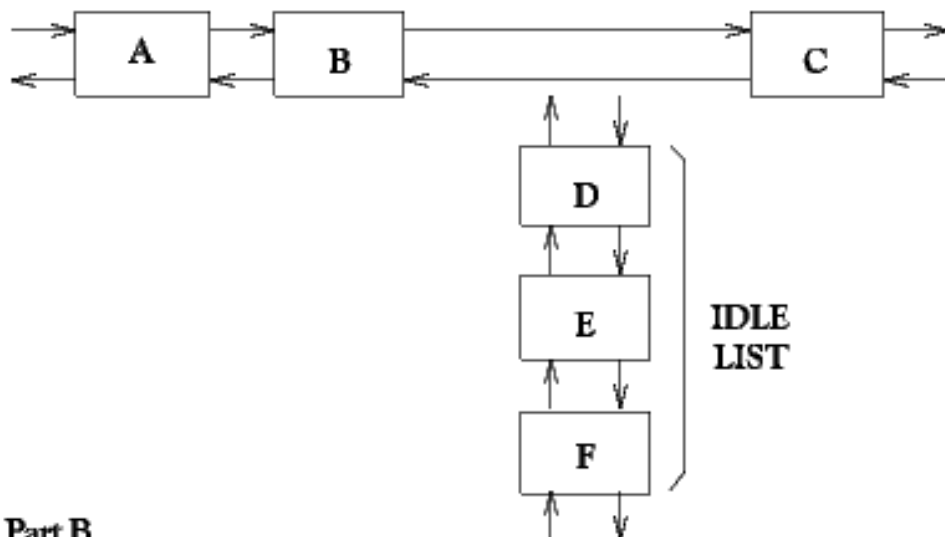
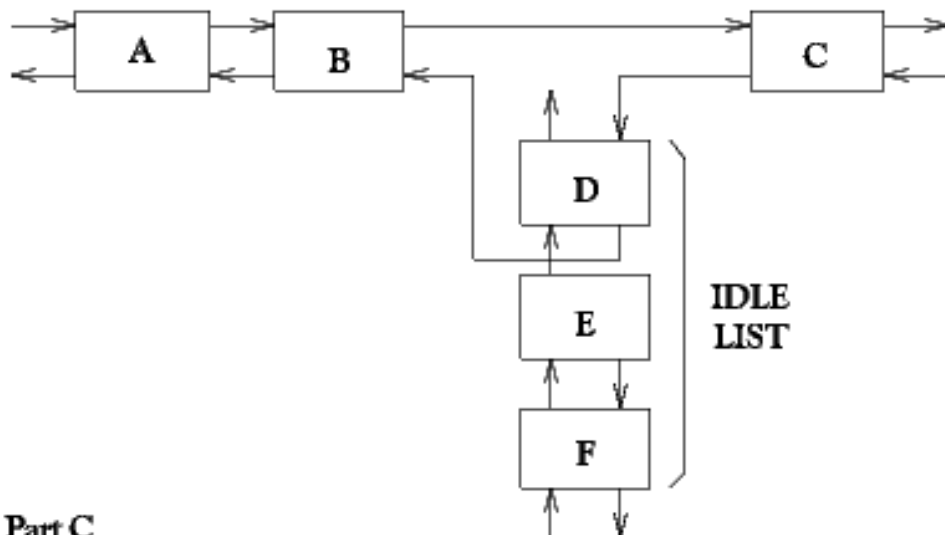
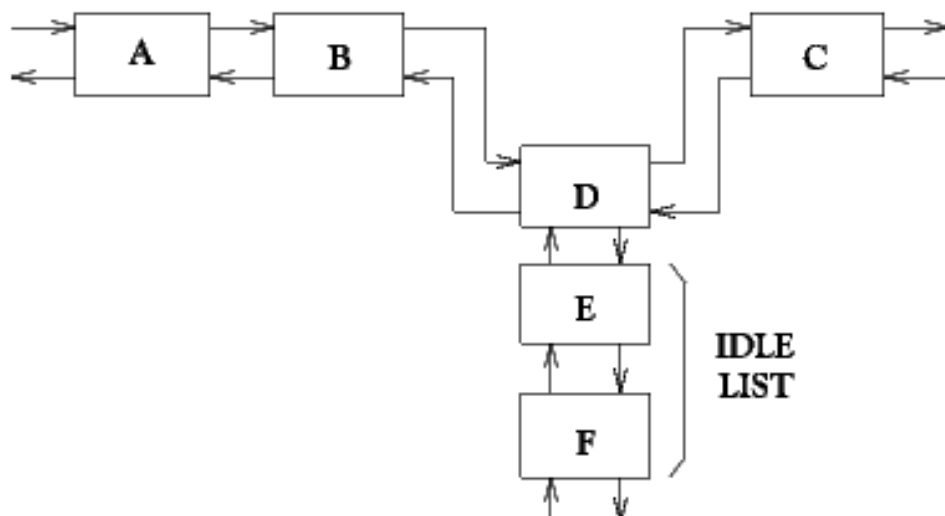
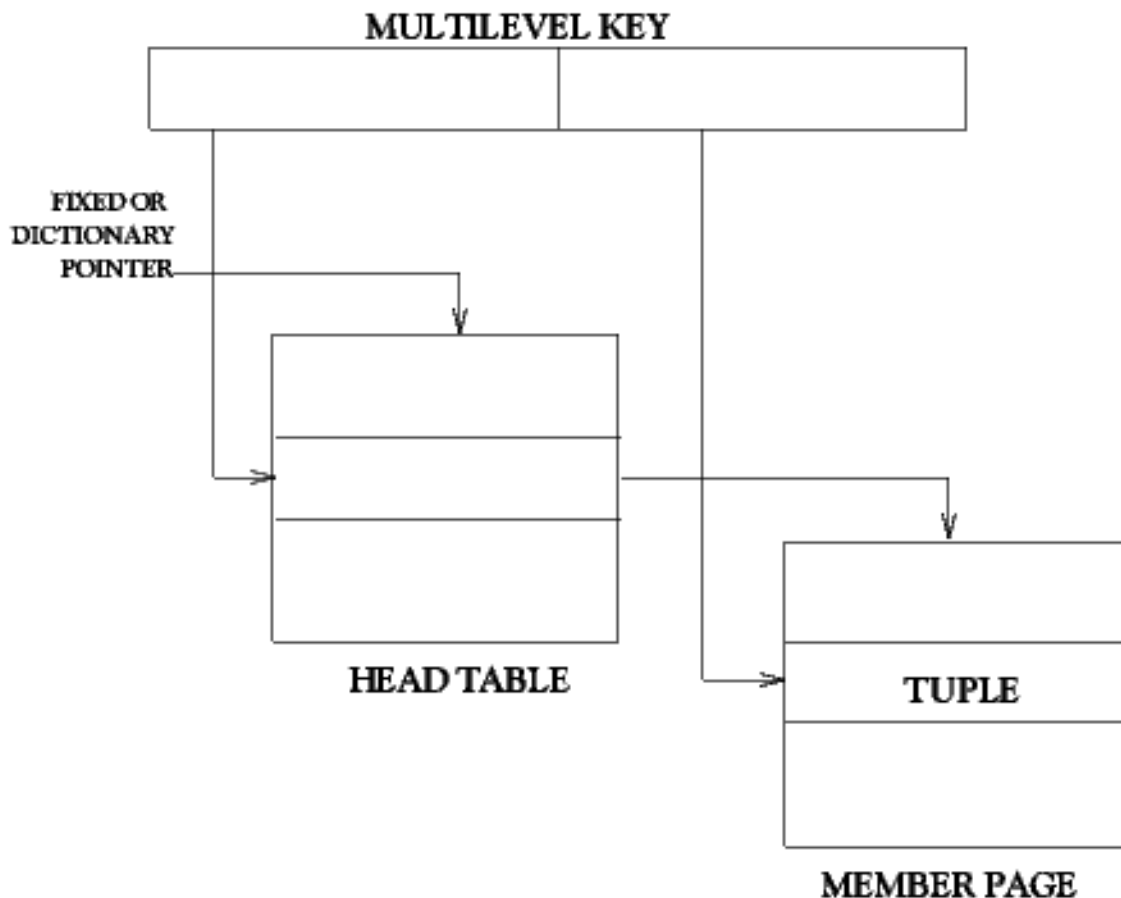
Part A**Part B****Part C**

Figure 2-14 Linking a Tuple from the Idle List

For relations that must grow when additional resources are added (for example, the relation containing circuit data blocks), a multilevel key is used to implement a link. Figure 2-15 shows how a multilevel key points to a tuple. The key is partitioned into high-order and low-order parts. The high-order part is used to look up a pointer in a *head table* (a table containing pointers to pages of memory). The head table used is fixed for dynamic access operations and is determined through a dictionary (a data structure that contains information required to locate other data structures) for normal database operations. The pointer obtained from the head table points to a *member page* in which the desired tuple is found by using the low-order part of the key as an index.

**Figure 2-15 Multilevel Key**

In practice, *linkage* (also called *mapping*) between tuples involves more than one link. The 5ESS® switch system has three basic forms of linkage: one-to-one linkage, owner-member linkage, and linkage using couplers.

2.2.5.1 ONE-TO-ONE LINKAGE

One-to-one linkage is the simplest form of linkage used in the system. Figure 2-16 illustrates how a one-to-one linkage is implemented. Each tuple has one link to the other tuple.

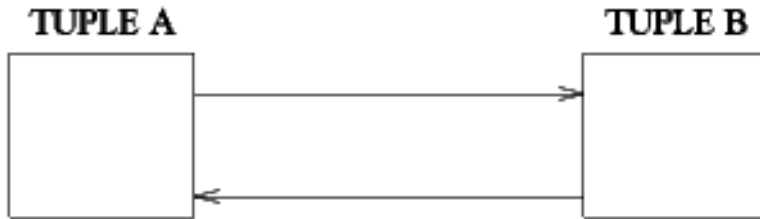


Figure 2-16 One-to-One Linkage

2.2.5.2 OWNER-MEMBER LINKAGE

Owner-member linkage is used to implement *one-to-many* associations between tuples. This includes as a special case one-to-one associations called *singletons*. This system forms a hierarchy of levels that has a tree-like structure. Figure 2-17 illustrates how an owner-member linkage is implemented. The tuple at the one end of the one-to-many association is called the *owner*; the tuples at the many end are called *members*. The linkage within the owner tuple (usually a head cell) is referred to as the *set-owner* linkage. The linkage at the member end is called the *set-member* linkage. The member tuples are joined in a doubly linked list through predecessor and successor links. The owner tuple has links to the head and tail of the member list. Each member tuple has a link to the owner tuple.

The figures that appear in the Audit messages of this document use a special notation to denote owner-member linkages. An owner-member linkage is represented as a line with a dot at one end (the owner end), and an asterisk at the other end (the member end).

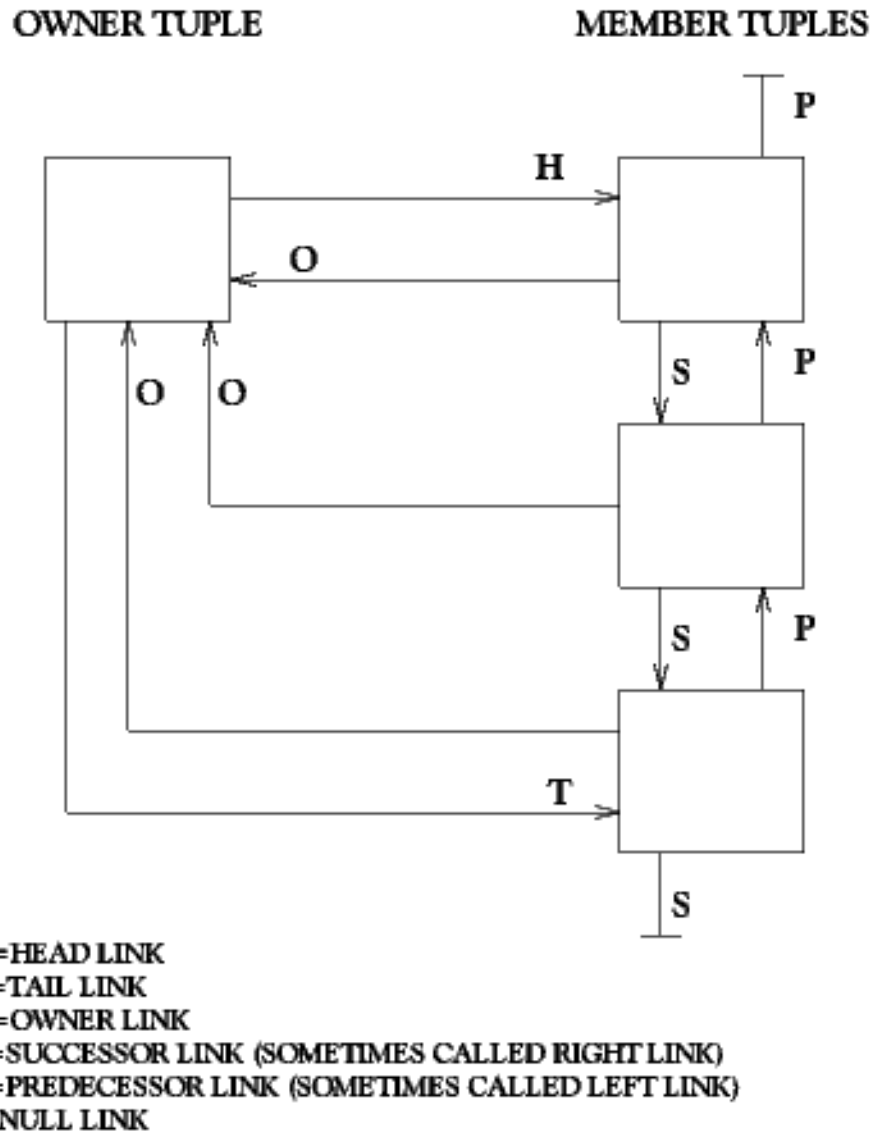


Figure 2-17 Owner-Member Linkage

2.2.5.3 COUPLERS

Couplers are special data structures that extend the basic owner-member linkage scheme to *many-to-many* associations between tuples. Figure 2-18 shows an example of a many-to-many association.

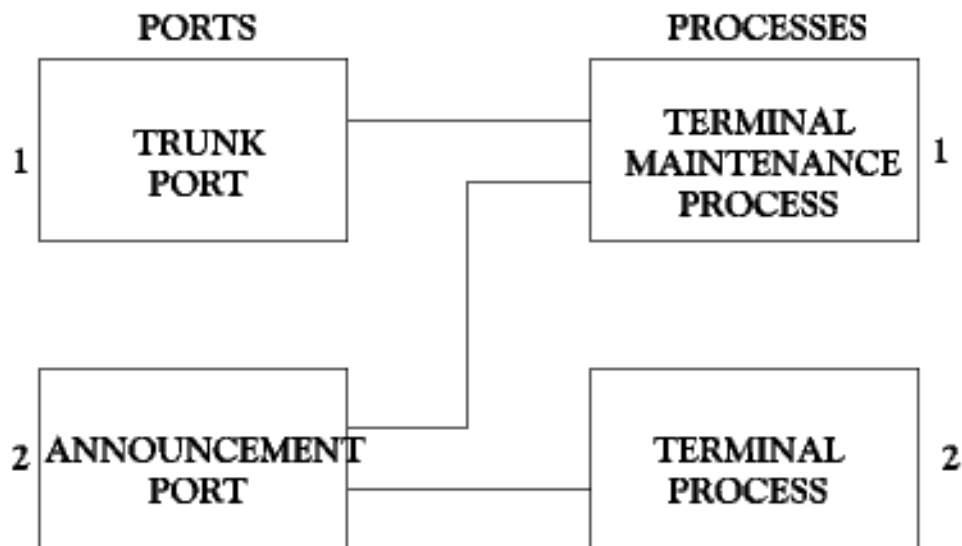


Figure 2-18 Many-to-Many Association

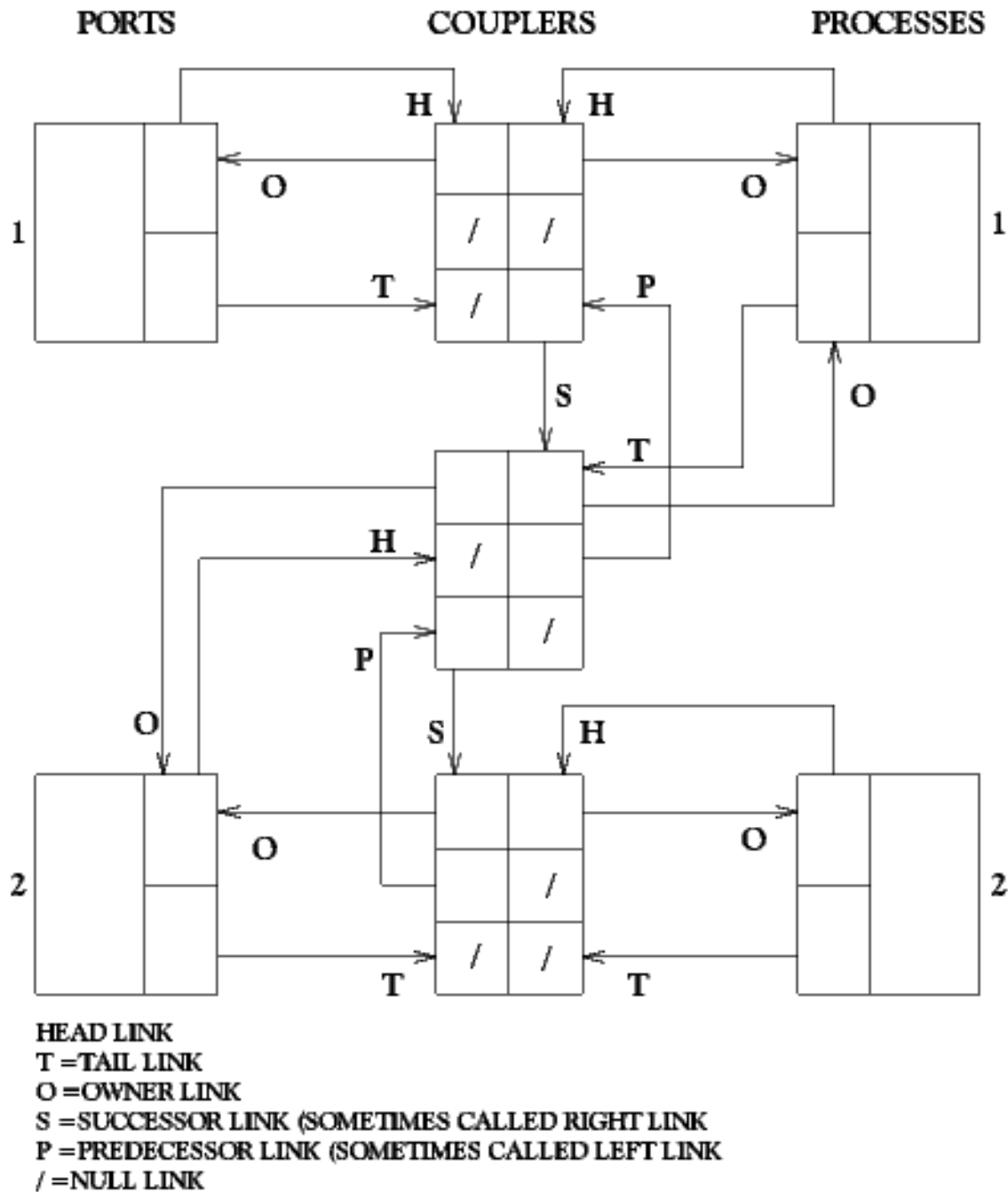


Figure 2-19 Use of Couplers

A terminal maintenance process is linked to both a trunk port and an announcement port while a terminal process is linked to the same announcement port. Figure 2-19 shows how three couplers are used to implement this many-to-many association. Each coupler consists of two sets of link items; each set may participate as a member in an owner-member linkage. Couplers A and B are linked using the righthand link set as members owned by process 1. Couplers B and C are linked using the lefthand link set as members owned by port 2. Coupler A is linked using the lefthand link set as a singleton to port 1. Coupler C is linked using the righthand link set as a singleton to process 2.

Couplers implement many-to-many associations between ports and processes, and between channel control blocks

and circuit data blocks.

2.2.5.4 MASTER/SLAVE LINKAGE

When a tuple is a member in more than one owner-member linkage, one linkage is called the *master linkage*; the others are called *slave linkages*. These rules apply to master and slave linkages:

- Master links to a tuple are never null. When the tuple is not owned (directly or indirectly) by a process, it is linked using the master linkage to a system or idle list and owned by a head cell.
- Slave links to a tuple are always null when the tuple is linked to a system or idle list owned by a head cell.
- Slave links cannot exist without a corresponding master link.

2.3 PROCESSES

There are many concurrent activities in the 5ESS[®] switch environment. Two activities are concurrent if they have both begun but have not yet terminated. Each activity involves the detection of and response to a series of events. An example is the activity resulting from the origination of a phone call. Events and responses in this activity include digit collection, ringing, and talking. Since many phone calls originate simultaneously, the resulting concurrent activities must be supported by the processor to achieve efficient use of switching resources. This is possible since each event in an activity typically needs a small amount of processing time compared to a much longer time between events.

The OSDS handles concurrent activities through programming entities called *processes*. The arrival of an event activates a process. The process then acts on the event and waits for the arrival of another event or terminates.

2.3.1 PROCESS IMPLEMENTATION

Figure 2-20 illustrates the implementation of processes. In general, each process consists of a program, a process control block, and a stack. In practice, however, more than one process may share a stack because processes that are not currently activated may temporarily relinquish their stacks to conserve memory.

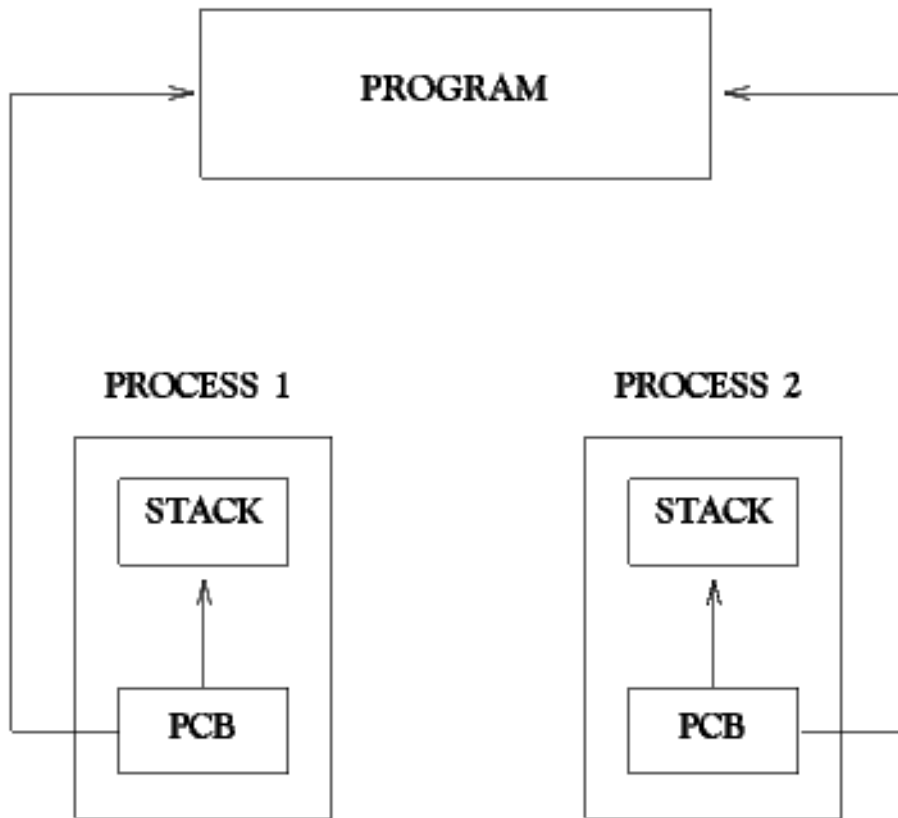


Figure 2-20 Process Implementation

The program detects and responds to events. It may be shared by several processes; only one copy of the program is required in memory for all the processes that use it (an analogy is several separate processors each fetching instructions from a common read-only memory). The program executes when control passes to the process; it may execute in its entirety or in segments. If the program executes in segments, a program counter in the process control block ensures that execution resumes from the point that was reached in the program when the process stopped.

C-language programs require a stack to implement function calls and returns, to pass parameters, and to allocate storage for local variables. Each process is given a stack when the process is created. The process control block contains an item (the stack control block) that indexes the stack. This arrangement allows processes to take real-time breaks within nested function calls.

The process control block stores all the data required to represent and administer the process, including:

- Process ID
- Process state (ready, running, waiting, and so forth)
- Process priority
- Associated program name
- Processor registers (including the program counter)
- Stack index
- Process linkage (for linking process control blocks into lists; refer to the process control block (PCB) and the

process control block linkage area (PCBLA) audit descriptions for details)

- Message linkage (for linking message control blocks to processes; refer to the message control block (MCB) audit description for details)
- Timer linkage (for linking timer control blocks to processes; refer to the timer control block (TCB) audit description for details).

2.3.2 PROCESS TYPES

The majority of processes used in the 5ESS[®] switch system are associated with the terminals. For each plain old telephone service (POTS) call, there are two *terminal processes*, one controlling each terminal (a call coordination process is not required). Each terminal process controls the associated physical terminal and thus keeps a partial view of the call.

Other processes provide services on a system-wide basis. These processes are called *system processes*. Examples are scanning processes that scan for peripheral inputs and sort them according to urgency, dispensing processes that distribute inputs to terminal processes, and routing processes that assist in locating the called terminals. System processes are generally long-lived. They are created and terminated independent of call processing and do not share their programs with other processes. Some system processes, which perform jobs such as peripheral input/output, are activated periodically on a timed basis (because of real-time requirements); others are activated on demand.

3. AUDITS OVERVIEW

3.1 INTRODUCTION

There are three different kinds of audits:

- **Application audits** are programs that verify the consistency of dynamic data used by 5ESS[®] switch application code.
- **UNIX[®] RTR audits** are programs that verify the consistency of data associated with the control unit (CU), equipment configuration database (ECD), file manager (FMGR), memory manager (MMGR), and other UNIX[®] RTR data in the administrative module (AM).

Refer to the Input Messages Manual, 235-600-700, and Output Messages Manual, 235-600-750, for more information about these audit-related messages.

- **Static data audits** are programs that verify the consistency of static office-dependent data (ODD).

3.2 APPLICATION AUDITS

All references to "audits" in this section refer to application audits only.

Application audits verify the consistency of dynamic data. The existence of an audit does not guarantee that the data structure it checks is consistent all the time, but it does mean that the audit eventually detects inconsistencies and attempts to recover from them.

By detecting and correcting data inconsistencies, audits help prevent problems, such as resource failures or data errors, which could cause call processing or other switch activity to fail or run in a degraded manner. By providing a recovery mechanism that can be invoked by asserts or single process purges, audits help prevent escalation to higher levels of initialization.

Generally, audits report all errors they detect to the read-only printer (ROP). The exceptions are due to brevity control restrictions and the printing status of audit message classes on the processor involved. (Refer to the Input Messages and Output Messages Manuals for more information on brevity control and message classes).

Audit error reports may be followed by one or more reports that include additional debugging data associated with the error being reported. When an audit finishes, an audit completed report is produced. The completed report states the total number of errors the audit found.

Audits do not take corrective action without reporting an error to the ROP, unless the audit is running as part of a selective initialization or in post-initialization mode.

Audit recovery may include requesting a single process purge or the scheduling of other *related* audits (audits of data structures associated with the data being recovered). A common event number is used for all ROP reports associated with the same stimulus. For example, an assert might request an audit that might, in turn, schedule other related audits. The assert reports and all audit reports would contain the same event number. Technicians can use this information when analyzing ROP output.

If an audit is unable to correct an error without human intervention, the audit sends a *manual action* error report to the ROP. Manual action error reports are marked with a capital A in the left margin of the first line of the report. The audit continues to find (and report) the same manual action error until the problem is corrected by office personnel.

3.2.1 ERROR REPORT DESCRIPTIONS

Each audit error report entry includes the following sections:

- **Error Description** explains what the error being reported is about.
- **Error Effect** describes what might happen if the error went uncorrected.
- **Corrective Action Taken** describes what the audit does to try to correct the error.
- **Manual Action Required** specifies the actions office personnel must take to recover from the error. It contains information only if the audit is unable to correct the error without human intervention: otherwise, if no manual action is needed, it says *None*.
- **Dump Description** describes the data included in the error report. Five fields are always included: *Error Address*, *Bad Data*, *Good Data*, *Logical Key*, and *Dump*. These fields contain information relevant to understanding the error being reported.

The field names do not always indicate the kind of data they contain. For example, the error address may not be an address, and bad data may not be *bad* data. Also, additional optional data dumps may be provided; if so, then *Dump* is listed as *Dump 1*, and succeeding optional data dumps are listed as *Dump 2*, *Dump 3*, and so on. The error description explains what each field contains for the particular error being reported.

All data dumps are unformatted and printed in hexadecimal.

3.2.2 ERROR REPORT FORMATS

There are two basic types of audit error reports. One for errors that the audit will attempt to recover itself (see the **Corrective Action Taken** field in the audit error report description):

```
AUD  SM=2  CR  ERROR-CODE=NONEPROC  EVENT=326
ERROR-ADDR=H'33c250  BAD-DATA=H'82026d
LOG-KEY=H'47  GOOD-DATA=H'0
```

The second is for manual action errors (see the **Manual Action Required** field in the error report description):

```
A  AUD  DAP=OKP  TKQUE  ERROR-CODE=PORT  EVENT=756
ERROR-ADDR=H'0  BAD-DATA=H'0
LOG-KEY=H'a10e0002  GOOD-DATA=H'0
```

Refer to the Output Messages manual, 235-600-750, for more complete descriptions of these and other audit output messages.

3.2.3 AUDIT ENVIRONMENTS

Application audits run in the following 5ESS[®] switch environments:

- OKP AM (operational kernel process in the administrative module)
- SMKP AM (switch maintenance kernel process in the administrative module)
- CMP (communications module processor)
- SM (switching module)
- PH (protocol handler)

- MH (message handler)
- QGP (quad gateway processor)
- PI (packet switching interface processor)

Refer to the APP:AUDITS appendix in the Appendixes section of the Output Messages manual, 235-600-750, for a list of audits that exist in each environment.

Most audits are *intraprocessor*; that is, they only check data within one of the environments. Intraprocessor audits with the same name may exist in more than one environment, but each instance is independent of the others. For example, the OKP AM, SMKP AM, and SM environments each have a process control block (PCB) audit that checks the same kind of data. However, each version is independent of the others and makes checks only for its own environment.

All other audits are *interprocessor*; that is, they check data between two of the environments (typically the CMP and SM environments with the CMP in control). In this case, the same audit name in different environments refers to different pieces of the same audit. There are only a few interprocessor audits.

3.2.4 AUDIT SCHEDULING

Audits run in different modes (refer to Table 3-1):

- **Routine audits** are scheduled automatically when the processor has no higher priority work. They are *segmented* (that is, they periodically give up control of the processor to let higher priority work be performed).
- **Elevated audits** run only by request. They may be scheduled by an input message (refer to section 3.2.5 of this chapter for more information), or by internal requests (from asserts, internal interfaces, single process purges, or for related recovery when another audit has detected an error).

Elevated audits run segmented and preempt execution of any routine audits.

- **Directed audits** are run only by an internal request from an assert, through an internal interface, or by escalation of a routine or elevated audit through an audit failure code.

Directed audits run *unsegmented*; that is, they do not give up control of the processor until they are finished, and preempt execution of both routine and elevated audits.

Directed audits are only used for system-critical resources. Since they run unsegmented, all other processing (including call processing) is blocked until they are finished.

- **Selective initialization, or post initialization mode audits**, are a fixed set (and sequence) of audits scheduled by system integrity during or immediately after a selective initialization to correct errors in dynamic data that is not being reinitialized. The audits are run unsegmented and do not generate any reports. This is the only exception to the rule that audits do not take any action without reporting an error.

Audits can be inhibited (prevented from running) by an input message. However, even when an audit is inhibited, it can be executed manually.

If an audit is scheduled through an input message, a completed report is always sent to the ROP. In all other cases, a report is sent only if the audit has detected an error. In particular, even if an assert schedules an audit, there is no evidence on the ROP that the audit was run if the audit did not detect any errors.

If an audit is scheduled by an internal request, all reports from that audit are associated with an event number

supplied by the requester. In particular, audits scheduled by asserts or single process purges have the same event number as the assert or single process purge.

In particular, this is true of *related audits*. When an audit completes, if it has found errors, a set of audits of data related to the data just recovered is scheduled (for example the related structure may have a linkage to the structure just recovered). Related audits run elevated and use the same event number as the original audit. Since related audits are *not* scheduled when an audit completes with no errors, this mechanism causes recovery to ripple out from the original stimulus and die out when recovery actions have been successful.

Table 3-1 Summary of Audit Scheduling Mechanisms

Scheduling Source	Routine Audits	Elevated Audits	Directed Audits
Input Messages	No	Yes	No
Asserts	No	Yes	Yes
Single Process Purges	No	Yes	No
Related Audits	No	Yes	No
Internal Interfaces	No	Yes	Yes
Audit Failure Codes	No	No	Yes
Automatic Scheduling	Yes	No	No

The following table summarizes the effect of each type of audit scheduling on other switch activity:

Table 3-2 Summary of Audit Scheduling Effects

Routine Audits	Elevated Audits	Directed Audits	Selective Initialization
None.	Minimal background delays.	Disables call processing until the audit completes.	Helps preserve stable calls and maintenance states through a selective initialization.

3.2.5 INPUT MESSAGE INFORMATION

The Audit Input Message Summary Table lists the input messages available for application audits in each environment.

Table 3-3 Audit Input Messages Summary

Input Message	Operation
AUD:{audit}, SM=# AUD:{audit}, cmp={0 1} AUD:{audit}, env={OKP SMKP}	Run an audit
INH:AUD={audit}, SM=# INH:AUD={audit}, CMP=# INH:AUD={audit}, ENV={OKP SMKP}	Inhibit an audit ^a
ALW:AUD={audit}, SM=# ALW:AUD={audit}, CMP=# ALW:AUD={audit}, ENV={OKP SMKP}	Allow an audit ^a
STP:AUD, SM=# STP:AUD, CMP=# STP:AUD, ENV={OKP SMKP}	Stop execution ^b
Notes: a. "ALL" can be substituted for "audit". This will inhibit or allow all audits in the processor with one input message. b. Will only stop execution of the currently running audit. If no audit is currently running, the input message has no impact.	

3.2.6 AUDIT ERROR REPORT ANALYSIS

Occasionally, an audit detects (and reports) an error that requires human intervention to be corrected. Until office personnel has taken the required corrective action, the error continues to be re-detected and re-reported. Typically, the intervention needed is a change to the ODD.

Audit error reports for errors that are known to require human intervention, or manual action, for correction are indicated by a capital A in the left margin of the first line of the report. By examining the audit report(s), ODD population rules, and the ODD itself, office personnel can determine exactly what should be done to correct the inconsistency.

Other audit reported errors (that is, those not marked with the A) are almost always corrected by the audit itself. In cases where the audit is *not* able to correct the error (or the error is repeatedly reintroduced), the same audit(s) reports the same error(s) on the same data across multiple events or multiple audit invocations. Such repetitions should not be confused with either of the following:

- single events that include more than one instance of a particular error report or set of error reports
- multiple appearances of the same audit reporting different errors in each appearance, or reporting the same errors on different data each time

As long as reports associated with an event eventually stop appearing and do not recur, recovery has been taken automatically. Technicians should refer questions or concerns about particular sets of audit error reports to their next line of support.

3.3 STATIC DATA AUDITS

The static ODD (SODD) audit verifies that the data stored in the base relations of the ODD conform to the database population rules. Execution of the audit is initiated by a manual input message. The audit can also be invoked directly to verify a single database component. Data errors detected by the audit are logged in files. A manual input message is required to display the contents of an error file in a readable form. Data errors are not automatically corrected.

The EXC:AUD-SODD input message invokes the SODD audit. The audit is activated either at the time of the input message, or at the time specified in its schedule parameter block. Once activated, the execution of the audit proceeds until the allocated duration has elapsed. The audit process is then suspended and remains dormant until the input message is re-executed. The audit process then resumes from the point at which it was suspended. The normal operational scope of the audit is to run all available components.

The SODD audit is executed in a cyclic mode. During a single cycle, all the relations within the scope of the audit are examined. When the end of the cycle is reached, an audit completion report is printed and a new cycle is started.

In a typical scenario, the audit process is activated periodically by a scheduled input message, and during each period of activity, the execution of the audit cycle is continued from the previous point of suspension. The duration of each of the active periods is specified by the originating input message, and the times of day at which the periods start are specified in a (standard) schedule parameter block entered with the input message.

Controlled by a process operating in the AM, the audit is applied to the data bases that are currently active in the AM and the SMs. Unless restrictions are imposed, all tuples and parameters for which audit products are available will be audited with respect to the applicable population rules.

When the audit is applied to a partitioned base relation that is distributed among the AM and the SMs (or among the SMs), the required audit processes are executed concurrently on the AM. One instance of a redundant relation is separately audited with respect to the population rules, and the consistency of the redundant data is not verified. During the auditing of a relation, the integrity of the data involved is not influenced by RC/V activity.

The AM collects and logs error data. For each 24-hour period, two error logs are maintained: one for the current audit cycle, and the other for the previous cycle. The erroneous data is not corrected.

When a specific component of the audit is requested manually, the request is queued behind any previous requests. The subsequent processing of the manual request queue depends on the situation prevailing at the time. If the

SODD audit is active (and there is sufficient time remaining), manually requested audits are executed as soon as the current audit function has finished. If the SODD audit is inactive, manually requested audits are executed with the highest priority when the audit activity is resumed.

After an audit has been executed by manual request, it will not be executed again in the current cycle. When made in the preemptive mode, a manual request stops the active audit (provided that it was not requested manually) and invokes the requested audit function immediately. The preemptive mode also enables a manually requested audit to execute when the entire SODD audit is inhibited.

3.4 OPERATIONAL ASPECTS

To execute a complete audit cycle, an interval of audit activity is initiated periodically by a scheduled input message. The duration of the interval is determined by an input message parameter, and the starting time is specified by a schedule parameter block attached to the input message. Note that a typical audit cycle requires several weeks to complete, depending on the size of the office. There is only a logfile for the current cycle and the previous cycle. Therefore, the technician must check the logfile each cycle. When there are errors, the technician must store the logfile before it is overwritten.

The GImaxauds parameter controls a number of allowed concurrent audit processes (one per processor). The concurrent processors are different processors for the same relation component. The valid range of the parameter is 1 - 3, and the default value, which is 3, can not be changed with RC/V. This parameter's value is checked when the SODD control process is restarted by user level automatic restart process (ULARP).

A report is printed when execution of an audit cycle or a manually requested audit component has completed, or when auditing is suspended at the end of an interval. This report also appears when the execution of the audit is stopped manually. Suspend and resume reports occur when inhibit and allow messages (without COMP) are processed.

Note that the SODD audit has no error correction mechanism. Errors detected by the audit must be corrected by hand, using RC/V procedures.

Table 3-4 shows what the SODD audit does with MML messages in various circumstances.

Table 3-4 SODD Audit Responses to Inputs

Input	State	Action
general INH	preemptive audit running	set flag to stop further dispatches
general INH	all other states	set flag, kill any running product
specific INH	this component running	set component flag, kill running
specific INH	all other states	set component flag
general EXEC	DUR < current end time	NG ack
general EXEC	DUR > current end time	reset duration
general EXEC	general INH	reset duration, send warning
specific EXEC	expired duration	queue component, send warning
specific EXEC preempt	manual audit running	queue component, send warning
specific EXEC preempt	nonmanual audit running	queue component, kill running
specific EXEC nonpreempt	an audit running	queue behind previous manual audits
specific EXEC nonpreempt	general INH	queue component, send warning
specific EXEC preempt	general INH	run component
specific EXEC nonpreempt	specific INH of same	NG ack
specific EXEC preempt	specific INH of same	NG ack
general STP	is a current duration	kill duration, kill running audit
general STP	no current duration	NG ack
specific STP	this component running now	kill and requeue ^a , "killed" msg
specific STP	not running now + queued as manual	requeue ^a , "cancelled" msg
specific STP	not running now + not queued as manual	NG ack

Notes:

- a. Requeue using this algorithm: if not run this cycle, requeue as part of a full audit; else, if run successful once or run more than once, unqueue; else, requeue at tail.

In the case of CP-ONLY and REDUNDANT relations, the log shows the completion of processors 0 and 193 (193 means redundant). Any error messages coming from the execution of these processors show processor -1 (the database manager uses -1 to mean any processor).

Specifying PROC on a redundant component has *no* effect on the behavior of the component audit. The input messages give an OK acknowledgement, but have no effect.

4. Using Audits

4.1 INTRODUCTION

The audit system detects, isolates, and corrects errors in the data structures involved in 5ESS[®] switch software. It corrects broken linkages, restores lost data resources, and helps identify the location of problems in the source code. The more that is known about the structure of the 5ESS[®] switch software, particularly of the modified relational database, the more effectively audits can be used for troubleshooting. When used for maintenance work in conjunction with other diagnostic tools (such as recent change check programs and UNIX[®] diagnostic reports), audit reports, along with other reports, can help the switch personnel reconstruct the series of events that caused the problem.

The audit system begins cycling through the audits when the system software is initialized. When there is a heavy call processing load, however, the audit cycle will slow down. The operator can also execute a given audit, causing it to run sooner than it would have as part of the cycle. To execute an audit, the operator types an input message requesting the audit. When the audit completes, a summary report is printed. If any errors are discovered, reports describing the errors are also printed. Error reports and summary reports are also generated when the audit runs as part of the cycle. These reports may be printed to the ROP or directed to a logging file. Unlike the reports printed due to operator requests, reports for audits executing as part of the cycle are only generated when the audit discovers errors. If manual actions are required to correct an error, the report is printed unconditionally to the ROP with an "A" in the priority of action field.

4.2 UNIX[®] RTR Audits

4.2.1 Purpose

This section explains which input messages are used to query and request software integrity services using UNIX[®] real-time reliable (RTR) system audits. UNIX[®] RTR audits are a part of the system integrity subsystem which is a collection of programs, processes, commands, database records, and performance records that work together to ensure the reliability of the operating system software.

4.2.2 Types of UNIX[®] RTR Audits

There are currently seven input messages that give the user the following audit capabilities.

- Allow the routine execution of audits (ALW:AUD).
- Request audits (AUD). There are twelve types of audits that can be requested.

CUMEM	= The control unit memory comparison audit.
CUSTAT	= The control unit status audit.
ECD	= The equipment configuration database audit.
ECDOWN	= The equipment configuration database owner audit.
FMGR	= The file manager audit.
FSBLK	= The file system block audit.
FSCMPT	= The file system compaction audit.
FSLINK	= The file system link audit.

MMGR	= The memory manager audit.
MSGBUF	= The message buffer audit.
PMS	= The plant measurements system database audit.
PROAD	= The process administration audit.

For more information on these audits, refer to the Audit Descriptions section in the Audits manual (235-600-400) and the input message pages in the Input Messages manual (235-600-700).

- Inhibit the routine execution of audits (INH:AUD).
- Determine the status of audits (OP:AUD).
- Extract error information for audits (OP:AUDERR).
- Analyze data produced by file system audits (OP:FNAME).
- Stop audits from execution (STOP:AUD)..

For information on the use of these input messages, refer to the Input Messages manual (235-600-700).

4.3 SODD Audits

4.3.1 Purpose

The SODD audit is used to audit and detect errors in the static office-depended database (SODD). It verifies relations, performs cross checks of attributes, and reports findings to switch personnel for resolution. For more information about SODD audits, refer to the SODD Audits manual (236-600-410).

4.4 APPLICATION AUDIT ANALYSIS

4.4.1 Purpose

Application audits verify the consistency of dynamic data. In all cases, the goal is to prevent or to help recover from problems that can cause switch functionality to fail or to run in a degraded manner.

4.4.2 Types of Audit Error Reports

There are two basic types of audit error reports.

- Errors that the audit will attempt to recover.
- Errors that require manual action.

Occasionally, in the course of reading static data needed to do a check, an audit will detect static data inconsistency (for example, between two static relations). In that case the audit will report the error using a "manual action" error code. The error report is printed with an 'A' in the left column of the first line to indicate that action is needed by switch personnel. The description in the Audits Manual provides specific information on what data is in error. These errors will be reported every time the audit runs until the static data is manually corrected.

4.4.3 Types of Data Audited

Not every piece of dynamic data is checked by an audit. Criteria for auditing include:

- What data errors are possible.
- What the effect of an error is, including the severity and scope.
- What other mechanisms might detect and/or correct the error.
- If the condition can be detected by an audit.
- If the condition can be corrected by an audit.

The types of data checked will often determine the audit name. For example, the PCB audit checks process control blocks. The PORTLA audit checks tuples of the RLPORTLA relation.

NOTE: Because data linkages are audited from both directions and sometimes other consistency checks are appropriate, multiple audits may reference the same pieces of data.

4.4.4 Types of Audit Checks

Audit checks can be categorized by type:

- Linkage consistency.
- Data key validation.
- Resource ownership and availability.
- Dynamic vs. static data consistency.
- Hardware vs. software consistency.
- Other data consistency.
- Data configuration consistency (such as "parent/child" state consistency).
- Stuck in a transient condition.
- Quarantine (recover an error detected elsewhere).

To determine the type of error, switch personnel must refer to the audit description and should not make assumptions based on the error code used in the report.

4.4.5 Audit Drivers

An audit driver is a set of code shared by multiple audits which controls segment breaks, performs syntactic checks, schedules semantic checks, and handles errors. Because the code is shared, a common set of error codes can be output by many different audits. If an audit calls a driver, the following message is included in the audit description:

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

4.4.6 Analyzing Audit Reports

4.4.6.1 Procedure for Analysis

When attempting to reconstruct a scenario that includes audit reports, switch personnel should consider all available information about known software or hardware problems, activity on the switch when the problem first appeared (such as recent change, growth or degrowth procedures, diagnostics, software update applications and so on).

This discussion should be considered a starting point. It is not (and could not be) exhaustive. The most effective way to learn to debug is to practice. Refer to the "Analysis Examples" section for more detailed examples.

Analysis of audit reports generally includes the following steps:

- (1) Understanding the meaning of the report.
- (2) Determining the source of the audit report.
- (3) Placing the report in context.

4.4.6.1.1 Understanding the meaning of the report

This can be done by looking up the error description in the Audits Manual. Find the section covering the audit reporting the error, and search the alphabetical list of error code descriptions for the one that was reported.

NOTE: In some cases, the same error report is used in the same way by multiple audits and so is described in a separate section of the manual (refer to the "Audit Drivers" section). In that case, the audit sections will each have a reference to the shared section.

4.4.6.1.1.1 Extracting data from the report

Sometimes the specific reason for the report may depend on data included in the report. The "Extracting Data From An Audit Report" section describes how to extract such data from audit reports.

4.4.6.1.2 Determining the Source of the Audit Report

When looking at audit reports, switch personnel should first try to determine the source of the audit. One thing that can help in this is the audit message class. Audit message classes include:

AUDT = Audits run due to a manual request.

AUDTMON = Audits scheduled as part of an ongoing event.

AUDTFST = Audits error reports which are the start of a new event.

Examples:

- (1) If the message class of the report is **AUDTMON** the audit must be coming out as a side effect of something else such as an assert, single process purge or other audit.
- (2) If the message class is **AUDTFST**, the audit is the start of a new event. However, even here it is possible that the audit is coming out due to a problem that was also detected by an assert (which did not happen to schedule the audit) or due to other audit recovery.

4.4.6.1.3 Placing the report in context

Often, audit error reports are just part of a larger set of reports. Using the data provided in the audit reports to reconstruct the relationships between the various data structures before and after each audit error can be very useful.

The following sections discuss some common scenarios.

4.4.6.1.3.1 Audits caused by an SPP or Assert

An assert may fire and schedule an audit which finds errors. As part of its recovery, the audit may purge a process. The SPP code may then schedule additional audits and so on.

The reports from all of these actions are typically tied together by a single event number (the exceptions are due to multiple triggers for the same audit and missing related audit specifications). For various reasons (such as differing message classes and priorities), messages do not always print in the same order as the events that triggered them occurred. Therefore it is important to try to reconstruct the sequence of events on the switch that resulted in the reports that were generated.

Example:

The audit report in question is an SMXRF audit DEADPID error and the dead process found in the rSMXRF tuple was purged in an earlier event. In that case it is safe to conclude that the audit error was a side effect of the SPP and further investigation should focus on the SPP itself.

4.4.6.1.3.2 Recurring Audits

Sometimes, the same set of audits may continue to report errors over and over. There are many possible causes for this. There may be static ODD inconsistencies causing two or more audits to view the same data differently causing them to "roll" back and forth, each undoing the recovery the other has done. Inconsistencies between the database dictionaries and dynamic access relation headtables can also cause this kind of symptom.

4.4.6.1.3.3 Sporadic Recurring Audits

In other scenarios the same audit may report errors sporadically but consistently over time. These cases can be difficult to tie to any specific event. Here again there are many possible causes including application code problems in little used legs of code, transient or "race" conditions between the audit and application code or even wild writes from some unrelated piece of code. If a "race condition" is suspected, one option is to inhibit the audit in question for some time and then request it manually. If the number of errors it then finds indicates that they have been accumulating, there probably isn't a race condition. If it finds few or no errors it is possible that a race condition exists.

4.4.6.1.3.4 Recurring Errors

It is also possible that the same error is being reintroduced over and over. If the appearance of the error can be tied to some other event (e.g. diagnostics, an assert, a hardware failure) it is possible that that other event is the root cause. However, sometimes a correlation is coincidental and not due to cause and effect.

4.4.6.1.3.5 Non-recurring Errors

Sometimes an audit report will appear that cannot be tied to another event (such as an assert, SPP, or other audit recovery) and does not reoccur. Even if it is not possible to determine the root cause, switch personnel can still use the Audits Manual to understand the meaning of the report and keep that information in mind when investigating other problems.

4.4.6.2 Analysis Aids

4.4.6.2.1 Extracting Data From An Audit Report

Audit reports include three different kinds of reports: the error report, optional data dump reports, and a completion report. The data dump reports provide unformatted data (such as the contents of a relation tuple) that may be needed for debugging. Depending on the software release, data type, and other factors, the dump report may or may not include a label for the data being dumped. However, the error code description in the Audits Manual (235-600-400) will specify the data provided by each dump report.

To interpret the dump, it is first necessary to get a picture of the data layout. This information is available in the Translations and Dynamic Data Reference Manual (235-600-2xx).

For example, here is an rlCHDB tuple dump:

```
AUD SM=7 CHDB DUMP EVENT=198
KEY=rlCHDB TUPLE WHERE
    <the_key>=
    <H'14b>
BLOCK-ADDR=H'4c904c
014B0000 00000000 00000000 00000000 00070000 00000000 00000000 00010000
00000000 00000000 00000000 00000001 01000000 00001000 00000000 00000000
00000000 00000000 00000000 00000000 00E00000 00001000 00000000 11050000
```

Note that the "BLOCK-ADDR" value is the beginning address of the data being dumped.

From the manual, the layout of an rlCHDB tuple looks like:

struct rlCHDB	rlCHDB
(a) uint	rlCHDB.chnum
(b) int	rlCHDB.offrange
(c) uint	rlCHDB.origtime
(d) int	rlCHDB.timrown
(e) int	rlCHDB.timrfwd
(f) int	rlCHDB.timrback
(g) uint	rlCHDB.tmstamp
(h) uint	rlCHDB.puls_time
(i) DMTONE (enum)	rlCHDB.tone
(j) int	rlCHDB.gf_tmrown
(k) int	rlCHDB.gf_tmrfwd
(l) int	rlCHDB.gf_tmrback
(m) uint	rlCHDB.bg_fwd_tmr
(n) uint	rlCHDB.bg_bwd_tmr
(o) uint	rlCHDB.bg_cyc_cnt
(p) uint	rlCHDB.fac_ckt
(q) uint	rlCHDB.protocol : 10
(r) DMLIST (enum)	rlCHDB.pc_list : 6
(s) char	rlCHDB.dig_buff [3]
(t) char	rlCHDB.ll_ag
(u) char	rlCHDB.rcv_mask
(v) char	rlCHDB.rep_ag
(w) char	rlCHDB.off_ag
(x) char	rlCHDB.scnmsk
(y) char	rlCHDB.onmintime
(z) char	rlCHDB.onrange
(A) char	rlCHDB.offmintime
(B) char	rlCHDB.digcnt
(C) char	rlCHDB.reptproc

(D) char	rlCHDB.cnst
(E) char	rlCHDB.acspt
(F) char	rlCHDB.timrtype
(G) char	rlCHDB.puls_st
(H) char	rlCHDB.fg_det_state
(I) char	rlCHDB.sndinvmsk
(J) char	rlCHDB.rcvinvmsk
(K) char	rlCHDB.gf_lr_ag
(L) char	rlCHDB.gf_tmrval
(M) char	rlCHDB.gf_tmrtype
(N) DMINPROC (enum)	rlCHDB.inproc : 7
(O) DMBOOL (enum)	rlCHDB.reseizflag : 1
(P) DMINPROC (enum)	rlCHDB.saveinproc : 7
(Q) uint	rlCHDB.save_ag : 7
(R) DMPCSUPV (enum)	rlCHDB.supvchg : 2
(S) uint	rlCHDB.ccgnum : 7
(T) ulong	rlCHDB.comp_loss : 6
(U) DMAUQUAR (enum)	rlCHDB.quarstate : 3
(V) uint	rlCHDB.snd_mask : 4
(W) uint	rlCHDB.gls_v : 4
(X) uint	rlCHDB.rq_tag : 4
(Y) uint	rlCHDB.rq_new_tag : 4
(Z) DMGRPTYPE (enum)	rlCHDB.trk_dir : 3
(a) uint	rlCHDB.ireseizflag : 2
(b) DMBOOL (enum)	rlCHDB.ccstrk : 1
(c) DMBOOL (enum)	rlCHDB.brgch : 1
(d) DMBOOL (enum)	rlCHDB.dcstrk : 1
(e) DMBOOL (enum)	rlCHDB.metallic : 1
(f) DMBOOL (enum)	rlCHDB.mctqing : 1
(g) DMBOOL (enum)	rlCHDB.msg_lost : 1
(h) DMBOOL (enum)	rlCHDB.osps_call : 1
(i) DMBOOL (enum)	rlCHDB.opcat : 1
(j) DMBOOL (enum)	rlCHDB.bit_inv : 1
(k) DMBOOL (enum)	rlCHDB.bit_mskng : 1
(l) DMBOOL (enum)	rlCHDB.mm_active : 1
(m) DMBOOL (enum)	rlCHDB.met_trk : 1
(n) DMBOOL (enum)	rlCHDB.pc_flag : 1
(o) DMBOOL (enum)	rlCHDB.gf_active : 1
(p) DMBOOL (enum)	rlCHDB.ack_alw : 1
(q) DMBOOL (enum)	rlCHDB.prg_alw : 1
(r) DMBOOL (enum)	rlCHDB.met_ans : 1
(s) DMBOOL (enum)	rlCHDB.pol_req : 1
(t) DMBOOL (enum)	rlCHDB.tsi_ram : 1
(u) uint	rlCHDB.rep_f : 1
(v) uint	rlCHDB.ll_f : 1
(w) uint	rlCHDB.outg_f : 1
(x) ulong	rlCHDB.wrkarea
(y) long	rlCHDB.glsprot
DMSETOWN (struct)	rlCHDB.chmmo
(z) uint	rlCHDB.chmmo.st_headk
(A) uint	rlCHDB.chmmo.st_tailk
DMSETMEM (struct)	rlCHDB.chubm
(B) uint	rlCHDB.chubm.st_ownk
(C) uint	rlCHDB.chubm.st_fork

(D)	uint	rlCHDB.chubm.st_back
(E)	uint	rlCHDB.chubm.st_link
	DMSETMEM (struct)	rlCHDB.chccm
(F)	uint	rlCHDB.chccm.st_ownk
(G)	uint	rlCHDB.chccm.st_fork
(H)	uint	rlCHDB.chccm.st_back
(I)	uint	rlCHDB.chccm.st_link

		3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0								
		1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0				

0		chnum																	offrange																		

4		origtime																	timrown																		

8		timrfwd																	timrback																		

12		tmstamp																	puls_time																		

16		tone																	gf_tmrown																		

20		gf_tmr fwd																	gf_tmrback																		

24		bg_fwd_tmr																	bg_bwd_tmr																		

28		bg_cyc_cnt																	fac_ckt																		

32		protocol									pc_list									dig_buff																	

36											ll_ag									rcv_mask									rep_ag								

40		off_ag									scnmsk									onmintime									orange								

44		offmintime									digcnt									reptproc									cnst								

48		acspt									timrtype									puls_st									fg_det_state								

52		sndinvmsk									rcvinvmask									gf_lr_ag									gf_tmrval								

56		gf_tmrtype		inproc		saveinproc		save_ag		R		

60		ccgnum		comp_loss		U		V		gls_v		

64		Z		a		b		c		d		

68		wrkarea										

72		glsprot										

76		st_headk					st_tailk					

80		st_ownk					st_fork					

84		st_back					st_link					

88		st_ownk					st_fork					

92		st_back					st_link					

With this information, we can find the value of any attribute in this rCHDB tuple. For example, the \lftone\lP field is 16 bits long beginning at offset 16 and has a value of 7.

4.4.6.2.2 Physical Vs. Logical Key Values

The relational database used in the administrative module (AM), switching module (SM) and communications module processor (CMP) environments stores the keys of dynamic relations in two different ways. In the key attributes of the relation, a *logical* value is stored. In the linkage attributes, the *physical* value is stored. The two values are related by a constant offset value defined for the relation:

$$\text{Logical Key} = \text{Physical Key} - \text{Offset}$$

Physical keys are non-negative (0 and greater) values. Logical keys may be either positive or negative. A negative logical key is used to indicate that the tuple has been reserved to act as a headcell instead of a data tuple.

4.4.7 Manually Scheduling an Audit

4.4.7.1 When an Input Message is Used

Although audits run continually, the operator may occasionally want to run specific audits to help locate a problem in either the software or the hardware.

- If a problem is suspected with a particular data structure, the audits associated with the data structure should be run to obtain data dumps that may be analyzed in detail to isolate the fault.

- If a hardware problem is suspected, an audit report may be useful for identifying the fault.

Using audits to discover hardware and software problems is a skill. Proficiency comes with practice and a knowledge of the overall software system. In general, however, the user should be alert to problems in the system revealed by recurring audit reports and assert messages.

To run specific audits, the operator enters an audit input message. The operator may have to turn off brevity controls and logging before running the audit to see all the audit output. Brevity may be turned off for a message class (for example, AUDTFST) or for a processor. The operator should not leave brevity off for too long or the logging file may become overloaded.

Refer to the APP:COMMAND-GRP appendix in the Appendixes section of the Input Messages manual (235-600-700) for a list of all input messages associated with the AUDIT command group.

4.4.7.2 System Response to Input Message

If all the information is entered correctly and the system is fully operational, the system acknowledges requests for specific audits with OK or PF (printout follows). If there is a problem, a response of NG (no good) plus a short explanation prints. The system response typically prints within five or ten seconds. For more information on system response to requests for specific audits, refer to the APP:AUD appendix in the Appendixes section of the Input Messages manual (235-600-700).

4.4.8 Analysis Examples

Several example analyses are given in this section. Although these examples may refer to real audits and real error codes, they may not reflect the current code or documentation for a given release. They are meant to illustrate scenarios and troubleshooting techniques. Switch personnel should use the current version of the Audits Manual (235-600-400) when attempting to analyze audit reports.

NOTE: In the following examples, H' refers to a hexadecimal number.

4.4.8.1 Case History One: Manual Action Error

In this example, the DD audit repeatedly prints the following error report:

```
A AUD ENV=OKP DD ERROR_CODE=MODATT EVENT=756
  ERROR-ADDR=H'0125a324 BAD-DATA=H'15
  LOG-KEY=H'32          GOOD-DATA=H'32
```

The 'A' in the left margin indicates that this error requires manual action for recovery. This explains why it is reported again and again. It will come out every time the audit runs until the static data is fixed.

The Audits Manual (235-600-400) entry for the DD audit, error code MODATT is reproduced here for reference.

Table 4-1 Error Code: MODATT

Error Code:	MODATT
Error Description:	A remote switching module (RSM) from static relation RLHSMRSM (rLHSMRSM.rsmnum[index]) has no corresponding tuple in the RLMODATT relation. That is, a database read failed for relation RLMODATT using rLMODATT.module = rLHSMRSM.rsmnum[index] where the RSM number was valid (!= 0). >
Possible Error Effect:	The RSM can not be made operational.
Corrective Action Taken:	None.
Manual Action Required:	Either RLHSMRSM is corrupt (it contains an incorrect RSM number) or the RSM has not been fully specified in the static ODD. Technicians should determine if

	the RSM number corresponds to a real SM and either remove the reference from RLHSMRSM or add all missing static ODD via recent change or ODBE. See poprules for RLMODATT and RLHSMRSM for more information.
Dump Description:	
<i>Error Address</i> =	Index into rHSMRSM.rsmnum[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The host SM (HSM) number. This is the key to the rHSMRSM tuple (rHSMRSM.hsmnum).
<i>Logical Index</i> =	The RSM number. This is the key to the rIMODATT tuple (rIMODATT.module) for which the database read failed.
<i>Dump</i> =	The RLHSMRSM tuple.

Looking at the audit description in the Audits manual and the data from the error report, the problem can be identified:

- (1) The static RLHSMRSM relation indicates that H'15 is a valid remote switching module (RSM) number for host switching module (HSM) H'32.
- (2) The rIMODATT tuple where the key attribute *module* = H'15 does not exist according to the database manager.
- (3) The static RLHSMRSM and RLMODATT relations are inconsistent. Either SM H'15 is a valid RSM and the rIMODATT tuple for it must be inserted, or it is not a valid RSM number and the rHSMRSM tuple keyed by *hsm* = H'32 must be updated to reflect that.

4.4.8.2 Case History Two: Audits Scheduled from Other Events

At times, audits are scheduled to run elevated as part of other events such as single process purges (SPPs) or asserts. Code attempting to use dynamic data may discover a problem and schedule the appropriate audit to clean up the data. In such cases, analysis should start with the assert.

4.4.8.2.1 Audit scheduled from an assert

In the following example, the first block to appear is from the PCBLA audit but the message class is AUDTMON, indicating that the audit is part of an ongoing event. Looking further down the output we see that the PCBLA audit was called to run (or "scheduled") from an assert. The audit report prints before the assert because the audit message class is a higher priority than the assert message class. This illustrates the importance of gathering all the reports related to an event before beginning the troubleshooting process.

In general, it is an excellent plan to look first at the assert to analyze the problem. Refer to the Assert Manual (235-600-500) for more information on asserts. The audit error reports a QUAR error in the rIPCBLA tuple with *Bad Data* = H'1. However, this particular assert fires due to an inability to follow one tuple to another through the linkages (refer to the Dynamic Data Manual [235-600-234] for a description). The top function in the stack trace sets the "quarantine" attribute in the rIPCBLA tuple so the audit will find an error and recover the rIPCBLA tuple to the idle list, then it prints the assert and schedules the PCBLA audit to be run. Therefore, in this case, looking at the assert is what is needed rather than looking at the PCBLA audit.

In this example, the RLPCBLA relation offset is 10, hence, the physical key of 9 and logical key of H'ffff (or -1), which is the idle list for the RLCCBCOM relation. If the stack trace were not available, a search of the code for instances of this assert could be restricted to code dealing with the RLCCBCOM relation and its rIPCBLA owned idle list. In this example, the stack trace can be analyzed to find the function that fired the assert.

```
S570-76 99-03-02 07:51:58 001183 AUDTMON N1 i99004sym
AUD SM=1 PCBLA ERROR-CODE=QUAR EVENT=8
ERROR-ADDR=H'1fb4200 BAD-DATA=H'6
LOG-KEY=H'9 GOOD-DATA=H'0
```

S570-76 99-03-02 07:51:58 001184 AUDTMON N1 i99004sym

AUD SM=1 PCBLA DUMP PART-1 EVENT=8

ERROR-ADDR=H'1fad900

KEY=PCBLA H'ffff

BLOCK-ADDR=H'1fad900

```
FFFF0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00030000 00000000
00010000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 FFFF0000 FFFF0000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000
```

S570-76 99-03-02 07:52:07 001186 AUDTMON N1 i99004sym

AUD SM=1 PCBLA DUMP PART-2 EVENT=8

ERROR-ADDR=H'1fad900

KEY=PCBLA H'ffff

BLOCK-ADDR=H'1fad9c4

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 02630000 00000000 00000000
```

S570-76 99-03-02 07:52:10 001187 OVLD N1 i99004sym

* OP OVRLD SM=1

REAL TIME NONE

RESOURCE CCB COM

CONTROLS DNET

CONTROLS AVRT

S570-76 99-03-02 07:52:13 001189 OVLD N1 i99004sym

REPT OVERLOAD HISTORY

MODULE REALTIME RESOURCE

SM=1 NONE CCB COM

END

S570-76 99-03-02 07:52:15 001190 AUDTMON N1 i99004sym

AUD SM=1 PCBLA DUMP PART-1 EVENT=8

ERROR-ADDR=H'1fad900

KEY=CCB COM H'1dc

BLOCK-ADDR=H'22bf630

```
01DC0000 00000000 00000000 00000000 01DC4000 00000000 8C000000 FF000000
00000000 FFFFFFFF FFFF0000 00000000 00000000 00080008 01100000 09060006
01010103 0A400235 00000235 00970046 00000623 029D0104 00010013 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 01DCFFFF FFFFFFFF 00FF0000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 01DCFFFF FFFFFFFF
00FF0000
```

S570-76 99-03-02 07:52:24 001193 AUDTMON N1 i99004sym

AUD SM=1 PCBLA DUMP PART-2 EVENT=8

ERROR-ADDR=H'1fad900

KEY=CCB COM H'1dc

BLOCK-ADDR=H'22bf6f4

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```


00000000 00000000 00000000 00000000 00000000 00000000 00000000 0009029B
 00000000 00000000 00000000

S570-76 99-03-02 07:52:32 001195 AUDTMON N1 i99004sym
 AUD SM=1 PCBLA COMPLETED ERRORS=1 EVENT=8

S570-76 99-03-02 07:52:42 001197 ASRT N1 i99004sym
 INIT SM=1,1 LVL=RPI EVENT=8 COMPLETED
 DEF-CHK-FAIL=21291 AUD-SCHED=PCBLA ELEV-MODE
 FAILING-ADDR=H'3c462c SM-MODE=NORMAL TIME=45:58.1
 PROCESS: BG=431,5,H'10bb656,RPI CM=NONE, FG=NONE,,

S570-76 99-03-02 07:52:51 001199 ASRTMON N1 i99004sym
 REPT SM=1 STACK TRACE ENV=OSDSM SRC=DCF EVENT=8
 USER: 003C462C 006637AC 0068F84A 0068EC7E 0068D95C 004BBEC0
 004EE476 010BBB76 00290630

S570-76 99-03-02 07:52:59 001203 ASRTMON N1 i99004sym
 REPT SM=1 STACK FRAME ENV=OSDSM SRC=DCF EVENT=8
 FUNC ADDR: H'3c462c
 PARAMETERS: FFFF0006 022FB564 0000FFFF FFFF0229 1ED00008
 8DC60068 F84A0006 00000793 00000288 00010236
 LOCAL DATA: 00000000 1A95F501 E01E2902 00000000 05000000
 2B532C46 3C00848D 0800668D 08000000 00000000
 00000000 00008D00 00000900 00000000 00000000
 00000000 0000188D 0800668D 08000000 00000081
 2902D01F 2902F4D9 FA019A8D 08000081 2902668D
 08002E0E B000008D 668D0800 668D0800 00000000
 00002B53 96B41F00 668D0800 08000000 8AA5AA00
 03002B53 668D0800 668D0800 00000000 21000000
 00000000 FDFFFFFFFF 0000204E 00001001 D01E2902

S570-76 99-03-02 07:53:07 001206 ASRTMON N1 i99004sym
 REPT SM=1 STACK FRAME ENV=OSDSM SRC=DCF EVENT=8
 FUNC ADDR: H'6637ac
 PARAMETERS: 00060000 07930000 02880001 0236002F 00000229
 1EE00229 1ED00000 0000FF4A 00088E7A 0068EC7E
 LOCAL DATA: D01E2902 FFFFFFFF 000064B5 2F020600 FFFFAC37
 66009E8D 08000000 00001A95 F501E01E 29020000
 00000500 00002B53 2C463C00 848D0800 668D0800
 00000000 00000000 00000000 8D000000 09000000
 00000000 00000000 00000000 188D0800 668D0800
 00000000 00812902 D01F2902 F4D9FA01 9A8D0800
 00812902 668D0800 2E0EB000 008D668D 0800668D
 08000000 00000000 2B5396B4 1F00668D 08000800
 00008AA5 AA000300 2B53668D 0800668D 08000000

S570-76 99-03-02 07:53:15 001208 ASRTMON N1 i99004sym
 REPT SM=1 REGISTER DUMP ENV=OSDSM SRC=DCF EVENT=8
 REGISTER DATA= 02298100 00088D9A 01FAD9F4 02291FD0
 02298100 00000000 00088D66 00088D18
 00000000 00000000 00000000 00000009
 0000008D 00000000 00000000 00000000

```
S570-76 99-03-02 07:53:23 001210 ASRTMON N1 i99004sym
REPT SM=1 PMDB-IN SRC=DCF EVENT=8
MSG:  TYPE=124  LENGTH=4  PRIORITY=0
FROM:  PROC=2  PCRID=1  UNIQ=0
02880001
```

4.4.8.2.2 Audit scheduled from a Single Process Purge (SPP)

In the following example, the single process purge (SPP) appears first, and is caused by a write protect error. In other words, a function is attempting to write to a location of memory it is not allowed to touch. The fifth block in the event is a "STACK TRACE" and all those addresses may be traced back to the line numbers of the called functions. More detail is given about tracking down this problem in the Asserts section of the Software Analysis Guide (235-600-510).

An SPP purges the process that tried to do an illegal action, sets a "quarantine" attribute in some of the tuples linked to the rIPCBLA tuple so that audits will find an error and recover the data to the idle list, and then it schedules several audits as part of the cleanup. This cleanup can sometimes take quite a while, producing a number of ROP reports.

Similar to the assert case, the SPP error blocks sometimes appear further down in the event. Depending on the type of process, there may be associated data or processes that generate further reports.

```
S570-131186 97-07-29 15:10:53 005349 INT A i8.1
REPT SM=4,0,ACT HWLVL=1 SWLVL=SPP EVENT=64 COMPLETED
MP WRITE-PROT-ERR SW-ERR FAILING ADDR=H'8bd
PROCESS:BG=1210,5,PURGED CM=NONE, FG=NONE,,
```

```
S570-131186 97-07-29 15:10:59 005350 INT_MON A i8.1
REPT SM=4,0,ACT HWLVL=1 SWLVL=SPP EVENT=64 COMPLETED
MP WRITE-PROT-ERR
SW-ERR FAIL-ADDR=H'8bd ROM-WRITE DATA-BUS=H'0 TIME=8:2.3
PROCESS:BG=1210,5,PURGED CM=NONE, FG=NONE,, NORMAL
ORIG-HW-STATUS: MC0: ACT MC1: STBY
FINAL-HW-STATUS: MC0: ACT MC1: STBY
PREVIOUS TYPE/COUNT: 121 0
SHADOW TYPE/COUNT: 86 2
AUX DATA: H'0 H'200 H'0 H'1
ESCALATION-COUNTS: H'10000 H'0 H'0 H'0
```

```
S570-131186 97-07-29 15:11:09 005352 INT_MON A i8.1
REPT SM=4 HARDWARE CONTEXT SM2000 ORM EVENT=64
68040-REGISTERS: SSP=H'100cffc PC=H'101b02c SR=H'4
USP=H'11195d2 FP=H'111963e A5=H'250d342 A4=H'2387e8c
A0=H'8bb A1=H'22e6d90 A2=H'22e87d0 A3=H'250d39a
D0=H'ffff D1=H'94 D2=H'0 D3=H'f150a3c
D4=H'0 D5=H'0 D6=H'0 D7=H'0
PIC-REGISTERS: HI MED LOW
IRR: H'0 H'c0 H'40
IMR: H'e H'3b H'fe
ISR: H'0 H'0 H'0
```

```
S570-131186 97-07-29 15:11:13 005354 INT_MON A i8.1
REPT SM=4 MP HW REGS EVENT=64
ACTSR=H'3 EXCEP=2 PORTA=H'0 PRTBN=H'3b
PROC1=H'1 PROC2=H'0 PROC3=H'e0
```

```

RSTSR=H'0      HRSRC=H'0      SRSRC=H'0
RSMRK=H'44     HRSMR=H'0      SRSMR=H'0
SESR=H'0       BSNE=H'0       APESR=H'c
SESMR=H'c8     BSNE=H'0       APESM=H'f00c
CORES=H'0      CAXES=H'0      COCTL=H'0      CPDSR=H'1e
CORMR=H'0      CAXEM=H'0      MCIDR=H'1      CPISR=H'f1
SUBRR=H'ff     MEMBD=H'ffff    CDSR1=H'0
MBSR10=H'7c    MBSR11=H'7c     MBSR12=H'7c     MBSR13=H'ff
MBESR0=H'0     MBESR1=H'0     MBESR2=H'82     MBESR3=H'ff
MBESM0=H'82    MBESM1=H'82     MBESM2=H'82     MBESM3=H'ff
BUSCR=H'b3     SHADDR=H'0
SHBCR=H'b3     SHOPR=H'0

```

```

S570-131186 97-07-29 15:11:17 005356 INT_MON A i8.1
REPT SM=4 STACK TRACE ENV=OSDSM SRC=FR EVENT=64
USER: 0101B02C 0F150A3C 016952C4 01AE6BFA 01AE7B9E 01AE61DE
      01AD7922 01AD6A90 01616D98

```

```

S570-131186 97-07-29 15:11:21 005359 INT_MON A i8.1
REPT SM=4 STACK FRAME ENV=OSDSM SRC=FR EVENT=64
FUNC ADDR: H'101b02c
PARAMETERS: 00000000 00000000 00000000 00000000 022E87D0
             0250D39A 02387E8C 0250D342 016967F0 010D20F2
LOCAL DATA: 085E6901 64961101 FFFFFFFF D0872E02 01000000
             2E882E02 ED01D087 2E026A6A 69013696 11010100
             00000000 3802562D 5002D087 ED010003 0504BA04
             0000028D 18002201 0504BA04 00042700 52256101
             32961101 D0872E02 00000000 0E961101 00000000
             00000000 3C0A150F FFFF1EB0 01013E96 1101D087
             2E025F00 D0872E02 00000000 010094D8 69011101
             CA000100 CD008204 A4E76901 B2951101 7CFF2081
             E6032201 00042700 01000504 BA049095 1101D087

```

```

S570-131186 97-07-29 15:11:26 005361 INT_MON A i8.1
REPT SM=4 STACK FRAME ENV=OSDSM SRC=FR EVENT=64
FUNC ADDR: H'f150a3c
PARAMETERS: 022E87D0 0250D34A 00000000 01119708 01119718
             01AE6BFA 0250D342 04820000 0F150A3C 00000000
LOCAL DATA: 0196D087 2E029C8C 2E02F220 0D01F067 690142D3
             50028C7E 38029AD3 5002D087 2E020000 00000000
             00000000 00000000 00003C0A 150F7896 1101085E
             69016496 1101FFFF FFFFD087 2E020100 00002E88
             2E02ED01 D0872E02 6A6A6901 36961101 01000000
             00003802 562D5002 D087ED01 00030504 BA040000
             028D1800 22010504 BA040004 27005225 61013296
             1101D087 2E020000 00000E96 11010000 00000000
             00003C0A 150FFFFFF 1EB00101 3E961101 D0872E02

```

```

S570-131186 97-07-29 15:11:32 005362 INT_MON A i8.1
REPT SM=4 DATA=PCB,1210 ENV=OSDSM SRC=FR EVENT=64
ADDR=H'39fdd00
04BA002A 0A05FFFF FFFFFFFF 00000000 01119596 00800000 FFFFFFFF FFFFFFFF
00270100 0F150A3C 00000000 00000000 00000000 00000000 022E87D0 0250D39A
02387E8C 0250D342 011195C8 011195B4 00000000 00000000 00000000 00000000

```

00040400 00000000 0078D78C 00800018 FFFFFFFF FFFF0000 0000000B 00000000

S570-131186 97-07-29 15:11:52 005369 INT_MON A i8.1
REPT SM=4 DATA=PCBLA-1,1210 ENV=OSDSM SRC=FR EVENT=64
ADDR=H'39ff400

04BA0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00070500 00000000 00000000 00000100 00000000 00000000 01000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
FFFFF000 FFFF0000 00000000 FFFF0000 FFFF0000 FFFF0000 FFFF0000 FFFF0000
FFFFF000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000

S570-131186 97-07-29 15:11:56 005370 INT_MON A i8.1
REPT SM=4 DATA=PCBLA-2,1210 ENV=OSDSM SRC=FR EVENT=64
ADDR=H'39ff4d4
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
05090509 00000000 00000000

S570-131186 97-07-29 15:12:00 005371 AUDTMON A i8.1
AUD SM=4 SMXRF ERROR-CODE=DEADPID EVENT=64
ERROR-ADDR=H'2264fb0 BAD-DATA=H'0
LOG-KEY=H'41 GOOD-DATA=H'0

S570-131186 97-07-29 15:12:04 005372 AUDTMON A i8.1
AUD SM=4 SMXRF DUMP EVENT=64
ERROR-ADDR=H'2264fb0
KEY=SMXRF H'41
BLOCK-ADDR=H'2264fb0
0041023A 000000A4 00020000 10600000 04BA0405 00000000 000A0000 00000000

S570-131186 97-07-29 15:12:09 005373 AUDTMON A i8.1
AUD SM=4 SMXRF COMPLETED ERRORS=1 EVENT=64

S20B-65603 97-07-29 15:12:12 005374 no_cls A i8.1
* REPT SOP CANNOT OPEN SCC CHANNEL

S570-131186 97-07-29 15:12:13 005375 AUDTMON A i8.1
AUD SM=4 CCB COM ERROR-CODE=QUAR EVENT=64
ERROR-ADDR=H'3e68110 BAD-DATA=H'1
LOG-KEY=H'482 GOOD-DATA=H'0

S570-131186 97-07-29 15:12:18 005376 AUDTMON A i8.1
AUD SM=4 CCB COM DUMP PART-1 EVENT=64
ERROR-ADDR=H'3e68110
KEY=CCB COM H'482
BLOCK-ADDR=H'3e68110
04820000 00000000 00000060 04000000 04824000 00000000 8C640000 FF000000
00000000 FFFFFFFF FFFF0000 00000000 00000000 00080008 01980000 09060007
010301C3 580003F7 000003F7 00970007 00008000 007816A1 06201000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0482FFFF FFFFFFFF 00FF0000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 0482FFFF FFFFFFFF
00FF0000

S570-131186 97-07-29 15:12:22 005377 AUDTMON A i8.1
AUD SM=4 CCB COM DUMP PART-2 EVENT=64
ERROR-ADDR=H'3e68110
KEY=CCB COM H'482
BLOCK-ADDR=H'3e681d4
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 023A023A 00000000 00000000 00000000 04C40000
00000000 00000000 00000000

S570-131186 97-07-29 15:12:27 005378 AUDTMON A i8.1
AUD SM=4 CCB COM COMPLETED ERRORS=1 EVENT=64

S570-131186 97-07-29 15:12:31 005381 AUDTMON A i8.1
AUD SM=4 CDB COM ERROR-CODE=MEMBER EVENT=64
ERROR-ADDR=H'3f07f40 BAD-DATA=H'1
LOG-KEY=H'23a GOOD-DATA=H'0

S570-131186 97-07-29 15:12:36 005383 AUDTMON A i8.1
AUD SM=4 CDB COM DUMP EVENT=64
ERROR-ADDR=H'3f07f40
KEY=CDB COM H'23a
BLOCK-ADDR=H'3f07f40
023A0000 013A0010 00010004 005A0000 00000000 00000000 05090000 00000000

S570-131186 97-07-29 15:12:41 005384 AUDTMON A i8.1
AUD SM=4 CDB COM DUMP PART-1 EVENT=64
ERROR-ADDR=H'3f07f40
KEY=CCB COM H'482
BLOCK-ADDR=H'3e68110
04820000 00000000 00000000 00000000 04824000 00000000 8C640000 FF000000
00000000 FFFFFFFF FFFF0000 00000000 00000000 00080008 01980000 09060007
00000000 00000000 000003F7 00970007 00008000 007816A1 06201000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0482FFFF FFFFFFFF 00FF0000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 0482FFFF FFFFFFFF
00FF0000

S570-131186 97-07-29 15:12:45 005386 AUDTMON A i8.1
AUD SM=4 CDB COM DUMP PART-2 EVENT=64
ERROR-ADDR=H'3f07f40
KEY=CCB COM H'482
BLOCK-ADDR=H'3e681d4
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00090000
05240000 00000000 00000000

S570-131186 97-07-29 15:12:49 005387 AUDTMON A i8.1
AUD SM=4 CDB COM ERROR-CODE=ASSOC_EST EVENT=64
ERROR-ADDR=H'3dfdf40 BAD-DATA=H'0
LOG-KEY=H'23a GOOD-DATA=H'0

S570-131186 97-07-29 15:12:54 005388 AUDTMON A i8.1
AUD SM=4 CDB COM DUMP EVENT=64

ERROR-ADDR=H'3dfdf40
KEY=SMEST H'23a
BLOCK-ADDR=H'3dfdf40
023A013A 00980041 00040405 00605001 0026C100 04BA0405 DA380155 00000000

S570-131186 97-07-29 15:12:57 005389 AUDTMON A i8.1
AUD SM=4 CDBCOM COMPLETED ERRORS=2 EVENT=64

S570-131186 97-07-29 15:13:01 005390 AUDTMON A i8.1
AUD SM=4 CKTDATA ERROR-CODE=PID EVENT=64
ERROR-ADDR=H'3dfd840 BAD-DATA=H'4ba0405
LOG-KEY=H'202 GOOD-DATA=H'0

S570-131186 97-07-29 15:13:12 005392 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3dfd840
KEY=SMEST H'202
BLOCK-ADDR=H'3dfd840
02020124 00980041 0004040B 00005001 0026C100 04BA0405 DA380155 00000000

S570-131186 97-07-29 15:13:17 005393 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3dfd840
KEY=SMEST H'246
BLOCK-ADDR=H'3dfe0c0
024600D7 00980041 0004040B 00005001 0026C100 04BA0405 DA380155 00000000

S570-131186 97-07-29 15:13:28 005394 AUDTMON A i8.1
AUD SM=4 CKTDATA ERROR-CODE=AUOOS EVENT=64
ERROR-ADDR=H'3dfd840 BAD-DATA=H'b
LOG-KEY=H'202 GOOD-DATA=H'2

S570-131186 97-07-29 15:13:33 005395 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3dfd840
KEY=SMEST H'202
BLOCK-ADDR=H'3dfd840
02020124 00980052 0002040B 00015001 0026C100 FFFF0000 DA380155 00000000

S570-131186 97-07-29 15:13:40 005396 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3dfd840
KEY=SMEST H'246
BLOCK-ADDR=H'3dfe0c0
024600D7 00980041 0004040B 00005001 0026C100 04BA0405 DA380155 00000000

S570-131186 97-07-29 15:13:49 005398 AUDTMON A i8.1
AUD SM=4 CKTDATA ERROR-CODE=ASSOC_TUP EVENT=64
ERROR-ADDR=H'3f07840 BAD-DATA=H'66
LOG-KEY=H'202 GOOD-DATA=H'0

S570-131186 97-07-29 15:13:54 005399 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3f07840

```
KEY=CDBCOM H'202
BLOCK-ADDR=H'3f07840
02020000 01240000 01080004 005A0000 00000000 00000000 00710204 00000000
```

```
S570-131186 97-07-29 15:14:00 005401 AUDTMON A i8.1
AUD SM=4 CKTDATA ERROR-CODE=AUOOS EVENT=64
ERROR-ADDR=H'3dfd840 BAD-DATA=H'b
LOG-KEY=H'202 GOOD-DATA=H'2
```

```
S570-131186 97-07-29 15:14:05 005404 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3dfd840
KEY=SMEST H'202
BLOCK-ADDR=H'3dfd840
02020124 00980052 0002040B 00015001 0026C100 FFFF0000 DA380155 00000000
```

```
S570-131186 97-07-29 15:14:09 005407 AUDTMON A i8.1
AUD SM=4 CKTDATA DUMP EVENT=64
ERROR-ADDR=H'3dfd840
KEY=SMEST H'246
BLOCK-ADDR=H'3dfe0c0
024600D7 00980041 0004040B 00005001 0026C100 04BA0405 DA380155 00000000
```

```
S570-131186 97-07-29 15:14:14 005411 AUDTMON A i8.1
AUD SM=4 CKTDATA COMPLETED ERRORS=20 EVENT=64
```

4.4.8.3 Case History Three: CKTDATA Audit

This case study illustrates how ROP generated from a circuit data (CKTDATA) audit is analyzed. The ROP was generated due to dynamic data inconsistencies detected by the CKTDATA audit. In this case, the switch personnel gathers the information regarding event 4 provided in the following ROP output.

```
S0-13799 99-04-26 09:22:02 000289 AUDTFST FIVE
AUD SM=192 CKTDATA ERROR-CODE=VERTICAL EVENT=4
ERROR-ADDR=H'5bda9f4 BAD-DATA=H'4
LOG-KEY=H'91c GOOD-DATA=H'4
```

```
S0-13799 99-04-26 09:22:03 000290 AUDTFST FIVE
AUD SM=192 CKTDATA DUMP PART 1 OF 1 EVENT=4
KEY=r1SMEST TUPLE WHERE
  <the_key>
  =<H'91c>
BLOCK-ADDR=H'5bda9f4
091C00D6 00980052 0002040B 01215001 0026C100 FFFF0000 042687AC 00000000
```

```
S0-13799 99-04-26 09:22:05 000291 AUDTFST FIVE
AUD SM=192 CKTDATA DUMP PART 1 OF 1 EVENT=4
KEY=r1SMEST TUPLE WHERE
  <the_key>
  =<H'81c>
BLOCK-ADDR=H'5bd89f4
081C00D7 00000052 00000200 00014101 FFFF0000 FFFF0000 042687AC 00000000
```

```
S0-13799 99-04-26 09:22:06 000292 AUDTFST FIVE
```

```

AUD SM=192 CKTDATA DUMP PART 1 OF 1 EVENT=4
KEY=r1SMEST TUPLE WHERE
  <the_key>
  =<H'89c>
BLOCK-ADDR=H'5bd99f4
089C00D7 00000052 00000100 00014101 FFFF0000 FFFF0000 042687AC 00000000

S0-13799 99-04-26 09:22:08 000293 AUDTFST FIVE
AUD SM=192 CKTDATA DUMP PART 1 OF 1 EVENT=4
KEY=r1SMEST TUPLE WHERE
  <the_key>
  =<H'61c>
BLOCK-ADDR=H'5bd49f4
061C013A 00000052 00000200 00014001 FFFF0000 FFFF0000 042687AC 00000000

S0-13799 99-04-26 09:22:10 000294 AUDTFST FIVE
AUD SM=192 CKTDATA DUMP PART 1 OF 1 EVENT=4
KEY=r1SMEST TUPLE WHERE
  <the_key>
  =<H'65c>
BLOCK-ADDR=H'5bd51f4
065C013A 00000052 00000100 00014001 FFFF0000 FFFF0000 042687AC 00000000

S0-13799 99-04-26 09:22:12 000295 AUDTFST FIVE
AUD SM=192 CKTDATA COMPLETED ERRORS=1 EVENT=4
S0-13799 99-04-26 09:24:15 000296 FIVE
A RST PSUPH=192-0-0-5 COMPLETED

```

The CKTDATA audit steps through the static RLCKTDATA relation. The RLCKTDATA relation stores the definition for each circuit that is known by a switching module (SM). For each of these circuits, the CKTDATA audit verifies that the proper corresponding dynamic data exists and that this dynamic data is in a consistent state. For further details on the types of checks the CKTDATA audit makes, please refer to the Audits Manual (235-600-400).

The next step the switch personnel should take is to look up the CKTDATA audit VERTICAL error description in the Audit Manual. From the Audit Manual the switch personnel will learn that a VERTICAL error is reported when an inconsistency in maintenance state is found between the circuit under audit (the child circuit) and its parent circuit(s).

The Audit Manual will also describe what is dumped in each of the error report fields for a VERTICAL error. ERROR-ADDR contains the address of the r1SMEST tuple for the child circuit, BAD-DATA contains the r1SMEST basic status (bas_stat) of the child circuit, GOOD-DATA contains the number of parents this child circuit has, and LOG-KEY contains the r1SMEST circuit name (unit_id) of the child circuit. Lastly the Audit Manual will describe the data dumps that follow a given error report. For VERTICAL errors, the first report is the r1SMEST tuple corresponding to the child circuit and all the rest of the data dumps are the r1SMEST tuples for the parent circuits.

All of the audit data dump reports for this case history are of a switching module equipment status table tuple (r1SMEST). The RLSMEST relation is a dynamic relation that contains a tuple per circuit defined in the static RLCKTDATA relation. The dynamic state and activity of a circuit can be obtained from an r1SMEST tuple. The layout of this dynamic data can be found in the Dynamic Data Manual (235-600-2xx).

For analyzing VERTICAL errors, the r1SMEST tuple field values needed are the circuit name (unit_id), circuit type (type), basic status (bas_stat), and second qualifier (qual_2) fields. Refer to the "Extracting Data from an Audit Report" section for more information. The switch personnel will need to extract these field values from each of the r1SMEST tuples dumped.

The circuit name is the key to both the r1CKTDATA tuple and the r1SMEST tuple. The circuit type is defined by the

DMCKTTYPE enumeration. The basic status represents the maintenance state of the circuit. The second qualifier indicates the reason why the basic status is set to a particular maintenance state. Both the basic status and second qualifier fields are defined by the DMMRASTAT enumeration.

Again, the first report contains the rSMEST tuple for the circuit under audit which is the child circuit. The circuit name of the child circuit is 0x091C, the circuit type is 0x00D6 which corresponds to the PCPHPSIU DMCKTTYPE enumeration value, the basic status is 0x04 which corresponds to the SMOOS DMMRASTAT enumeration value, and the second qualifier is 0x0B which corresponds to the SMFE DMMRASTAT enumeration value. So, this child PH circuit is in the out-of-service family of equipment state. Family of equipment means that this circuit is out-of-service due to its parent circuit being out-of-service.

The same analysis now needs to be done for each parent circuit's rSMEST tuple. The results of this can be seen in the following table:

Circuit Name	Circuit
0x091C	PCPHI
0x081C	PCSHI
0x089C	PCSHI
0x061C	PCCC
0x065C	PCCC

Looking at the table, it can be seen that the parent PSU shelf circuits are running active/stand-by. The grandparent PSU common controller circuits are also running active/stand-by. Therefore, the VERTICAL inconsistency is that the child circuit is in the out-of-service family of equipment state, but none of the parent circuits are out-of-service.

Recovery for this VERTICAL error is to load the PH circuit on the MRA recovery queue. MRA will then take the PH circuit off of the queue and take the appropriate recovery action to make the child PH circuit's state consistent with its parents. In this case, MRA restored the PH circuit to the active state. The restore completed message is the last message seen on the ROP.

4.4.8.4 Case History Four: PCBLA Audit

This case history addresses the PCBLA audit. Call processing and terminal and switch maintenance are implemented by means of processes that are associated with tuples of relation RLPCBLA (process control block link area). Each rIPCBLA tuple has links to tuples in other relations that contain all the information about a call such as ports and time slots used. They also include controls of the call itself such as call waiting or forwarding data. Incorrect data in an rIPCBLA tuple can cause a variety of problems related to call processing.

The function of the PCBLA audit is to verify the integrity of the data and the link areas. The subsequent audit reports and data dumps are related as indicated by the common EVENT number.

```
S570-65669 95-10-06 22:21:13 050504 AUDTMON FIVE I
AUD SM=17 PCBLA ERROR-CODE=DEAD_LNKD EVENT=1055
ERROR-ADDR=H'1425900 BAD-DATA=H'4a4
LOG-KEY=H'3d9 GOOD-DATA=H'0
```

```
S570-65669 95-10-06 22:21:21 050512 AUDTMON FIVE I
AUD SM=17 PCBLA DUMP EVENT=1055
ERROR-ADDR=H'1425900
KEY=PCBLA H'3d9
BLOCK-ADDR=H'1425900
03D90000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 FFFF0000 FFFF0000 00000000 FFFF0000 FFFF0000 0000B100 00008000
01000010 00000000 00000000 00000000 00000000 FFFF0000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 B3C20000
```

```
S570-65669 95-10-06 22:21:29 050516 AUDTMON FIVE I
```

```
AUD SM=17 PCBLA DUMP EVENT=1055
ERROR-ADDR=H'1425900
KEY=PCB H'3d9
BLOCK-ADDR=H'808e76
010300B2 031F03B6 FFFF0000 000CBF0A 08B80000 FFFFFFFF FFFFFFFF FFFFFFF00
```

```
S570-65669 95-10-06 22:21:45 050521 AUDTMON FIVE I
AUD SM=17 PCBLA COMPLETED ERRORS=1 EVENT=1055
```

The previous error code, DEAD_LNKD, indicates that the operating system believes the process associated with this rPCBLA tuple to be dead, but the linkage fields are not null. This means that the system has resources tied to processes that are no longer active. The purpose of this check is to report the error and then release all of the resources that the dead process is holding. The ERROR-ADDR in the third line of the report is the physical address of the failing tuple, and BAD-DATA is the bad linkage indicator. LOG-KEY is the process number. Two data dumps accompany the PCBLA audit error report: the rPCBLA tuple and the process control block (PCB) associated with the process.

The process type can be found by looking at the *progi* field in the PCB dump. Refer to the "Extracting Data from an Audit Report" section for information on how to determine the value of this attribute.

When the PCBLA audit finds an error, the system will automatically call several other audits that will check the data that the rPCBLA tuple could be linked to. For example, the port linkage area (PORTLA) contains data for all defined line and trunk ports. When the rPCBLA tuple is recovered, it breaks the linkage to the rPORTLA tuple. The PORTLA audit reports the ONE2ONE error. Similarly, other data linked to the rPCBLA tuple may be detected by additional audits. Typically, these error reports have the same event number.

```
S570-65669 95-10-06 22:22:10 050527 AUDTMON FIVE I
AUD SM=17 PORTLA ERROR-CODE=ONE2ONE EVENT=1055
ERROR-ADDR=H'152f040 BAD-DATA=H'2
LOG-KEY=H'b3c2 GOOD-DATA=H'0
```

```
S570-65669 95-10-06 22:22:18 050528 AUDTMON FIVE I
AUD SM=17 PORTLA DUMP EVENT=1055
ERROR-ADDR=H'152f040
KEY=PORTLA H'b3c2
BLOCK-ADDR=H'152f040
B3C20100 00000A00 00009D9F 04CA3800 00000000 00019261 00000000 03E20000
```

```
S570-65669 95-10-06 22:22:26 050530 AUDTMON FIVE I
AUD SM=17 PORTLA COMPLETED ERRORS=1 EVENT=1055
```

The error code ONE2ONE is not described in the PORTLA section of the Audits Manual (235-600-400). It can be found in the Audit Drivers section of the manual under the heading DRERRCDS. This is because the error was detected by a common program (driver) invoked by the PORTLA audit.

The ONE2ONE error is caused by an invalid one-to-one linkage, meaning that the failing tuple is pointing to a tuple that does not point back. The audit driver re-initializes the rPORTLA tuple and, if it belongs on an idle link list, links it to the appropriate head cell. The ERROR-ADDR is the physical address of the failing tuple, BAD-DATA is the sub-index to the per-tuple linkage block, and LOG-KEY is the logical key of the tuple in error. The accompanying PORTLA dump is that of the corrupted tuple.

5. Audit Drivers

Audit Drivers

Software Release: 5E10 and later

1. AUDIT DRIVERS

An audit driver is a common program that is called by specific audits to control segment breaks, perform syntactic checks, schedule semantic checks, and handle errors. The driver eventually returns control to the audit control function that invoked the audit. The 5ESS[®] switch audits system uses the following drivers:

- Bit Mask Driver (DRBMSK).
- Dynamic Driver (DRDYN).
- Linkage Driver (DRLINK).
- Indirect Access Sequential Key Driver (DRINASEK).
- Path Map Driver (DRPHMAP).
- Static Driver (DRSTATIC).
- Traverse Driver (DRTRAV).
- Indexed Access Driver (DRINDEX).

Most audits invoke one driver, which then directs the activities of the invoking audit.

Program control differs for each of the drivers (see Figure 1). Each audit driver is individually explained in its respective description, and is located alphabetically in the "Audit Descriptions" section.

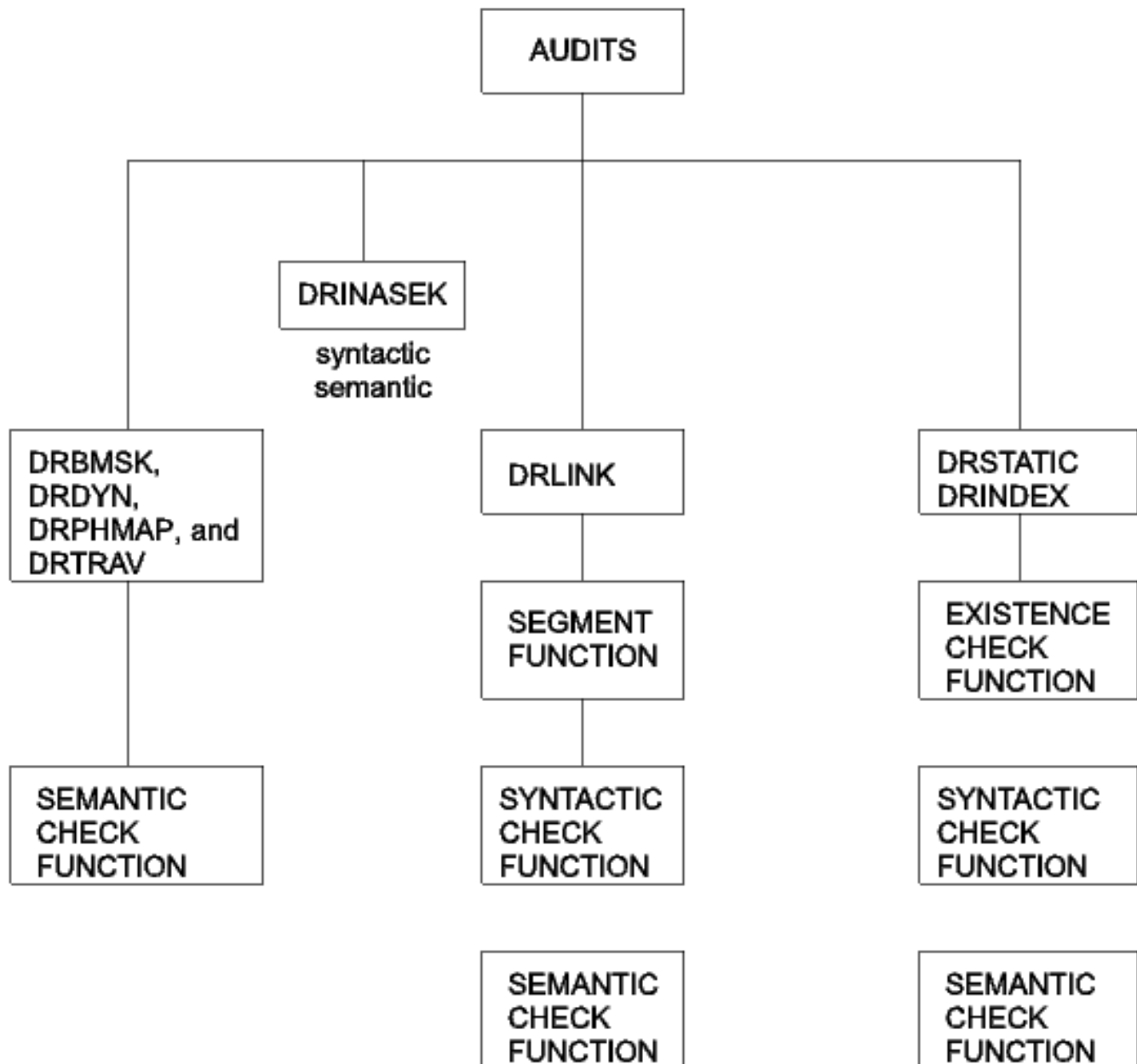


Figure 1 Audit Drivers

DRBMSK

Software Release: 5E10 and later

1. BIT MASK DRIVER (DRBMSK)

The Bit Mask Driver (DRBMSK) is used to sequence through bit masks in the data structures being audited. A bit mask is an array of bits; each bit controls some action or activity. Sequencing control information for all data structures audited with the DRBMSK driver is stored in a table. The table contains information such as the number of bits in the structure, the number of bits to be audited in each time segment, and information about how to access a database tuple. The DRBMSK driver runs in the Operating System for Distributed Switching-Switching Module (OSDS-SM) environment and is invoked by audits in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DRBMSK driver operates on the bit mask of the data structure being audited. It also operates on associated database tuples.

AUDIT CONTROL

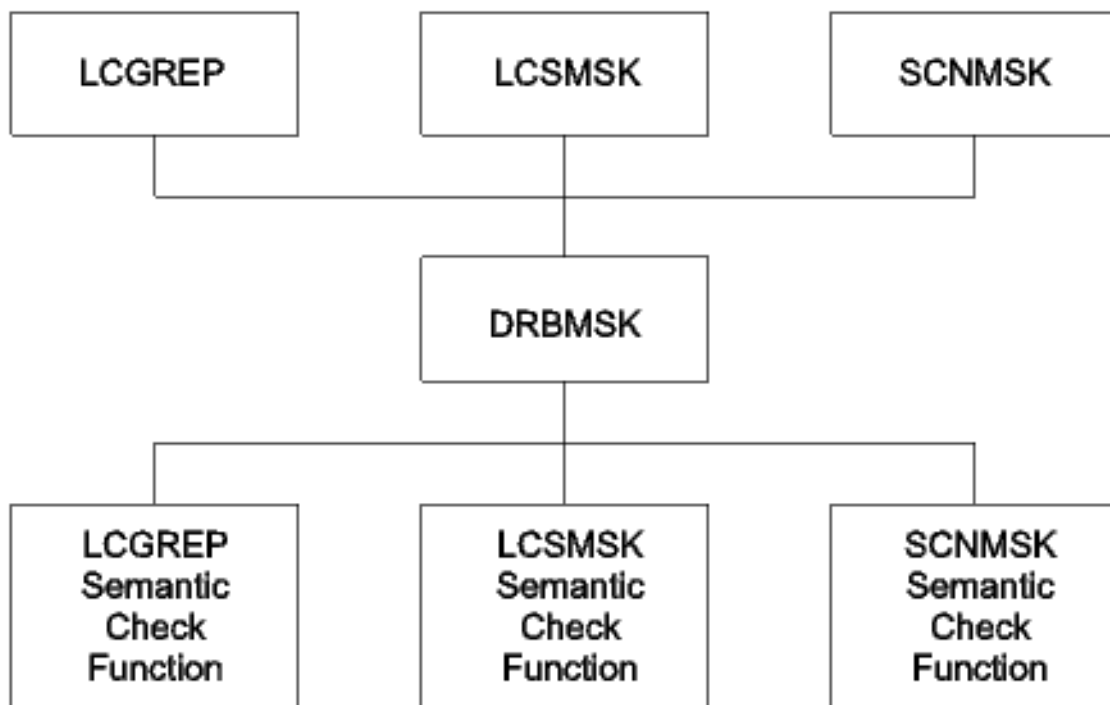


Figure 1 Bit Mask Driver

3. AUDIT CHECKS

The processing flow for the DRBMSK driver is shown in Figure 1. The invoking audit calls the driver. The DRBMSK driver has its own timing function and does not use the segmenting function to schedule its segment breaks. The DRBMSK driver always calls the semantic check function appropriate to the invoking audit. Generally, the processing flow is as follows:

- (a) The invoking audit calls the DRBMSK driver and passes to it the manner (ZAP, START, RESUME) in which it

should be run. The audit also passes the data structure ID to the DRBMSK driver.

- (b) The DRBMSK driver uses the element number, the bit position of the element, and the data structure ID to calculate a key and retrieve the appropriate database tuple.
- (c) The DRBMSK driver calls a semantic check function (determined from the data structure ID) to audit the desired bit using the database tuple retrieved in the previous step.

4. ERROR HANDLING

If an error is found, the audit invokes the appropriate error handling function.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

DRDYN

Software Release: 5E10 and later

1. DYNAMIC DRIVER (DRDYN)

The Dynamic Driver (DRDYN) is used to step through direct access relations with tuples that have sequential keys, similar to the Linkage Driver (DRLINK). However, unlike the DRLINK driver, the DRDYN driver does not perform any syntactic (linkage, quarantine state, or key) checks before invoking the semantic check function. The DRDYN driver is used as a common function; it provides the interface between the audit and audit control, and contains information on how to proceed from one tuple to the next. The DRDYN driver runs in both the Operational Kernel Process (OKP) in the Administrative Module (AM) and the Switching Module (SM) environments, and is invoked by audits in the segmented and non-segmented modes. As of the 5E6(1) Generic the DRDYN driver also runs in the Communications Module Processor (CMP) environment in the segmented and non-segmented modes.

2. DATA STRUCTURES

The DRDYN driver operates only on direct access relations.

The DRINDEX driver also refers to the following data:

- AUPRI_arr[] is the audit drivers per relation information array containing information about the current relation under audit. The per relation information contains:
 - the number of headcells tuples,
 - the number of dummy (unused) tuples,
 - if the relation definition specifies a quarantine bit,
 - the size of one tuple,
 - the number of tuples in the relation,
 - the offset between logical and physical keys,
 - the beginning address of the relation,
 - the address of the semantic check function,
 - the address of the idle list function,
 - the address of the recovery function,
 - the address of the selective initialization check function,
 - the index into the linkage mapping array (AUptlb_arr[]),
 - the relations access method,
 - the path and path type to traverse linkages to the owning process,
 - and the idle list mapping identification.

3. AUDIT CHECKS

The processing flow for the DRDYN driver is shown in Figure 1 and is explained below:

- (a) The invoking audit calls the DRDYN driver and passes to it the relation ID, the name of the semantic check function, and audit control information (manner).
- (b) The DRDYN driver will check if the audit control information parameter (manner) indicates that the current running audit has taken a system interrupt (such as, write protect, invalid operations, and so forth). If this has occurred the audit will zero the tuple. This tuple will then be recovered on the next invocation of the audit.
- (c) Before accessing a tuple in the relation, the DRDYN driver checks to ensure that sufficient time remains in the segment to perform the necessary checks by invoking the audit control real time break function. The audit control real time break function will ensure that the DRDYN driver takes segment breaks every 20 milliseconds during normal operation and 40 milliseconds during recovery. If enough time is remaining the semantic check function is invoked.
- (d) If the semantic check function fails then recovery will be provided internally to the semantic check function and not the DRDYN driver.

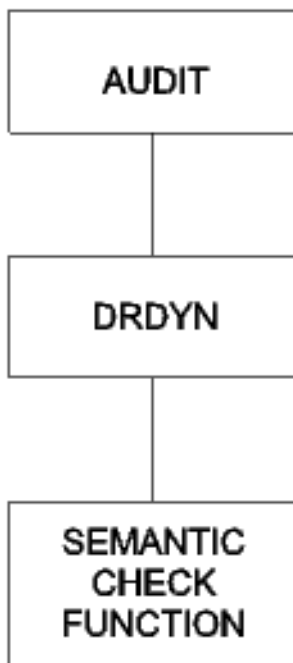


Figure 1 Dynamic Driver

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

DRERRCDS

Software Release: 5E10 and later

1. ERROR HANDLING**Table 1 Error Code: EXTRATUP**

Error Code:	EXTRATUP
Error Description:	An extra tuple was detected.
Possible Error Effect:	When recent change attempts to create a tuple, it will already exist.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key value.
<i>Dump</i> =	Tuple.

Table 2 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	While recovering a tuple found in error an attempt is made to link the tuple found in error to its appropriate head cell. However, when performing integrity checks on the head cell, the head cell's head or tail linkage has been found to be corrupt.
Possible Error Effect:	All applications are unable to access a tuple from this head cell (or idle list) head cell resulting in resource failures to the system.
Corrective Action Taken:	The audit driver will recover the head cell and will then proceed to recover all the lost resources that are supposed to be linked to this head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple found in error.
<i>Bad Data</i> =	Logical key of the head cell found in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple found in error.
<i>Dump</i> =	The head cell tuple.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	The audit driver has found the logical key of the tuple under audit to be corrupt.
Possible Error Effect:	Use of this tuple may cause asserts and/or call failures.
Corrective Action Taken:	The audit driver will reinitialize the tuple. If the tuple belongs on an idle list the audit driver will link the tuple found in error to the appropriate head cell (idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple found in error.
<i>Bad Data</i> =	Logical key from the tuple found in error.
<i>Good Data</i> =	Derived logical key for this tuple.
<i>Logical Index</i> =	Derived logical key of the tuple found in error.
<i>Dump</i> =	Tuple found in error.

Table 4 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit driver has found a tuple associated with an indirect sequential relation to missing from the database.
Possible Error Effect:	Applications requiring this resource may be denied.
Corrective Action Taken:	The tuple is inserted in the data base and initialized by the audit driver. If the tuple belongs on an idle list the audit driver will link the tuple found in error to the appropriate head cell (idle list).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address where tuple should have located in the data base.
<i>Bad Data</i> =	DB return code from attempt to insert tuple: success = 0; failure < 0.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the lost tuple.
<i>Dump</i> =	NA.

Table 5 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	A tuple has been found with bad member linkage. This tuple believes it is in a set member linkage (that is, owner, successor, predecessor) but one of the links is broken.
Possible Error Effect:	A tuple is invalidly linked to another tuple and may cause asserts and/or call failures.
Corrective Action Taken:	The audit driver will initialize the tuple and places it on the head cell (idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple found in error.
<i>Bad Data</i> =	The sub-index to the per tuple linkage block. The index into the per tuple linkage block array (AUptlb_arr[]) can be calculated by adding the bad data to the ptlb_index value in the per relation information (AUPRI_arr[]) for the relation under audit.
<i>Good Data</i> =	NA..
<i>Logical Index</i> =	Logical key of the tuple found in error.
<i>Dump 1</i> =	The member tuple found in error.
<i>Dump 2</i> =	The successor tuple of the member tuple found in error.
<i>Dump 3</i> =	The predecessor tuple of the member tuple found in error.
<i>Dump</i> =	The owner tuple of the member tuple found in error.

Table 6 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	A tuple has been found with a bad one-to-one linkage. This tuple points to another tuple but that tuple doesn't point back.
Possible Error Effect:	A tuple is invalidly pointing to another tuple and may cause asserts and/or call failures.
Corrective Action Taken:	The audit driver will reinitialize the tuple. If the tuple belongs on an idle list the audit driver will link the tuple found in error to the appropriate head cell (idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple found in error.
<i>Bad Data</i> =	The sub-index to the per tuple linkage block. The index into the per tuple linkage block array (AUptlb_arr[]) can be calculated by adding the bad data to the ptlb_index value in the per relation information (AUPRI_arr[]) for the relation under audit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple found in error.
<i>Dump 2</i> =	The tuple found in error.
<i>Dump 2</i> =	The tuple that the tuple found in error was pointing to.

Table 7 Error Code: OWNER

Error Code:	OWNER
Error Description:	A tuple has been found with bad owner linkage. The tuple believes it owns a member tuple but that member tuple does not point back.
Possible Error Effect:	A tuple invalidly owns another tuple and may cause asserts and/or call failures.
Corrective Action Taken:	The audit driver will initialize the owner tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of owner tuple found in error..
<i>Bad Data</i> =	The sub-index to the per tuple linkage block. The index into the per tuple linkage block array (AUptlb_arr[]) can be calculated by adding the bad data to the ptlb_index value in the per relation information (AUPRI_arr[]) for the relation

	under audit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the owner tuple found in error.
<i>Dump 1</i> =	The owner tuple found in error.
<i>Dump 2</i> =	The member tuple at the head of the list.
<i>Dump 3</i> =	The member tuple at the tail of the list.

Table 8 Error Code: QUAR

Error Code:	QUAR
Error Description:	A tuple has been found with its quarantine bit set. The quarantine bit may have been marked by selective initialization (bad data = AUSISET), another audit (bad data = AUAUSET), or application code (bad data = AUDEFCECK).
Possible Error Effect:	This tuple is lost to the system until the audit has recovered this tuple.
Corrective Action Taken:	The audit driver will reinitialize the tuple. If the tuple belongs on an idle list the audit driver will link the tuple found in error to the appropriate head cell (idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple found in error.
<i>Bad Data</i> =	The contents of the quarantine bit from the tuple found in error.
<i>Good Data</i> =	AUNOTSET (0).
<i>Logical Index</i> =	Logical key of the tuple found in error.
<i>Dump</i> =	The tuple found in error.

Table 9 Error Code: SGHDCELL

Error Code:	SGHDCELL
Error Description:	The head cell (idle list tuple) associated with the tuple under recovery was found to be not in the data base.
Possible Error Effect:	This tuple will be lost to the system until the head cell (idle list tuple) is inserted into the data base.
Corrective Action Taken:	The audit that checks the head cell relation will be scheduled as a related audit to try and recover the head cell if possible.
Manual Action Required:	This error could be caused by: (1) The static data associated with the dynamic head cell tuple is not populated in the ODD or, (2) The global parameter, located in the ODD, that indicates the number of tuples in the head cell relation is incorrect. If this error repeats over and over then manual action is required to either populate the corresponding static data or to update the global parameters.
Dump Description:	
<i>Error Address</i> =	Address of the head cell tuple.
<i>Bad Data</i> =	The relation identification of the head cell (owner) tuple.
<i>Good Data</i> =	The relation identification of the member tuple under recovery.
<i>Logical Index</i> =	Logical key of the head cell tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: TRAVERSE

Error Code:	TRAVERSE
Error Description:	A tuple linked to a bad tuple is being recovered.
Possible Error Effect:	A linked set of tuples is being recovered (possibly a call processing process).
Corrective Action Taken:	The audit driver will initialize the tuple and places it on the head cell (idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple found in error.
<i>Dump</i> =	The tuple found in error.

Table 11 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	A dummy (unused) tuple in some relation contains nonzero data. Since the dummy tuple is never used for anything, something has written into it in error.
Possible Error Effect:	Something is invalidly attempting to use the dummy tuple.
Corrective Action Taken:	The audit driver will zero the dummy tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of dummy tuple found in error..
<i>Bad Data</i> =	Data to be found non-zero in the dummy tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the dummy tuple found in error.
<i>Dump</i> =	The dummy tuple found in error.

Table 12 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after taking a system interrupt (such as, write protect, out-of-range read, and so forth).
Possible Error Effect:	May cause further resets, audits, and/or asserts.
Corrective Action Taken:	None, all data that would have caused the reset is reinitialized on the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the current tuple under audit.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the current tuple under audit.
<i>Dump</i> =	The current tuple under audit.

DRINASEK

Software Release: 5E10 and later

1. INDIRECT ACCESS SEQUENTIAL KEY DRIVER (DRINASEK)

The Indirect Access Sequential Key Driver (DRINASEK) steps through relations with tuples that have indirect sequential keys up to a minimum number of tuples for a given application. It then steps through nonsequential keys from the specified tuple to the end of the relation. The minimum number of tuples per application is office dependent. Unlike the Linkage Driver (DRLINK), DRINASEK steps through relations whose access type is indirect. The driver checks the syntactic data (linkage, quarantine state, and key) for each populated tuple. It then calls the appropriate semantic function.

The DRINASEK driver runs in both the Operational Kernel Process (OKP) in the Administrative Module (AM) and Switching Module (SM) environments in both the segmented and non-segmented modes. As of the 5E6(1) Generic the DRINASEK driver also runs in the Communication Module Processor (CMP) in both the segmented and non-segmented modes.

This driver performs existence checks (check for missing tuples and a vacant tuple 0), and linkage checks on each non-vacant tuple. Since the driver performs existence checks on each tuple, the structure must have tuples sequentially populated up to the minimum number of tuples required by the application. Vacant tuples can be present from this point to the bottom of the relation. This driver guarantees that the minimum number of tuples exist, where this number for any given application is office dependent data (ODD).

2. DATA STRUCTURES

The DRINASEK driver operates on relations with an indirect access type. The relations must have sequential keys starting with key value 1.

The DRINASEK driver also refers to the following data:

- AUPRI_arr[] is the audit drivers per relation information array containing information about the current relation under audit. The per relation information contains:
 - the number of headcells tuples,
 - the number of dummy (unused) tuples,
 - if the relation definition specifies a quarantine bit,
 - the size of one tuple,
 - the number of tuples in the relation,
 - the offset between logical and physical keys,
 - the beginning address of the relation,
 - the address of the semantic check function,
 - the address of the idle list function,
 - the address of the recovery function,
 - the address of the selective initialization check function,
 - the index into the linkage mapping array (AUptlb_arr[]),

- the relations access method,
 - the path and path type to traverse linkages to the owning process,
 - and the idle list mapping identification.
- AUPMlone_one[] is the audit drivers per mapping information array containing information about all one-to-one linkage mappings. The one-to-one per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the one-to-one linkage block in the near tuple,
 - and the offset to the start of the one-to-one linkage block in the far tuple.
- AUPMlmember[] is the audit drivers per mapping information array containing information about one-to-many linkage mappings. In this mapping the near relation is the member tuple and the far relation is the owner tuple. The one-to-many per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the member linkage block in the near tuple,
 - and the offset to the start of the owner linkage block in the far tuple.
- AUPMlowner[] is the audit drivers per mapping information array containing information about one-to-many linkage mappings. In this mapping the near relation is the owner tuple and the far relation is the member tuple. The one-to-many per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the owner linkage block in the near tuple,
 - and the offset to the start of the member linkage block in the far tuple.
- AUptlb_arr[] is the per tuple linkage block array containing all one-to-one and one-to-many linkage mappings for a given relation.

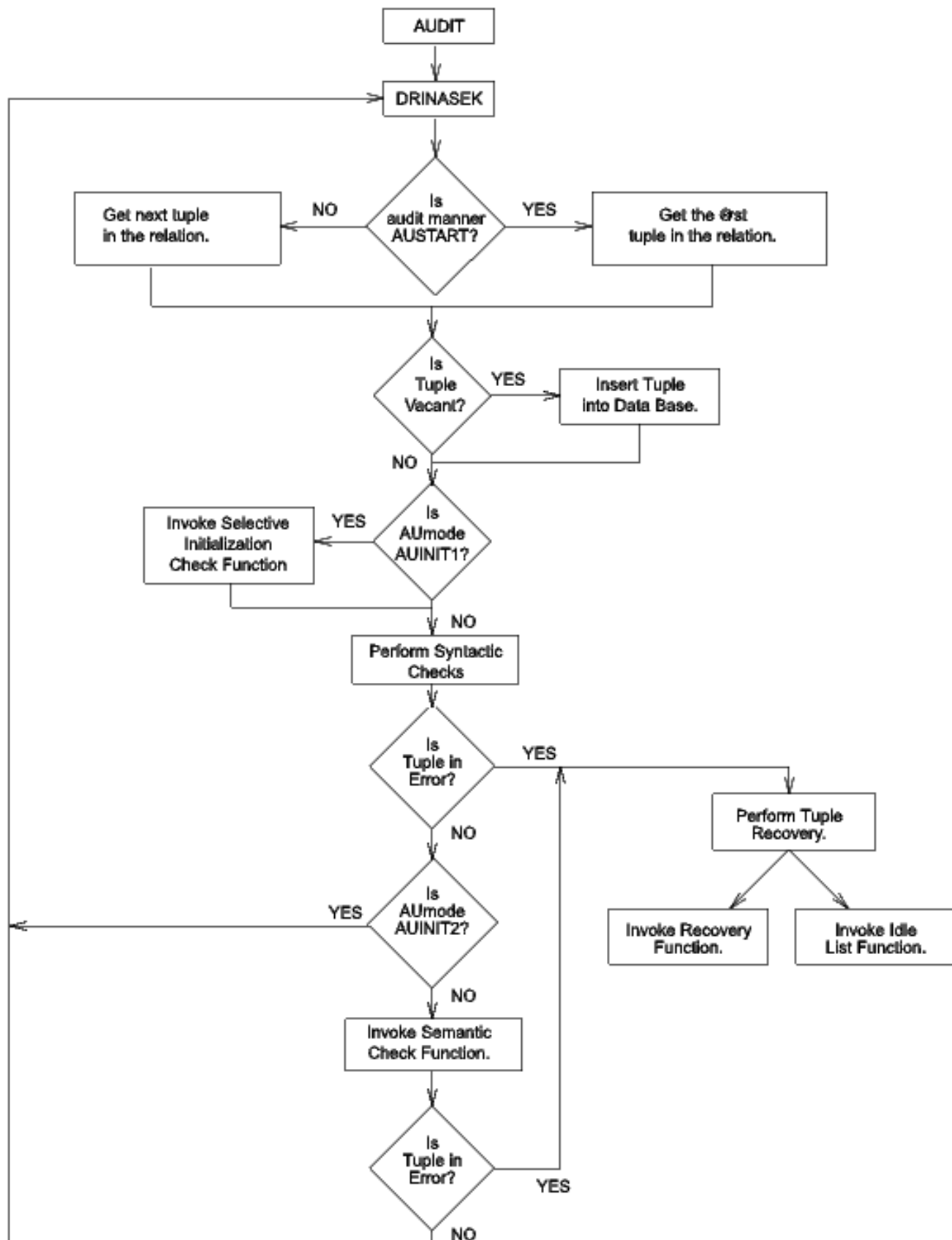


Figure 1 Indirect Access Sequential Key Driver

3. AUDIT CHECKS

The processing flow for the DRINASEK driver is shown in Figure 1 and is explained.

The invoking audit calls the DRINASEK driver and passes to it the audit control information (manner) and the relation identification.

- The DRINASEK driver will check if the audit control information (manner) indicates that the current running audit has taken a system interrupt (such as, wild write, invalid operation, and so forth). If this has occurred the audit will zero the tuple. The tuple will then be recovered on the next invocation of the audit.
- Before accessing a tuple in the relation, the DRINASEK driver checks to ensure that sufficient time remains in the segment to perform the necessary checks by invoking the audit control real time break function. The audit control real time break function will ensure that the DRINASEK driver takes segment breaks every 20 milliseconds during normal operation and 40 milliseconds during recovery. If enough time is remaining the tuple is access using Data Base Manager (DBM) primitives.
- The driver sequences through the relation detecting and inserting missing tuples between key value 1 and the minimum value. The minimum key value is determined by invoking an internal DRINASEK driver function.

For the tuples between 1 and N, the audit takes the following actions. If a tuple is found but some previous tuples are missing, the existing tuple is audited before control is returned to the invoking audit to recover the missing tuples. If consecutive vacant tuples are found, control returns to the invoking audit and the tuples are recovered.

- After the existence check is performed, the DRINASEK driver will check if the tuple is a head cell (idle list for another relation) tuple by using data in the per relation information (AUPRI_arr.auNO_HC). If this is a head cell tuple the DRINASEK driver will perform only syntactic checks on this tuple and not semantic checks.
- If the audit control information indicates that the audit was entered in the initialization one mode (AUMode = AUNIT1) and this is not head cell tuple, the DRINASEK driver will invoke the selective initialization function (AUPRI_arr[].sck_fcn) to check if the tuple under audit is in a transient state.
- If the tuple not a head cell tuple the DRINASEK driver will then perform syntactic type checking on this tuple. The DRINASEK driver will check the integrity of the tuple's key, quarantine state, and all one-to-one and one-to-many linkages.
- If all syntactic checks pass the DRINASEK driver will invoke then invoke the semantic check function (AUPRI_arr[].sem_fcn). The semantic check function will not be invoked if the audit is in the initialization two (AUMode = AUNIT2) mode. The semantic check function will check the internal consistency of the tuple under audit. If the semantic check function passes all checks the DRINASEK driver will access the next tuple and repeat all the checks.
- If any of the checks fail the DRINASEK driver will initiate recovery. If the DRINASEK driver can traverse the linkage of the tuple back to the owning process the DRINASEK driver will return to audit control to perform a single process purge (SPP) on this process. After the SPP the DRINASEK driver will invoke the audit's recovery function (AUPRI_arr[].rec_fcn) to zero the tuple and initialize certain fields (include the logical key). The DRINASEK driver will then invoke the audits' idle list function (AUPRI_arr[].idle_fcn) to retrieve information on the tuples idle list head cell. The driver will use this information to link the tuple to the appropriate idle list.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

DRINDEX

Software Release: 5E10 and later

1. INDIRECT ACCESS DRIVER (DRINDEX)

The indirect access driver (DRINDEX) is used by audits which must sequence through indirect access relations with non-sequential keys. This driver provides looping control, tuple retrieval, tuple syntactic (key, quarantine, and linkage) checking, database manager (DBM) return code checking, and audit control parameter checking. The DRINDEX driver runs in both the operational kernel process (OKP) in the Administrative Module (AM) and Switching Module (SM) environments in both segmented and non-segmented modes. As of the 5E6(1) Generic the DRINDEX driver also runs in the Communications Module Processor (CMP) in both segmented and non-segmented modes.

2. DATA STRUCTURES

The DRINDEX driver operates on relations with an indirect access type. The relations must have non-sequential keys.

The DRINDEX driver also refers to the following data:

- AUPRI_arr[] is the audit drivers per relation information array containing information about the current relation under audit. The per relation information contains:
 - the number of headcells tuples,
 - the number of dummy (unused) tuples,
 - if the relation definition specifies a quarantine bit,
 - the size of one tuple,
 - the number of tuples in the relation,
 - the offset between logical and physical keys,
 - the beginning address of the relation,
 - the address of the semantic check function,
 - the address of the idle list function,
 - the address of the recovery function,
 - the address of the selective initialization check function,
 - the index into the linkage mapping array (AUptlb_arr[]),
 - the relations access method,
 - the path and path type to traverse linkages to the owning process,
 - and the idle list mapping identification.
- AUPMlone_one[] is the audit drivers per mapping information array containing information about all one-to-one linkage mappings. The one-to-one per mapping informations contains:
 - the near relation identification,

- the far relation identification,
 - the offset to the start of the one-to-one linkage block in the near tuple,
 - and the offset to the start of the one-to-one linkage block in the far tuple.
-
- AUPMImember[] is the audit drivers per mapping information array containing information about one-to-many linkage mappings. In this mapping the near relation is the member tuple and the far relation is the owner tuple. The one-to-many per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the member linkage block in the near tuple,
 - and the offset to the start of the owner linkage block in the far tuple.
-
- AUPMlowner[] is the audit drivers per mapping information array containing information about one-to-many linkage mappings. In this mapping the near relation is the owner tuple and the far relation is the member tuple. The one-to-many per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the owner linkage block in the near tuple,
 - and the offset to the start of the member linkage block in the far tuple.
-
- AUptlb_arr[] is the per tuple linkage block array containing all one-to-one and one-to-many linkage mappings for a given relation.

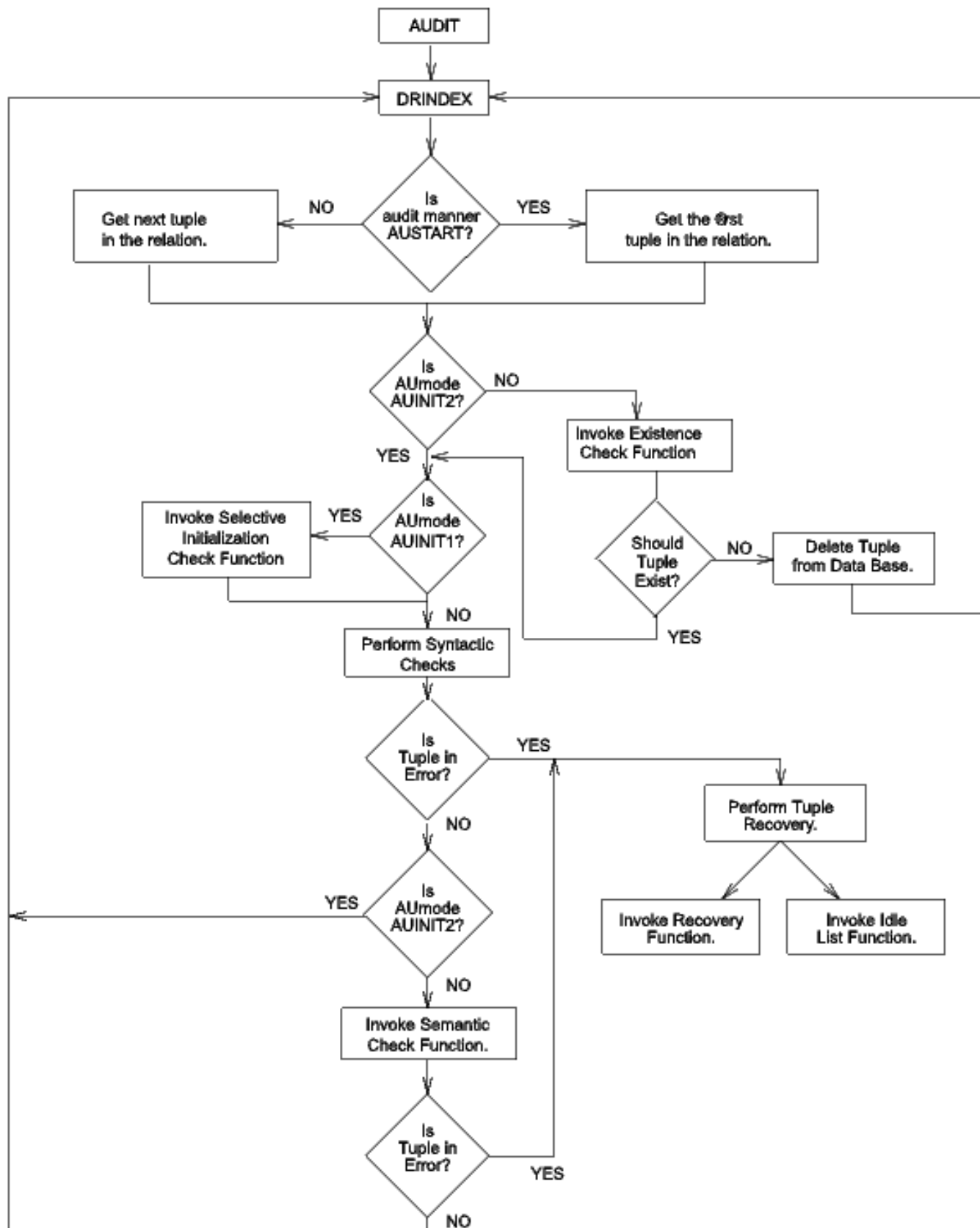


Figure 1 Processing Flow

3. AUDIT CHECKS

The processing flow for the DRINDEX driver is shown in Figure 1 .

- (a) The invoking audit calls the DRINDEX driver and passes to it the audit control information (manner), the relation identification, and a pointer to the existence check function.
- (b) The DRINDEX driver will check if the audit control information parameter (manner) indicates that the current running audit has taken a system interrupt (such as, write protect, invalid operation, and so forth). If this has occurred the audit will zero the tuple. This tuple will then be recovered on the next invocation of the audit.
- (c) Before accessing a tuple in the relation, the DRINDEX driver checks to ensure that sufficient time remains in the segment to perform the necessary checks by invoking the audit control real time break function. The audit control real time break function will ensure that the DRINDEX driver takes segment breaks every 20 milliseconds during normal operation and 40 milliseconds during recovery. If enough time is remaining the tuple is accessed.
- (d) Once the tuple is accessed the existence check function is invoked to indicate if the dynamic tuple should or should not exist. This function is not called if the DRINDEX driver is invoked in the initialization two mode (AUmode = AUINIT2). If the existence check function return code indicates that the tuple should not exist, the DRINDEX driver ensures that the tuple is not in the database. If the tuple should exist the DRINDEX driver ensures that the tuple is in the data base and then performs all syntactic checking.
- (e) After the existence check is performed, the DRINDEX driver will check if the tuple is a head cell (idle list for another relation) tuple by using data in the per relation information (AUPRI_arr.auNO_HC). If this is a head cell tuple the DRINDEX driver will perform only syntactic checks on this tuple and not semantic checks.
- (f) If the audit control information indicates that the audit was entered in the initialization one mode (AUmode = AUINIT1) and this is not head cell tuple, the DRINDEX driver will invoke the selective initialization function (AUPRI_arr[].sck_fcn) to check if the tuple under audit is in a transient state.
- (g) If the tuple not a head cell tuple the DRINDEX driver will then perform syntactic type checking on this tuple. The DRINDEX driver will check the integrity of the tuple's key, quarantine state, and all one-to-one and one-to-many linkages.
- (h) If all syntactic checks pass the DRINDEX driver will invoke then invoke the semantic check function (AUPRI_arr[].sem_fcn). The semantic check function will not be invoked if the audit is in the initialization two (AUmode = AUINIT2) mode. The semantic check function will check the internal consistency of the tuple under audit. If the semantic check function passes all checks the DRINDEX driver will access the next tuple and repeat all the above checks.
- (i) If any of the above checks fail the DRINDEX driver will initiate recovery. If the DRINDEX driver can traverse the linkage of the tuple back to the owning process the DRINDEX driver will return to audit control to perform a single process purge (SPP) on this process. After the SPP the DRINDEX driver will invoke the audit's recovery function (AUPRI_arr[].rec_fcn) to zero the tuple and initialize certain fields (include the logical key). The DRINDEX driver will then invoke the audits' idle list function (AUPRI_arr[].idle_fcn) to retrieve information on the tuples idle list head cell. The driver will use this information to link the tuple to the appropriate idle list.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not

found.

DRLINK

Software Release: 5E10 and later

1. LINKAGE DRIVER (DRLINK)

The linkage driver (DRLINK) is used by audits which must sequence through direct access relations. This driver provides looping control, tuple retrieval, tuple syntactic checking, and audit control parameter checking. The linkages are defined as syntactic data, as explained in the Software Overview section. Because the structure of this data is the same for all the tuples in the relations audited by the DRLINK driver, common code (the driver) is used to audit. The DRLINK driver runs in both the operational kernel process (OKP) in the administrative module (AM) and switching module (SM) environments in both segmented and non-segmented modes. The DRLINK driver also runs in the communications module processor (CMP) in both segmented and non-segmented modes.

2. DATA STRUCTURES

The DRLINK driver operates on relations with a direct access type. The relation must have sequential keys.

The DRLINK driver also refers to the following data:

- AUPRI_arr[] is the audit drivers per relation information array containing information about the current relation under audit. The per relation information contains:
 - the number of headcells tuples,
 - the number of dummy (unused) tuples,
 - if the relation definition specifies a quarantine bit,
 - the size of one tuple,
 - the number of tuples in the relation,
 - the offset between logical and physical keys,
 - the beginning address of the relation,
 - the address of the semantic check function,
 - the address of the idle list function,
 - the address of the recovery function,
 - the address of the selective initialization check function,
 - the index into the linkage mapping array (AUptlb_arr[]),
 - the relations access method,
 - the path and path type to traverse linkages to the owning process,
 - and the idle list mapping identification.
- AUPMlone_one[] is the audit drivers per mapping information array containing information about all one-to-one linkage mappings. The one-to-one per mapping informations contains:
 - the near relation identification,

- the far relation identification,
 - the offset to the start of the one-to-one linkage block in the near tuple,
 - and the offset to the start of the one-to-one linkage block in the far tuple.
-
- AUPMImember[] is the audit drivers per mapping information array containing information about one-to-many linkage mappings. In this mapping the near relation is the member tuple and the far relation is the owner tuple. The one-to-many per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the member linkage block in the near tuple,
 - and the offset to the start of the owner linkage block in the far tuple.
-
- AUPMlowner[] is the audit drivers per mapping information array containing information about one-to-many linkage mappings. In this mapping the near relation is the owner tuple and the far relation is the member tuple. The one-to-many per mapping informations contains:
 - the near relation identification,
 - the far relation identification,
 - the offset to the start of the owner linkage block in the near tuple,
 - and the offset to the start of the member linkage block in the far tuple.
-
- AUptlb_arr[] is the per tuple linkage block array containing all one-to-one and one-to-many linkage mappings for a given relation.

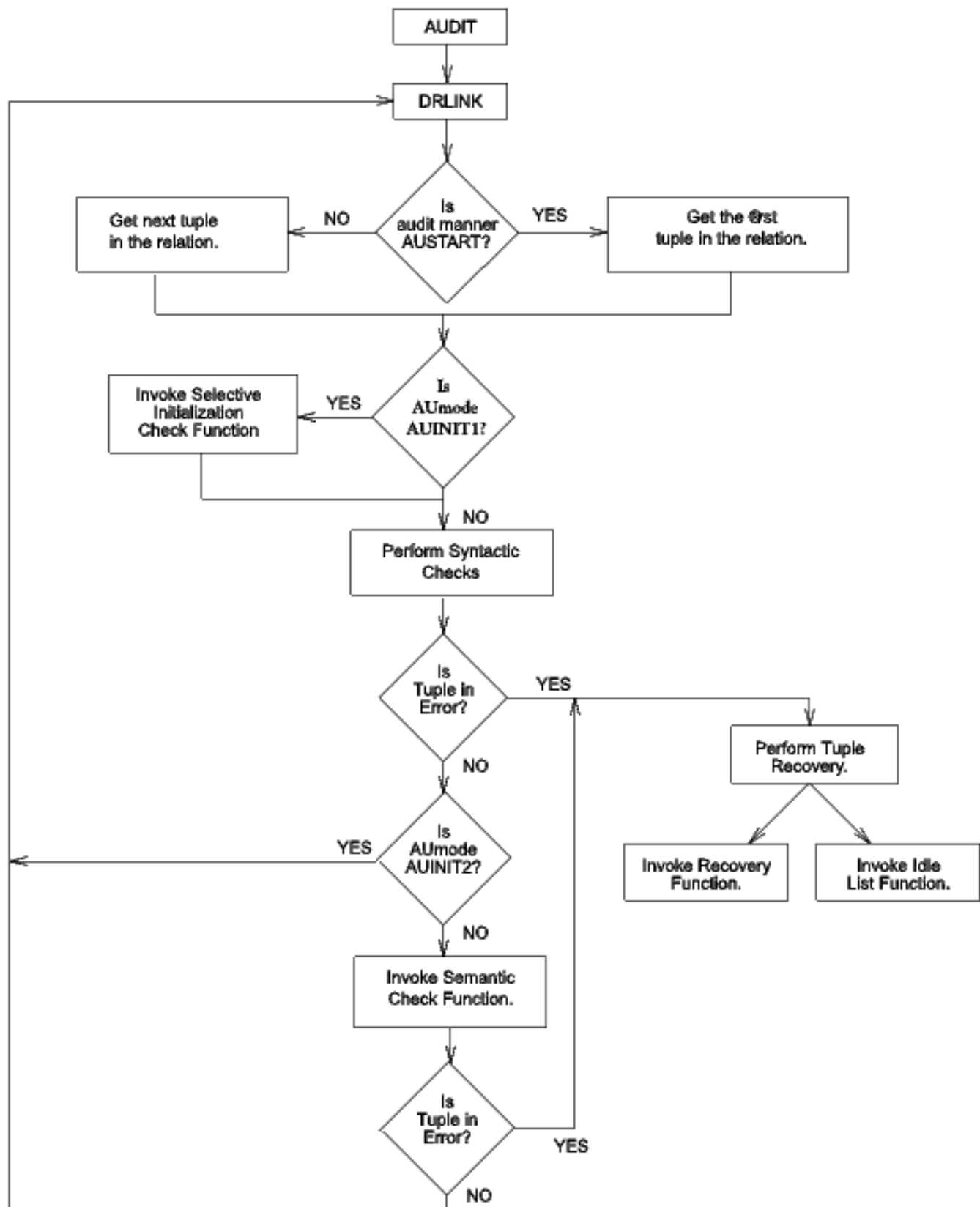


Figure 1 Linkage Driver

3. AUDIT CHECKS

The processing flow for the DRLINK driver is shown in Figure 1 and is explained below:

- The invoking audit calls the DRLINK driver and passes to it the audit control information (manner) and the relation identification.
- The DRLINK driver will check if the audit control information parameter (manner) indicates that the current running audit has taken a system interrupt (such as, wild write, invalid operation, and so forth). If this has occurred the audit will zero the tuple. This tuple will then be recovered on the next invocation of the audit.
- Before accessing a tuple in the relation, the DRLINK driver checks to ensure that sufficient time remains in the segment to perform the necessary checks by invoking the audit control real time break function. The audit control real time break function will ensure that the DRLINK driver takes segment breaks every 20 milliseconds during normal operation and 40 milliseconds during recovery. If enough time is remaining the tuple is accessed using the audit per relation information (AUPRI_arr[]).
- Once the tuple is accessed, the DRLINK driver will check if the tuple is a dummy (unused) tuple by using data in the per relation information (AUPRI_arr[].auNODUMMIES). If this is a dummy tuple the DRLINK driver will insure that this tuple contains all zeros.
- Next the DRLINK driver will check if the tuple is a head cell (idle list for another relation) tuple by using data in the per relation information (AUPRI_arr[].auNO_HC). If this is a head cell tuple the DRLINK driver will perform only syntactic checks on this tuple and not semantic checks.
- If the audit control information indicates that the audit was entered in the initialization one mode (AUMode = AUINIT1) and this is not a dummy or head cell tuple, the DRLINK driver will invoke the selective initialization function (AUPRI_arr[].sck_fcn) to check if the tuple under audit is in a transient state.
- If the tuple accessed is not a dummy or head cell tuple the DRLINK driver will then perform syntactic type checking on this tuple. The DRLINK driver will check the integrity of the tuple's key, quarantine state, and all one-to-one and one-to-many linkages.
- If all syntactic checks pass the DRLINK driver will then invoke the semantic check function (AUPRI_arr[].sem_fcn). The semantic check function will not be invoked if the audit is in the initialization two (AUMode = AUINIT2) mode. The semantic check function will check the internal consistency of the tuple under audit. If the semantic check function passes all checks the DRLINK driver will access the next tuple and repeat all the above checks.
- If any of the above checks fail the DRLINK driver will initiate recovery. If the DRLINK driver can traverse the linkage of the tuple back to the owning process the DRLINK driver will return to audit control to perform a Single Process Purge (SPP) on this process. After the SPP the DRLINK driver will invoke the audit's recovery function (AUPRI_arr[].rec_fcn) to zero the tuple and initialize certain fields (include the logical key). The DRLINK driver will then invoke the audits' idle list function (AUPRI_arr[].idle_fcn) to retrieve information on the tuples idle list head cell. The driver will use this information to link the tuple to the appropriate idle list.

4. ERROR HANDLING

See the Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

DRPHMAP

Software Release: 5E10 and later

1. PATH MAP GLOBAL ARRAY DRIVER (DRPHMAP)

The path map global array driver is used to retrieve entries in two global arrays: the peripheral control *SLC*[®] 96 data path array (pcS96PATH) and the peripheral control *SLC*[®] 96 mapping array (pcS96MAP). Only one audit, the path map audit (PATHMAP) in the peripheral control audits subgroup, uses DRPHMAP. The driver serves as the intermediary between the PATHMAP audit and the audit control program. It schedules segment breaks and keeps track of the array entries the audit checks. The driver operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and initialization modes.

2. DATA STRUCTURES

The path map global array driver checks two peripheral control global arrays used by the PATHMAP audit: pcS96PATH and pcS96MAP. Containing the names of available *SLC*[®] 96 digital facilities interfaces (S-DFIs), the data path array describes the peripheral interface data bus (PIDB) time slots through the digital carrier line units (DCLUs) in the switching module to the module processors. The mapping array is a bit representation of pcS96PATH; it contains a busy/idle flag for each field in pcS96PATH.

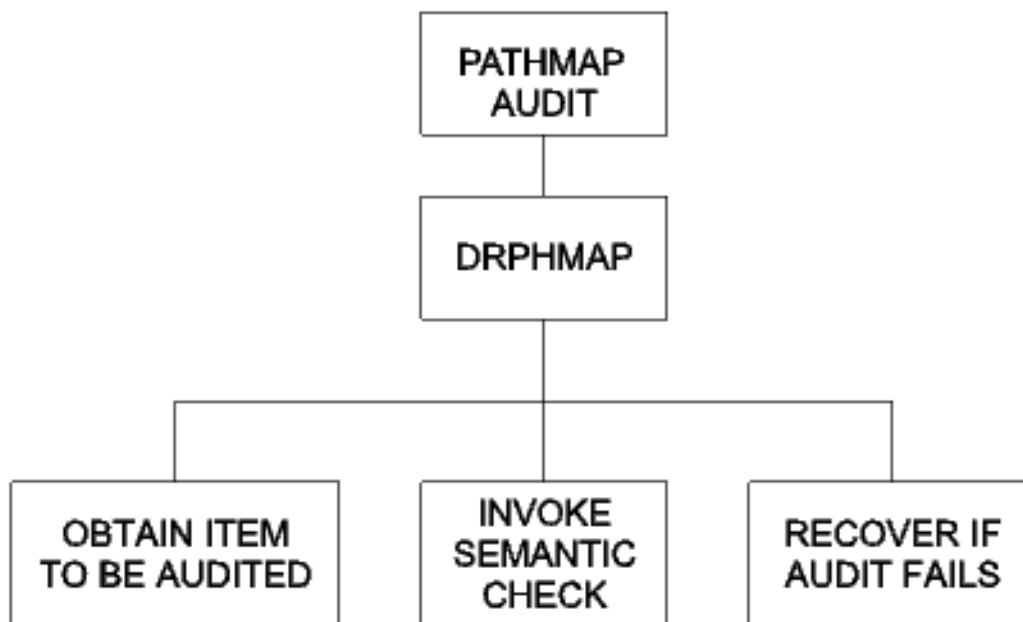


Figure 1 Path Map Global Array Driver

3. AUDIT CHECKS

The processing flow for DRPHMAP is shown in Figure 1 .

- (a) The invoking audit (PATHMAP) calls the driver, and passes to it the pointers to the item number and the *SLC*[®] 96 digital carrier line unit (DCLU) number of the previous audit segment.
- (b) The driver loops through pcS96MAP and pcS96PATH for the number of items to be checked per real-time segment. It calls a function to obtain the next entries from the two global arrays, and calls the semantic check function of the PATHMAP audit.

- (c) At the end of a segment, the driver saves the item number and DCLU number of the last entry obtained so it can continue checking the array during the next segment. If the driver has gone through all the entries in the array, it returns a completion message to the PATHMAP audit.

When the DRPHMAP driver cannot obtain an entry in either array, a failure message is returned to the PATHMAP audit. The audit assumes that all the entries have been checked, and returns control to the audit control program.

4. ERROR HANDLING

When the DRPHMAP driver cannot obtain an entry in either array, a failure message is returned to the PATHMAP audit. The audit assumes that all the entries have been checked, and returns control to the audit control program. When the semantic check fails, the driver calls the PATHMAP recovery function. See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

DRSTATIC

Software Release: 5E10 and later

1. STATIC DRIVER (DRSTATIC)

The static driver (DRSTATIC) reads successive tuples of relations in the database. Unlike the linkage driver (DRLINK), the DRSTATIC driver can audit static or dynamic relations with non-sequential keys. The DRSTATIC driver runs in both the operation kernel process (OKP) in the administrative module (AM) and the switching module (SM) environments in both the segmented and non-segmented modes. The DRSTATIC driver also runs in the communications module processor (CMP) in both segmented and non-segmented modes.

2. DATA STRUCTURES

The DRSTATIC driver operates on dynamic or static relations with non-sequential keys. The DRSTATIC driver uses a global buffer (AUGtupbuf) to store the contents of the current tuple under audit.

3. AUDIT CHECKS

The processing flow for the DRSTATIC driver is shown in Figure 1 and is explained below:

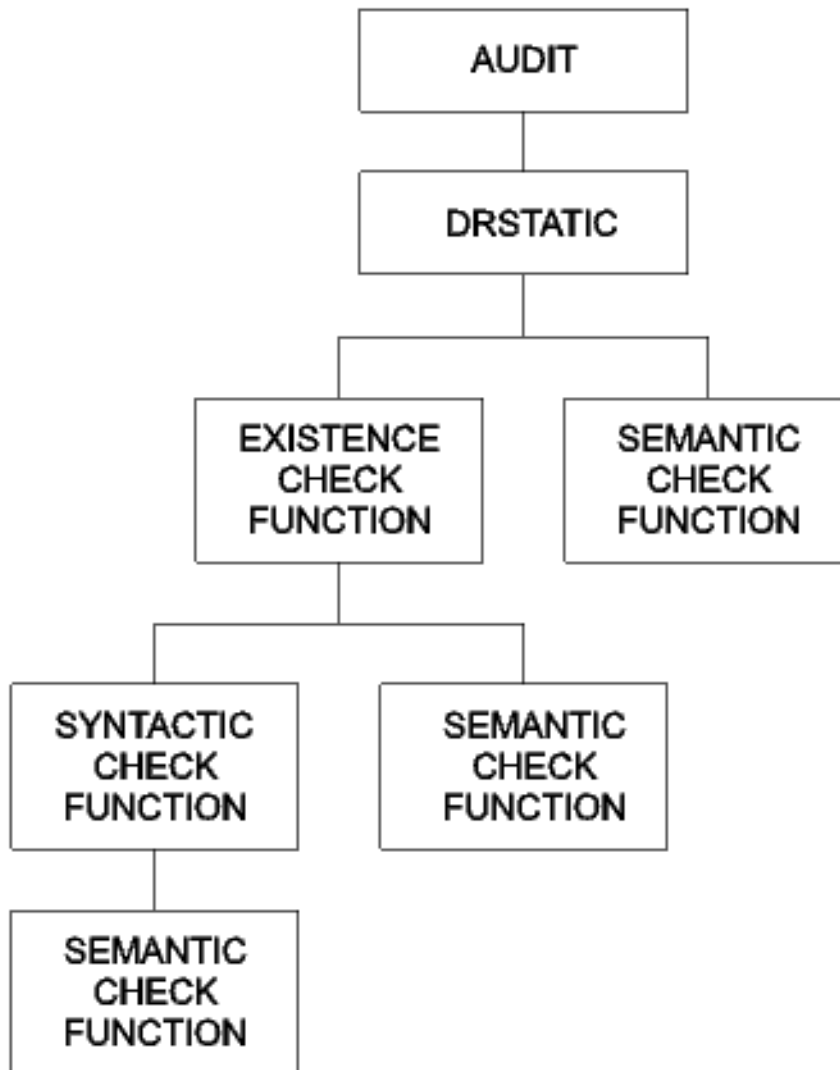


Figure 1 Static Driver

- The invoking audit calls the DRSTATIC driver and passes to it the audit control information (manner), the relation identification, a pointer to a buffer to store the contents of one tuple, and a pointer to the semantic check function.
- The DRSTATIC driver will check if the audit control information parameter (manner) indicates that the current running audit has taken a system interrupt (such as, write protect, invalid operation, and so forth). If this has occurred the audit will zero the tuple. This tuple will then be recovered on the next invocation of the audit.
- Before accessing a tuple in the relation, the DRSTATIC driver checks to ensure that sufficient time remains in the segment to perform the necessary checks by invoking the audit control real time break function. The audit control real time break function will ensure that the DRSTATIC driver takes segment breaks every 20 milliseconds during normal operation and 40 milliseconds during recovery. If enough time in the segment is remaining the tuple is accessed.
- Once the tuple is accessed the semantic check function is invoked. If the semantic check function fails, recovery is handled by the invoking audit and not the DRSTATIC driver.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for all DRSTATIC error codes.

DRTRAV

Software Release: 5E10 and later

1. TRAVERSE DRIVER (DRTRAV)

The traverse driver (DRTRAV) is used to sequence through relations that consist of set-member lists. The DRTRAV driver is used as a common function; it provides the interface between the audit and audit control, and contains information on how to proceed from one tuple to the next. The DRTRAV driver invokes a semantic check function for each tuple. If the driver takes a segment break, it saves the successor (key) of the last tuple audited. Only one audit, the hunted idle port audit (IDLPR), invokes the DRTRAV driver. It runs in both the operational kernel process (OKP) and operating system for distributed switching-switching module (OSDS-SM) environments and is invoked by IDLPRT in the segmented, nonsegmented, and initialization modes.

2. DATA STRUCTURES

The DRTRAV driver operates on relations that consist of set-member linked lists. It obtains the per-mapping and per-relation information from the invoking audit, unlike the DRLINK and DRSTATIC drivers, which obtain the information from a global data file.

3. AUDIT CHECKS

The processing flow for the DRTRAV driver is shown in Figure 1 and is explained:

- (a) The invoking audit calls the DRTRAV driver and passes to it the manner (ZAP, START, RESUME) in which it should be run. The audit also passes a structure containing the relation ID, the number of tuples to audit per segment, semantic check function, mapping information, number of lists to audit, and keys of the lists (headcells) to the driver.
- (b) If the manner is START or RESUME, the DRTRAV driver does one of the following: starts with the first list and obtains the first tuple from that list, traverses to the next tuple in the list, or moves on to the next list.

If the DRTRAV driver is invoked in the ZAP manner, an error is reported, the tuple reconstruction function is called, and the driver returns a completion message to the invoking audit.
- (c) The invoking audit passes the mode to the DRTRAV driver, which tells it how many tuples to audit per segment.
- (d) For each tuple successfully traversed, the DRTRAV driver calls the semantic check function. When the driver takes a segment break and the maximum number of tuples to audit per segment is met, the driver checks to see if it is at the end of the relation. If it is not at the end of the relation, the driver saves the successor (key) of the last tuple audited. If it is at the end of the relation, the driver returns a completion message to the invoking audit.

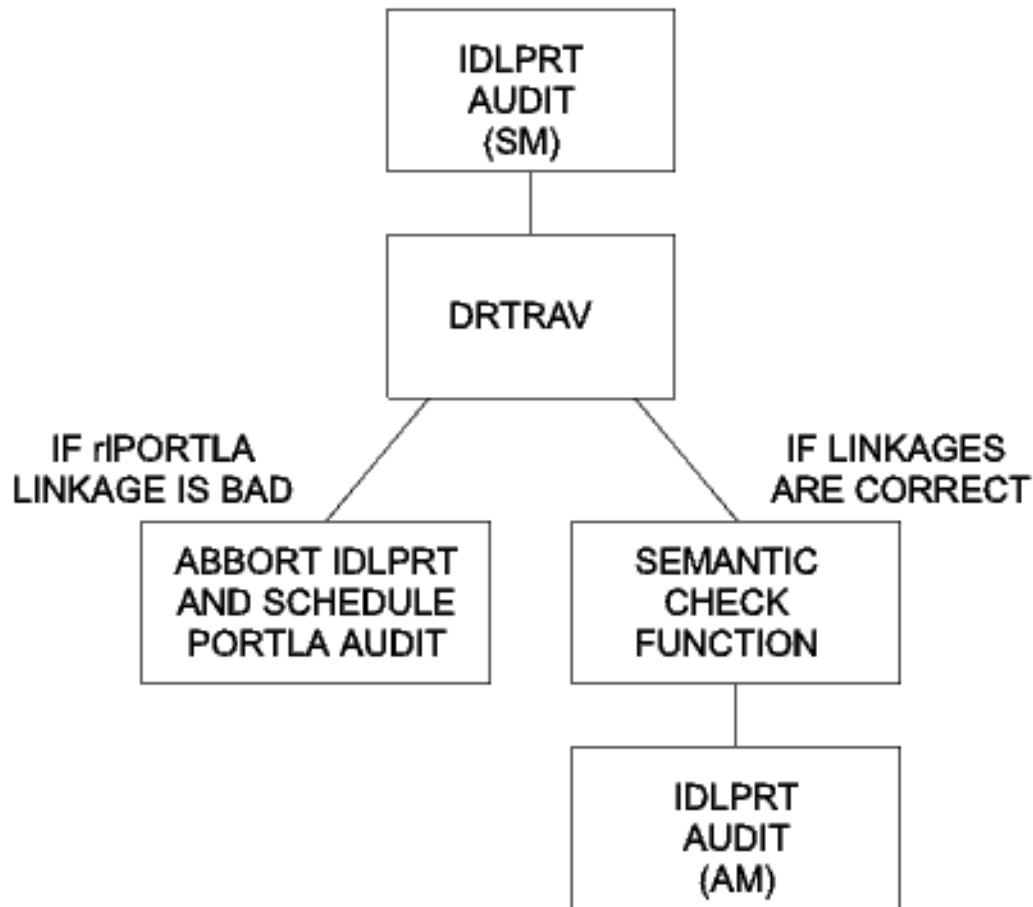


Figure 1 Traverse Driver

4. ERROR HANDLING

When the DRTRAV driver cannot traverse to a valid tuple (that is, a linkage check fails) or the semantic check fails, a failure message is returned to the invoking audit.

A failure in the IDLPRT audit causes the port linkage area audit (PORTLA) to be run in the elevated, segmented mode. See driver error codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

6. Audit Descriptions:
A - AP

AATA

Software Release: 5E10 and later

1. AUTOPLEX[®] AUTOMATIC TASK ADMINISTRATOR (AATA) AUDIT

The advanced mobile phone system (AMPS) automatic task scheduler is a system process (SP) that resides in the administrative module (AM) for all AMPS applications.

Its purpose is to distribute terminal maintenance OA&M messages that come in from the executive control processor (ECP) and are transported by the data link network (DLN) to the appropriate application functional area for processing.

The purpose of the AATA audit is to examine each entry in the process control block (PCB) for an AATA parent, then validate that the process has a slot in the TMAATSKBUF.

Any process that purports to have any AATA parent and has no slot in the TMAATSKBUF is removed. The AATA audit runs in the operational kernel process (OKP).

2. DATA STRUCTURES

The AATA-SP task buffer is an array in the OKP of up to twenty *TMAATSKBUF* slots, each one containing information about an active or pending task and the process ID (*atep_pid*) associated with the task. The AATA-SP checks the validity of the TMAATSKBUF.

The PCB contains both *OSbpcb[]* and *OSbprogstate[]* arrays. These arrays contain information about the process. This audit uses the *OSbpcb[].state* and *OSbprogstate[].parent* to determine whether the TMAATSKBUF needs to be have a slot for the process.

3. AUDIT CHECKS

The AATA audit uses no audit drivers; it loops through the PCBs looking for processes whose parent is the AATA-SP. The DEADPID error represents a PCB whose parent is an AATA-SP but has no entry in TMAATSKBUF. In particular, it ensures that the process (*atep_pid*) in the TMAATSKBUF slot still exists. The AATA audit runs in the OKP.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The process owning the PCB is parented by the AATA-SP, but there is no entry in TMAATSKBUF for the process. When the process is alive (<i>OSbpcb[].state != OSEMPY</i>) and it is owned by the AATA-SP, (<i>OSbprogstate[].parent == GLPGAATA</i>), the process must be in an <i>atep_pid</i> in TMAATSKBUF.
Possible Error Effect:	This process may not be needed any longer.
Corrective Action Taken:	The terminal process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The error code returned from the check of TMAATSKBUF.
<i>Good Data =</i>	The state of the process control block (<i>OSbpcb[].state</i>).
<i>Logical Index =</i>	The index to the PCB.

<i>Dump 1</i> =	The contents of the OSbpcb[] entry where the error was found.
<i>Dump 2</i> =	The process identifier of the process with no entry in TMAATSKBUF.

Table 2 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit is being entered to provide recovery after taking a reset. When a reset occurs (usually triggered by an illegal or out of range memory access), the audit which was running at the time is aborted, and is then rescheduled to run in AUZAP mode to provide recovery of related data left inconsistent as a result of the reset.
Possible Error Effect:	The audit may continue to cause memory access violations, increasing the probability of drastic recovery action and preventing the audit from checking for further errors.
Corrective Action Taken:	The data which caused the reset is initialized by the next entry of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index to the PCB.
<i>Dump</i> =	The contents of the OSbpcb[] entry running when the interrupt occurred.

ACAHSS

Software Release: 5E10 and later

1. AUTOMATIC CALL ASSURANCE HUNDRED SECOND SCAN (ACAHSS) AUDIT

Automated call assurance and hundred second scan (ACA and HSS) data are measurements compiled on an hundred second basis in switching modules (SMs). These measurements are done on trunk members using 100 second scanning and consist of various data used to determine trunk usage (or non-usage) activity. Data used to store this activity and flags to determine whether a trunk member should report its usage activity are stored in the RLACA_HSS relation, which is audited by the ACAHSS audit. The ACAHSS audit runs in all SMs, in routine and elevated segmented modes.

2. DATA STRUCTURES

One tuple exists in the RLACA_HSS relation for every trunk group member populated in the SM that has the ACA and HSS feature active. The ACAHSS audit performs checks on RLACA_HSS semantic data based on the following fields:

ccthres	= Call count threshold, where <code>rlACA_HSS.ccthres == rIFC_TRUNK.shtcallcnt</code> .
grpnum	= A trunk group number associated with the trunk port, where <code>rlACA_HSS.grpnum == rIFC_TRUNK.tgroup</code> .
lhtthres	= Long holding time threshold, where <code>rlACA_HSS.lhtthres == rIFC_TRUNK.lhtthres</code> .
member	= A trunk group member number associated with the trunk port where <code>rlACA_HSS.member == rIFC_TRUNK.member</code> .
port	= The key of the <code>rlACA_HSS</code> tuple. For every <code>rlACA_HSS</code> tuple, there must exist a corresponding tuple in the <code>RLFC_TRUNK</code> static relation where <code>rlACA_HSS.port == rIFC_TRUNK.port.member</code> .
shtthres	= Short holding time threshold, where <code>rlACA_HSS.shtthres == rIFC_TRUNK.sht_thres</code> .

3. AUDIT CHECKS

For each tuple in the `rlACA_HSS` relation, the ACAHSS audit makes the following checks:

DYN_STAT	= Verify that the short holding time, long holding time, and call count threshold is valid in <code>RLACA_HSS</code> based on data in <code>RLFC_TRUNK</code> .
EXTRATUP	= Verify that there is a corresponding <code>rIFC_TRUNK</code> static tuple (where <code>rIFC_TUNK.port == rlACA_HSS.port</code>) and if there is a corresponding <code>rlRT_TRKG</code> tuple which is marked as having ACA and HSS measurements assigned (where <code>rlRT_TRKG.grpnum == rIFC_TRUNK.tgroup</code> and <code>rlRT_TRKG.nlactive == DBYES</code> or <code>rlRT_TRKG.sht_thres > 0</code> or <code>rlRT_TRKG.lht_thres > 0</code>).
INV_STATE	= Verify that the group and member data is valid in <code>RLACA_HSS</code> based on data in <code>rIFC_TRUNK</code> .
TRKG	= Verify that there is a corresponding tuple in the <code>RLRT_TRKG</code> relation (where <code>rlACA_HSS.grpnum == rlRT_TRKG.grpnum</code>).

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error

code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	An rIACA_HSS tuple was found with the short holding time, long holding time, and/or call count threshold in an invalid state. The shthres , lththres and ccthres fields in RLACA_HSS should match the sht_thres , lht_thres , and shtcallcnt fields respectively in the RLRT_TRKG static relation.
Possible Error Effect:	Asserts out of measurement scanning applications, and invalid trunk measurements.
Corrective Action Taken:	The shthres , lththres and ccthres fields are corrected to their counterparts in the RLRT_TRKG static relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACA_HSS tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIACA_HSS tuple (rIACA_HSS.port).
<i>Dump 1 =</i>	The contents of the rIACA_HSS tuple.
<i>Dump 2 =</i>	The contents of the associated tuple in the RLRT_TRKG relation.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An rIACA_HSS tuple was found with no corresponding tuple in the RLFC_TRUNK relation, or for which the corresponding tuple in RLRT_TRKG indicates that the ACA and HSS feature is not active on this particular trunk port. For every tuple in the RLACA_HSS relation, a corresponding rIFC_TRUNK static tuple must exist (where rIFC_TRUNK.port == rIACA_HSS.port) and a corresponding rIRT_TRKG tuple must exist which is marked as having ACA and HSS measurements assigned (where rIRT_TRKG.grpnum == rIFC_TRUNK.tgroup and rIRT_TRKG.nlactive == DBYES or rIRT_TRKG.sht_thres > 0 or rIRT_TRKG.lht_thres > 0).
Possible Error Effect:	Asserts out of measurement scanning applications, and invalid trunk measurements.
Corrective Action Taken:	The rIACA_HSS tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACA_HSS tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIACA_HSS tuple (rIACA_HSS.port).
<i>Dump =</i>	The contents of the rIACA_HSS tuple.

Table 3 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An rIACA_HSS tuple was found with invalid trunk group and member data. The grpnum and member fields in RLACA_HSS must match the tgroup and member fields in RLFC_TRUNK, where rIFC_TRUNK.port == rIACA_HSS.port.
Possible Error Effect:	Asserts out of measurement scanning applications, and invalid trunk measurements.
Corrective Action Taken:	The grpnum and member fields in RLACA_HSS are corrected to match tgroup and member in the rIFC_TRUNK static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACA_HSS tuple.
<i>Bad Data =</i>	The RLACA_HSS grpnum field << 16 the RLACA_HSS member field.
<i>Good Data =</i>	The RLFC_TRUNK tgroup field << 16 the RLFC_TRUNK member field.

<i>Logical Index</i> =	The key of the rIACA_HSS tuple (rIACA_HSS.port).
<i>Dump 1</i> =	The contents of the rIACA_HSS tuple.
<i>Dump 2</i> =	The contents of the associated tuple in the RLFC_TRUNK relation.

Table 4 Error Code: TRKG

Error Code:	TRKG
Error Description:	A rIFC_TRUNK static tuple was found for which no rIRT_TRKG tuple exists.
Possible Error Effect:	Trunk routing failures.
Corrective Action Taken:	None.
Manual Action Required:	Manual actions include: (1) Using ODBE, determine if the trunk group represented by the rIACA_HSS tuple key (trunk port) exists by reading the RLGROUP_PORT relation with rIGROUP_PORT.portgrp.grpnum set to the group number in RLFC_TRUNK (rIFC_TRUNK.tgroup). This information is dumped as the <i>Bad Data</i> portion of this message. Set the group type (rIGROUP_PORT.portgrp.grptype) to TM_TRK. Set the member number in RLGROUP_PORT to the member number which is dumped out as part of the rIACA_HSS tuple dumped with this message. (2) If a rIGROUP_PORT tuple exists, insert a corresponding tuple in the RLRT_TRKG relation with rIRT_TRKG.grpnum set to rIGROUP_PORT.portgrp.grpnum. (3) If a rIPORT_GROUP tuple does not exist, use Recent Change to Delete the ACA and HSS feature from this trunk port. Then remove the trunk port and insert it again to insure that the RLRT_TRKG and RLFC_TRUNK data is consistent. Verify that the rIGROUP_PORT static tuple is created. Finally, add the ACA and HSS feature to the trunk port, if desired.
Dump Description:	
<i>Error Address</i> =	The address of the rIACA_HSS tuple.
<i>Bad Data</i> =	The tggroup field in the rIFC_TRUNK static tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIACA_HSS tuple (rIACA_HSS.port).
<i>Dump 1</i> =	The contents of the rIACA_HSS tuple.
<i>Dump 2</i> =	The contents of the associated tuple in the RLFC_TRUNK relation.

ACB

Software Release: 5E10 and later

1. ALLOCATION CONTROL BLOCK (ACB) AUDIT

An OSDS process may dynamically request memory allocation. An allocation control block (ACB) is associated with each allocatable block of memory. When a request is made, if enough successive blocks are available, the request is honored by linking the first ACB in the series of blocks to the process control block (PCB). Otherwise the request is denied. Memory deallocation reverses this procedure by moving the blocks off of the process's list and onto the ACB free list.

2. DATA STRUCTURES

In addition to the ACBs themselves (OSacb[]), the audit verifies the following global data:

- The process to ACB linkage fields in the PCBs (rIOSPCB.acb0 and **acbn** in the SM, otherwise OSbpcb[].acb0 and **acbn**).
- The global indexes OSffacb and OSlfacb which indicate the first and last members of the list of available blocks of allocatable memory.
- The memory blocks associated with the ACBs (OSacbbuf[]).

3. AUDIT CHECKS

The ACB audit does not use any audit driver. The following list gives a basic indication of the order of the checks made by this audit.

If the audit is being (re)entered:

ZAP = Check for system interrupt handling.

If the audit is verifying the idle list:

ALNK2QUE = Semantic check of OSlfacb.

BADOWNER = Check ACB.proc.

BLNKINQUE = Semantic check of OSffacb.

LNK2ALST = Range check OSlfacb.

LNKIALST = Range check idle ACB's **llink** field.

PT2PT = Point-to/point-back check between ACBs.

If the audit is verifying an ACB list attached to a process:

ALNK2QUE = Semantic check of PCB.acbn.

BLNKINQUE = Semantic check of PCB.acb0.

LNK2ALST = Range check PCB.acb0.

LNKIALST = Range check ACB.llink.
 PT2PT = Point-to/point-back check between ACBs.

If the audit is doing semantic checks on all ACBs:

GRPINFO = Default value checks for ACB fields.
 INV_LENGTH = Semantic check for the 0th ACB.
 MEMBAD = ACB/OSacbbuf[] cross checks.

If the audit is looking for lost ACBs:

INV = Check for lost ACBs.
 INVOWNER = Check potentially lost ACB's **proc** field.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALNK2QUE

Error Code:	ALNK2QUE
Error Description:	An index to the last element in a list of ACBs (allocation control blocks) (OSIfacb if the free list is being checked; otherwise, PRGSTATE.acbn for some non-idle process) indicates an ACB which either has an incorrect owner (not OSNULL for the free list, different from the process number in all other cases) or has a non-null (not OSNULL) rlink field.
Possible Error Effect:	The list may not be accessed correctly. For the free list, this could mean lost ACBs (leading to overload situations) or non-idle ACBs being taken for new processes (affecting the process to which they rightfully belong). For ACB lists attached to PCBs (process control blocks), this problem could cause the owning process to lose messages or falsely believe it has messages which actually belong to the idle list or other processes. In practice, because of defensive checks in the OS (operating system), this condition is much more likely to result in asserts from the OS which schedules the ACB audit in the directed mode. It is also likely that ACB "INV" errors will be seen along with this one.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. Otherwise, if the list is the free list, the audit will rebuild it by verifying OSfifacb (index to the first free ACB) and, if it is correct, verifying the list from the top down as far as possible. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	The owner of the list which the audit is trying to access (either OSNULL or a PCB index).
<i>Good Data</i> =	Contents of the ACB's owner field.
<i>Logical Key</i> =	The ACB's index (= OSIfacb in the free list case, = PRGSTATE.acbn otherwise).
<i>Dump 1</i> =	The ACB.
<i>Dump 2</i> =	The owning PCB or nothing, if the free list was being checked.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	An ACB's (allocation control block's) proc field shows a different owner than the owner of the list in which it was found. Note that the ACB's rlink field has already been verified (PT2PT) and its llink has been checked to be in range (BLNKINQUE/LNKIALST).
Possible Error Effect:	If the owning list belongs to a PCB, then the associated process is tying up memory which could have been allocated to another process. Regardless of where the ACB was found, it is possible that it actually belongs on a list belonging to some process, in which case that process will be missing some of its allocated memory. If the ACB was found on the free list, then INV errors are also likely to show up.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. If this is the free list of ACBs, relink the list around the bad ACB and let an INV error fire to put it back on the idle list, if that is where it really belongs. If the list is attached to a PCB (process control block), the associated process will be purged. Although the offending ACB may actually belong somewhere else and the remaining ACBs on this list are correct, this list is still corrupted and can not be trusted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	Contents of the ACB's proc field.
<i>Good Data</i> =	The owner of the list the ACB was found in (either OSNULL or a PCB index).
<i>Logical Key</i> =	The ACB's index (= OSIfacb in the free list case, = PRGSTATE.acbn otherwise).
<i>Dump</i> =	The ACB.

Table 3 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	While traversing a list (bottom up), an ACB was found with its llink field = OSNULL, but was not the first ACB in the list, according to the list owner. Specifically, if the ACB was found on the free list, it was not indicated by OSffacb (index to the first free ACB). If the ACB was found on a list belonging to a PCB it was not indicated by the process's PRGSTATE.acb0 (index to the first ACB belonging to the process).
Possible Error Effect:	The list is either broken, or the list owner's index to the first element is corrupt. This could have any of the effects of an ALNK2QUE, LNKIALST, or PT2PT error.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. Otherwise, if the list is the free list, the audit will rebuild it by verifying OSffacb (index to the first free ACB) and, if it is correct, verifying the list from the top down as far as possible.

	If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	Index of the ACB, the list's owner says is first on the list.
<i>Good Data</i> =	The owner of the list the ACB was found in (either OSNULL or a PCB index).
<i>Logical Key</i> =	The ACB's index.
<i>Dump 1</i> =	The ACB.
<i>Dump 2</i> =	The owning PCB or nothing, if the free list was being checked.

Table 4 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	An ACB is not the head of a block of memory (ACB.numblks = 0), but at least one of the following is also true: • ACB.rlink != OSNULL • ACB.llink != OSNULL • OSacbbuff[ACB_idx] != 0 • ACB.proc != owner of the block of memory this ACB is associated with (OSNULL for the free list, otherwise a process number).
Possible Error Effect:	The block of memory with which this ACB is associated is corrupted. If the block of memory is owned by a process, the effect depends on what the memory block is being used for. If the block is on the free list, there would be no effect until the block (or a part of it) was allocated.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. If the ACB is associated with a block of memory which is linked to a process, the audit will purge the process. Otherwise, the audit will reinitialize the ACB and reinsert it in the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	The owner of the block of memory this ACB is associated with (either OSNULL for the idle list or a PCB index).
<i>Good Data</i> =	Contents of the ACB's proc field.
<i>Logical Key</i> =	The ACB's index.
<i>Dump 1</i> =	The ACB.
<i>Dump 2</i> =	The owning PCB or nothing, if the free list was being checked.

Table 5 Error Code: INV

Error Code:	INV
Error Description:	An ACB has been found which is not associated with any PCB or the free list.
Possible Error Effect:	The ACB is lost and inaccessible to the system. If enough ACBs became lost, overload could occur and operating system (OS) performance would be severely affected.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. The ACB is reinitialized and linked to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	0 (Indicates that the audit did not find the ACB on any list).
<i>Good Data</i> =	1 (Indicates that the audit should have found the ACB on some list).
<i>Logical Key</i> =	The ACB's index.
<i>Dump</i> =	The ACB.

Table 6 Error Code: INV_LENGTH

Error Code:	INV_LENGTH
Error Description:	One of the following is true: - The ACB (allocation control block) being checked is the first ACB (index = 0) and should, therefore, always be the head of a block of memory, but is not (ACB.numblks = 0). - An ACB's numblks field contains an out-of-range value (< 0 or $\text{numblks} + \text{ACB index} > \text{OSACBNUM}$).
Possible Error Effect:	If the ACB is not idle, the memory allocated to the process with which it is associated is corrupted. The effect will depend on how the process is using that memory. Otherwise, ACBs may be allocated invalidly, causing usage-dependent problems for the processes receiving the ACBs.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. The audit sets ACB.numblks = 1. Later checks (such as INV) will invoke more recovery as needed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	ACB.numblks.
<i>Good Data</i> =	The ACB owner (ACB.proc).
<i>Logical Key</i> =	The ACB's index.
<i>Dump</i> =	The ACB.

Table 7 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	An ACB (allocation control block) that the audit did not see on any list has been found. Before taking an INV error, the audit will verify that the ACB is truly lost. For example, if the audit were running in the segmented mode, the ACB may have changed lists while the audit took a timebreak. In this case, the audit would not have seen the ACB before now. The audit uses the ACB's proc field to determine which list to check. This error says that the proc field is out-of-range [not OSNULL and not a PCB index].
Possible Error Effect:	The ACB is effectively lost and inaccessible to the system. If enough ACBs became lost, overload could occur, and OS performance would be severely affected.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. If possible, the ACB is unlinked from the list that it is in (the linkages of the ACBs to which it points are reset). In any case, the ACB is cleared and linked to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB.
<i>Bad Data</i> =	Contents of the ACB's proc field.
<i>Good Data</i> =	OSNULL.
<i>Logical Key</i> =	The ACB's index.
<i>Dump</i> =	The ACB.

Table 8 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Either the tail of an ACB (allocation control block) list was null when the head was not, or the tail of an ACB list is out-of-range. The list owner (see good data) is either a PCB or the free list (OSNULL). If it is a PCB, the out-of-range value is the process's PRGSTATE.acbn field. If the owner is the free list, the out-of-range value is the last free ACB index (OSlfacb).
Possible Error Effect:	If the owner of the list is a PCB, that process may fail to access memory allocated for it. The results of this depend on the process and intended use of the allocated memory. If the free list is involved, this could result in lost ACBs which could, eventually, lead to operating system for distributed switching (OSDS) ACB overloads.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. If the ACB list belongs to a PCB, that process will be purged. Otherwise, the audit will attempt to rebuild the idle list from the top (OSffacb) down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The out of range ACB index.
<i>Good Data =</i>	The list owner (a PCB index or OSNULL for the idle list).
<i>Logical Key =</i>	OSNULL.
<i>Dump =</i>	The PCB (if any) which owns the list.

Table 9 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	While traversing a list, an ACB (allocation control block) was found with an out-of-range llink field (not OSNULL and not an ACB index).
Possible Error Effect:	The list is broken. For the free list, this could mean lost ACBs (leading to overload situations). For ACB lists attached to PCBs (process control blocks), this problem could cause the owning process to lose allocated memory. In practice, because of defensive checks in OS (operating system), this condition is much more likely to result in asserts from OS which schedule the ACB audit in the directed mode. It is also likely that ACB INV errors will be seen along with this one.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. Otherwise, if the list is the free list, the audit will rebuild it by verifying OSffacb (index to the first free ACB) and, if it is correct, verifying the list from the top down as far as possible. It will then link this segment (top down) with the segment (bottom up) that the audit has already checked. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the ACB.
<i>Bad Data =</i>	The ACB's llink field.
<i>Good Data =</i>	OSACBNUM (llink should have been OSNULL or less than this value).
<i>Logical Key =</i>	The ACB index.
<i>Dump =</i>	The ACB.

Table 10 Error Code: MEMBAD

Error Code:	MEMBAD

Error Description:	An ACB is the head of a block of memory but one of the following is also true: - An allocation control block (ACB) is idle (ACB.proc = OSNULL), but the associated memory address (oSacbbuf[ACB_index]) is not null (!= 0). - A non-idle ACB (ACB.proc = process number) has an invalid associated memory address (OSacbbuf[ACB_index] != OSallpool + (ACB_index = OSALBLKSZ)).
Possible Error Effect:	The block of memory with which this ACB is associated is corrupted. If the block of memory is owned by a process, the effect depends on what the memory block is being used for. If the block is on the free list, there would be no effect until the block (or a part of it) was allocated.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. If the ACB is associated with a block of memory which is linked to a process, the audit will purge the process. Otherwise, the audit will reinitialize the ACB and reinsert it in the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the ACB.
<i>Bad Data =</i>	The associated memory address (OSacbbuf[ACB_index]).
<i>Good Data =</i>	The ACB's index.
<i>Logical Key =</i>	The ACB's index.
<i>Dump 1 =</i>	The ACB.
<i>Dump 2 =</i>	The owning PCB or nothing, if the free list was being checked.

Table 11 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	While traversing a list, an ACB was found whose llink field contained the index of an ACB whose rlink field was not equal to the first ACB's index.
Possible Error Effect:	The list is broken. For the free list, this could mean lost ACBs (leading to overload situations) or non-idle ACBs being taken for new processes (affecting the process to which they rightfully belong). For ACB lists attached to PCBs, this problem could cause the owning process to lose allocated memory or attempt to use memory allocated to another process. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the ACB audit in the directed mode. It is also likely that ACB INV errors will be seen along with this one.
Corrective Action Taken:	If the audit is running in segmented mode, it will escalate to directed mode. Otherwise, if the list is the free list, the audit will rebuild it by verifying OSffacb (index to the first free ACB) and, if it is correct, verifying the list from the top down as far as possible. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the ACB.
<i>Bad Data =</i>	The ACB's llink field.
<i>Good Data =</i>	The rlink field of the ACB indexed by bad data.
<i>Logical Key =</i>	The ACB's index.
<i>Dump =</i>	The ACB.

Table 12 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will idle the last ACB it was looking at (if any), abort, and be restarted in directed mode. Otherwise, it will just be restarted in directed mode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ACB (if any).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Key</i> =	Index of the ACB (if any).
<i>Dump</i> =	The ACB (if any).

ACCSDB**Software Release:** 5E10 and later**1. AUTOMATED CALLING CARD SERVICE DATA BLOCK (ACCSDB) AUDIT**

The automated calling card service data block audit verifies call data used by the operator actions (OA) subsystem. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode. The audit driver AUdrindex performs existence, key, quarstate and member linkage (point-to and point-break) checks. The existence function checks for extra tuples in the RLACCSDB. The semantic function verifies that the busy/idle and link attributes have compatible information. The recovery function recovers the fields in the rIACCSDB tuple that have errors.

2. DATA STRUCTURES

The data structure checked by the ACCSDB audit is the automated calling card service data block (RLACCSDB). The RLACCSDB relation has a one-to-one linkage with RLPTRDB. Figure 1 illustrates the basic audit structure.

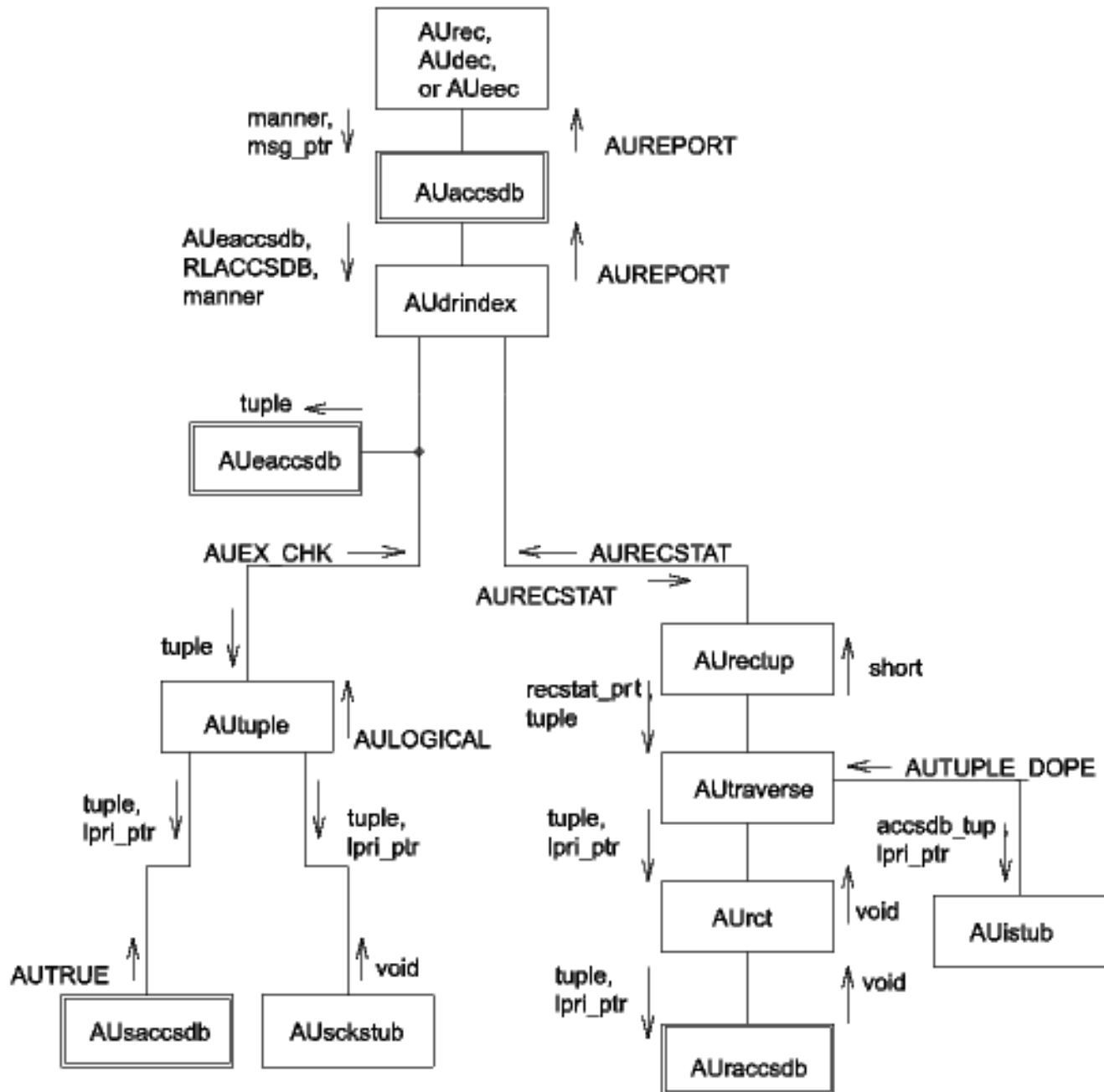


Figure 1 (ACCSD-1) Audit Structure Chart

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The AUaccsdb audit calls the AUdrindex driver. The driver calls the existence function, AUeaccsdb() to check for extra tuples, and then calls AUtuple(), which performs key, quarstate, and member linkage checks. If valid, it goes on to invoke the semantic function.

When an extra tuple error occurs, that tuple is deleted. A semantic error causes the rIACCSDB tuple to be zeroed, the lport and bsy_idl fields restored. The rIACCSDB tuple contents are printed and the related audits LPRTACCS and PTRDB are scheduled to run.

The audit uses these error codes, and also all those used by the audit driver, AUdrindex.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The bsy_idl attribute was read, but returned bad information.
Possible Error Effect:	The rIACCSDB tuple will be lost to the system.
Corrective Action Taken:	The tuple is zeroed, and the key initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIACCSDB.
<i>Bad Data =</i>	bsy_idl attribute in rIACCSDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIACCSDB.
<i>Dump =</i>	rIACCSDB tuple.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIACCSDB is not busy when it should be. The tuple is busy, but the acptl link is zero.
Possible Error Effect:	The rIACCSDB tuple will be lost to the system.
Corrective Action Taken:	The tuple is zeroed, and the key initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIACCSDB.
<i>Bad Data =</i>	DMLINK in rIACCSDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIACCSDB.
<i>Dump =</i>	rIACCSDB tuple.

Table 3 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIACCSDB is not idle when it should be. The tuple is idle, but the acptl link is not zero.
Possible Error Effect:	The rIACCSDB tuple will be lost to the system.
Corrective Action Taken:	The tuple is zeroed, and the key initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIACCSDB.
<i>Bad Data =</i>	DMLINK in rIACCSDB.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	rIACCSDB.
<i>Dump =</i>	rIACCSDB tuple.

ACDINFO

Software Release: 5E10 and later

1. AUTOMATIC CALL DISTRIBUTOR INFORMATION BLOCK (ACDINFO) AUDIT

The Automatic Call Distributor Management Information System (ACD/MIS) is an ISDN feature that is similar to the Operator Services Position System (OSPS) in earlier releases. This feature provides a full set of agent and supervisor service capabilities to commercial applications and internal service provider applications. In the ACD/MIS system, queued calls are presented to logged in agents at ISDN positions. Each agent is part of a serving team that is capable of handling certain type of calls. All agent activities are reported to the Management Information System (MIS), and each agent's call can be monitored by a supervisor. The RLACDINFO relation contains information about a queued call to a particular ISDN agent position. Information representing a queued call is stored in tuples of the RLACDINFO relation until the terminal processes representing the queued calls are initialized. Once used, the rIACDINFO tuples are returned to an idle list for use in the next queued call to be presented to an ACD agent. The purpose of the ACDINFO audit is to ensure the integrity of the tuples in the RLACDINFO relation. The ACDINFO audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in segmented mode. Refer to Figure 1 for a pictorial view of the linkages related to the rIACDINFO tuple.

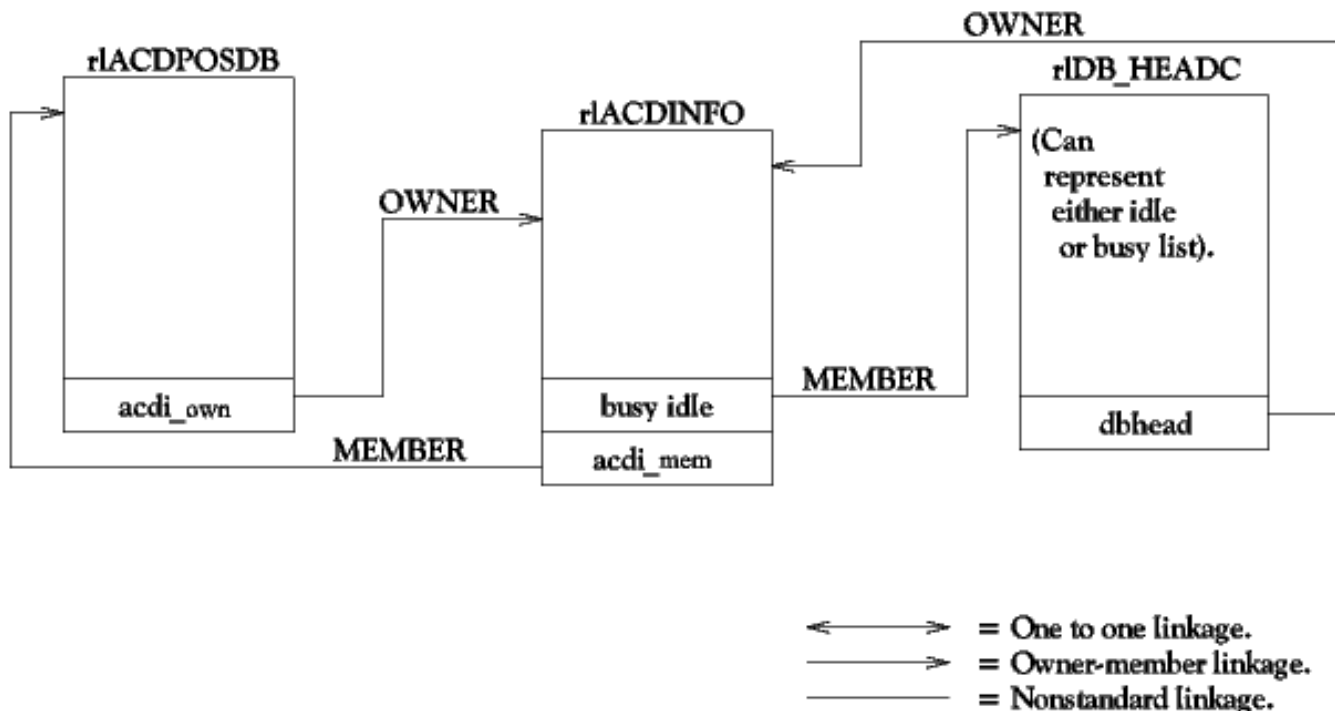


Figure 1 (ACDINFO-1) RLACDINFO Linkages

2. DATA STRUCTURES

The RLACDINFO dynamic relation contains information used to describe a call that has been queued to an ACD ISDN agent position. The agent must be logged into the 5ESS[®] switch, and the information in the tuple is used only until the terminal process representing the call is initialized. RLACDINFO is an indexed compaction access relation that is keyed by the **acdiinfo** field. Each tuple contains the type of ACD call indicated in the **acdtype** field, the miscellaneous call identifier indicated in the **acdmiscid** field, the queue from which the call came from as indicated in the **call_type** field, the logical key of the associated rIBRCSDb tuple as indicated in the **brcsdbkey** field, the linkage to an idle or busy list defined in the RLDB_HEADC relation indicated in the **busyidle** field, a set member linkage field to the owning rIACDPOSDB tuple indicated in the **acdi_mem** field, a stable state field indicated by

tup_state field, and other fields. The ACDINFO audit will verify that idle rIACDINFO tuples are linked to the idle list defined in the RLDB_HEADC relation. If the rIACDINFO tuple indicates it is in use, the ACDINFO audit will verify that it is linked to the busy list defined in the RLDB_HEADC relation. In addition, if the busy tuple is in a stable state, the ACDINFO audit will verify that the **brcsdbkey** points to a valid rIBRCSDB tuple associated with this queued call. The audit will also verify that this rIBRCSDB tuple is associated with an rIDALB tuple.

3. AUDIT CHECKS

The ACDINFO audit uses the DRINASEK audit driver to traverse through each tuple in the RLACDINFO relation. For each tuple, the ACDINFO audit will make the following checks: If the ACDINFO audit is entered:

- BRCSLINK** = Check to make sure that a busy rIACDINFO tuple points to a valid rIBRCSDB tuple associated with the queued call.
- DALBLINK** = Once a valid rIBRCSDB tuple is found for a busy rIACDINFO tuple, verify that it is associated with an rIDALB tuple.
- NOTBUSY** = Check to make sure an rIACDINFO tuple linked to an RLACDPOSDB tuple is linked to the busy list defined in relation RLDB_HEADC.
- NOTIDLE** = Check to make sure that an rIACDINFO tuple that is not linked to an rIACDPOSDB tuple is linked to the idle list defined in relation RLDB_HEADC.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BRCSLINK

Error Code:	BRCSLINK
Error Description:	An rIACDINFO tuple associated with a queued call was found to be linked to the wrong rIBRCSDB tuple. In other words, the acinfokey field in the rIBRCSDB tuple does not point back to this rIACDINFO tuple.
Possible Error Effect:	This may indicate an rIACDINFO tuple that was not released after the queued call was processed. This may lead to the possible loss of the rIACDINFO tuple resource.
Corrective Action Taken:	The rIACDINFO tuple in question is recovered by the audit and linked to the idle list defined in the RLDB_HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDINFO tuple in error.
<i>Bad Data =</i>	The value of the acinfokey field found in the rIBRCSDB tuple.
<i>Good Data =</i>	The value of the logical key of the rIACDINFO tuple under audit.
<i>Logical Index =</i>	The same value as in good data.
<i>Dump 1 =</i>	The rIACDINFO tuple in error.
<i>Dump 2 =</i>	The associated rIBRCSDB tuple.

Table 2 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	An rIACDINFO tuple associated with a queued call was found to be linked to the proper rIBRCSDB tuple. However, this rIBRCSDB tuple was not linked to a valid rIDALB tuple. Therefore, this rIBRCSDB tuple is not associated with a call.
Possible Error Effect:	This may indicate an rIACDINFO tuple that was not released after the queued call was processed. This may lead to the possible loss of the rIACDINFO tuple

	resource.
Corrective Action Taken:	The rIACDINFO tuple in question is recovered by the audit and linked to the idle list defined in the RLDB HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDINFO tuple in error.
<i>Bad Data =</i>	The value of the brcdb_ink.link field in the rIBRCSDB tuple which represents the physical linkage to an rIDALB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The value of the logical key of the rIACDINFO tuple under audit.
<i>Dump 1 =</i>	The rIACDINFO tuple under audit.
<i>Dump 2 =</i>	The associated rIBRCSDB tuple.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rIACDINFO tuple was found to be linked to an rIACDPOSDB tuple indicating that it is a busy tuple associated with a queued call. However, the rIACDINFO tuple was not linked to the busy list defined in the RLDB HEADC relation.
Possible Error Effect:	This may indicate an rIACDINFO tuple that was not released after the queued call was processed. This may lead to the possible loss of the rIACDINFO tuple resource.
Corrective Action Taken:	The rIACDINFO tuple in question is recovered by the audit and linked to the idle list defined in the RLDB HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDINFO tuple in error.
<i>Bad Data =</i>	The physical key of the owning rIDB HEADC tuple.
<i>Good Data =</i>	The physical key of the busy list rIDB HEADC tuple.
<i>Logical Index =</i>	The value of the logical key of the rIACDINFO tuple under audit.
<i>Dump =</i>	The rIACDINFO tuple under audit.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	An rIACDINFO tuple was found that was not linked to an rIACDPOSDB tuple indicating that it should be idle. However, the rIACDINFO tuple was not linked to the idle list defined in the RLDB HEADC relation.
Possible Error Effect:	This may indicate an rIACDINFO tuple that was not released after the queued call was processed. This may lead to the possible loss of the rIACDINFO tuple resource.
Corrective Action Taken:	The rIACDINFO tuple in question is recovered by the audit and linked to the idle list defined in the RLDB HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDINFO tuple in error.
<i>Bad Data =</i>	The physical key of the owning rIDB HEADC tuple.
<i>Good Data =</i>	The physical key of the idle list rIDB HEADC tuple.
<i>Logical Index =</i>	The value of the logical key of the rIACDINFO tuple under audit.
<i>Dump =</i>	The rIACDINFO tuple under audit.

ACDPOSDB

Software Release: 5E10 and later

1. AUTOMATIC CALL DISTRIBUTOR POSITION BLOCK (ACDPOSDB) AUDIT

The Automatic Call Distributor/Management Information System (ACD/MIS) is an integrated services digital network (ISDN) feature. This feature provides a full set of agent and supervisor service capabilities for service provider applications. In the ACD/MIS system, queued calls are presented to logged in agents at ISDN positions. Each agent is part of a serving team that is capable of handling certain types of calls. All agent activities are reported to the MIS and an agent's call can be monitored by a supervisor.

The RLACDPOSDB relation contains information about an ACD agent associated with an ISDN position. The tuples in the RLACDPOSDB relation can be thought of as an extension of its associated rITEDB tuple. There is a one to one correspondence between the rACDPOSDB and rITEDB tuples.

The purpose of the ACDPOSDB audit is to ensure the integrity of the tuples in the RLACDPOSDB relation. The ACDPOSDB audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

Refer to Figure 1 for a pictorial view of the linkages related to the rACDPOSDB tuple.

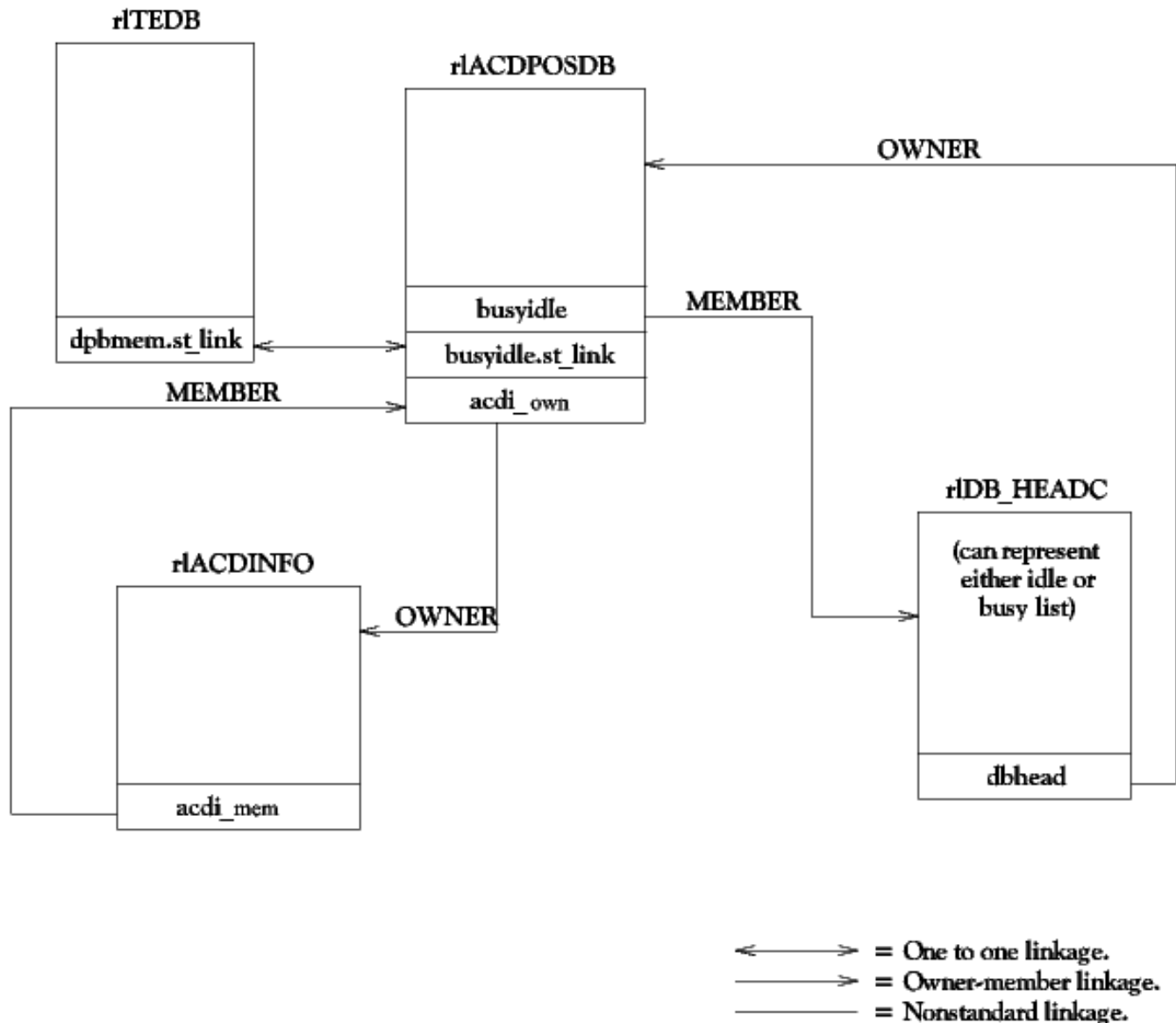


Figure 1 (ACDPOSDB-1) RLACDPOSDB Linkages

2. DATA STRUCTURES

The RLACDPOSDB dynamic relation contains information used to describe an ACD/MIS agent that has logged into the switch at an ISDN position represented by an associated rITEDB tuple. The RLACDPOSDB is an indexed compaction access relation that is keyed by the **acdposdb** field. Each rACDPOSDB tuple contains information about the agent such as the position number indicated by the **posnum** field, the state of the agent represented by the **pos_state** field, the agent ID represented by the **agent_id** field, the internal operator ID represented by the **intopid** field, the associated serving team of the agent represented by the **serv_tm** field, as well as various other fields.

For rACDPOSDB tuples linked to a corresponding rITEDB tuple through their **busyidle.st_link** field, these tuples are also linked to a busy list defined in the RLDB_HEADC relation through their **busyidle.st_ownk** field.

For idle rACDPOSDB tuples not linked to an rITEDB, these tuples are linked to an idle list defined the

RLDB_HEADC relation through their **busyidle.st_ownk** field.

Busy rIACDINFO tuples associated with queued calls for this agent are linked off the **acdi_own** field in the corresponding rIACDPOSDB tuple.

3. AUDIT CHECKS

The ACDPOSDB audit uses the DRINASEK audit driver to traverse through each tuple in the RLACDPOSDB relation. For each tuple, the ACDPOSDB audit will make the following checks:

BASEQUAL2	= Make sure the cw_button field in the rIACDPOSDB tuple matches the corresponding value in the associated rIFEATBN tuple.
LNK_CONFIG	= Make sure the config field in the rIACDPOSDB tuple matches the corresponding value in the associated rIIS_TERM tuple.
LOGTONE	= Make sure the serving team field, serv_tm , is 0 for an rIACDPOSDB tuple representing a logged out agent.
LPORT	= Make sure this rIACDPOSDB tuple has a corresponding rIIS_TERM tuple.
NOTBUSY	= Make sure an rIACDPOSDB tuple linked to an rLTEDB tuple is linked to the busy list in RLDB_HEADC relation.
NOTIDLE	= Make sure that an rIACDPOSDB tuple that is not linked to an rLTEDB is linked to the idle list in the RLDB_HEADC relation.
NUCA	= Make sure the hunted call appearance, field acd_hca , in the rIACDPOSDB matches the corresponding value in the associated rIIS_LINE tuple.
ODDLOSTUP	= Make sure the rIACDPOSDB tuple has a corresponding rIFC_LINE tuple.
PLS_COMP	= Make sure the cpe_comp field in the rIACDPOSDB tuple matches the corresponding value in the associated rIFEATBN tuple.
PORT	= Make sure the rIACDPOSDB tuple has a corresponding rLPT_PT tuple.
SPCL_LINE	= Make sure the ltm_idx field in the rIACDPOSDB matches the corresponding value in the associated rIFC_LINE tuple.
TEDBLINK	= Make sure that a busy rIACDPOSDB tuple is linked to a valid rLTEDB tuple.
TSNO_MATCH	= Make sure the posnum field in the rIACDPOSDB tuple matches the corresponding value in the associated rIIS_TERM tuple.
TYPE	= Make sure the rIACDPOSDB tuple has a corresponding rIIS_LINE tuple.

The following checks are made only on an rIACDPOSDB tuple that represents a logged in agent.

BDPID	= Make sure the mon_pid field in the rIACDPOSDB represents a valid process ID.
BRCSFLAG	= Make sure the bridge available flag, brg_avail , in the rIACDPOSDB of a monitored agent is set to DBYES.
DBLOCKIDLE	= Make sure the position blocked flag, pos_blk , in the corresponding rIDPB tuple is set to DBYES for an rIACDPOSDB tuple representing an agent in the B channel block state.

DPB_LINK	= Make sure the linkage between the corresponding rITEDB tuple and rIDPB tuple is good for an agent who is in the B channel block state.
IDL_STATE	= Make sure that an rACDPOSDB representing an agent in an available state has its corresponding entry in the position available list, Rlpalist, set to DBYES.
INTABT	= Make sure the internal operator id field, intopid , matches the corresponding value in the associated rIOLOGINID tuple.
INV_STATE	= Make sure the position state, field pos_state , in the rACDPOSDB is valid.
LOGPID	= Make sure the rACDPOSDB tuple has a corresponding rIOLOGINID tuple.
UNEQUAL	= Make sure the serving team field, serv_tm , matches the corresponding value in the associated rIOLOGINID tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The cw_button field in the rACDPOSDB tuple has been corrupted. The value does not match the value found in the corresponding static rFEATBN tuple.
Possible Error Effect:	The button lamps associated with this agent position may not function properly.
Corrective Action Taken:	The cw_button field is set to the value defined in the rFEATBN tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rACDPOSDB tuple.
<i>Bad Data =</i>	The value of the cw_button field in the rACDPOSDB before correction.
<i>Good Data =</i>	The value of cw_button obtained from the rFEATBN tuple.
<i>Logical Index =</i>	The rACDPOSDB tuple.
<i>Dump 1 =</i>	The rACDPOSDB tuple.
<i>Dump 2 =</i>	The rFEATBN tuple.

Table 2 Error Code: BDPID

Error Code:	BDPID
Error Description:	The process id contained in the mon_pid in the rACDPOSDB representing a logged in agent is not associated with a valid process.
Possible Error Effect:	The Monitor Bridge Terminal Process (MBTP) did not cleanup after itself when the process ended. This may leave resources in an unusable state.
Corrective Action Taken:	The mon_pid field in the rACDPOSDB tuple is set to OSnullpid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rACDPOSDB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rACDPOSDB tuple.
<i>Dump =</i>	The rACDPOSDB tuple.

Table 3 Error Code: BRCSFLAG

Error Code:	BRCSFLAG
Error Description:	The brg_avail flag in the rACDPOSDB has not been set to DBYES within two

	passes of the audit when the mon_pid field in this rIACDPOSDB tuple indicates a process id of the Monitor Bridge Terminal Process (MBTP).
Possible Error Effect:	The bridge should be made available within two passes of the audit. If not, this could mean a hung resource.
Corrective Action Taken:	The brg_avail field in the rIACDPOSDB tuple is set to DBNO and the corresponding mon_pid field is set to OSnullpid. This will cause the PCBLA audit to purge the MBTP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the brg_avail flag in the rIACDPOSDB tuple.
<i>Good Data =</i>	The value of DBYES.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump =</i>	The rIACDPOSDB tuple.

Table 4 Error Code: DBLOCKIDLE

Error Code:	DBLOCKIDLE
Error Description:	The pos_blk field in the corresponding rIDPB tuple was set to DBNO when the agent position represented by the rIACDPOSDB indicates that the agent is in the B channel blocked state.
Possible Error Effect:	Call processing may not process the B channel blocked state correctly for this agent.
Corrective Action Taken:	The pos_blk field in the rIDPB tuple is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The value of the pos_blk field in the rIDPB tuple.
<i>Good Data =</i>	DBYES.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump 1 =</i>	The rIACDPOSDB tuple.
<i>Dump 2 =</i>	The rIDPB tuple.

Table 5 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The linkage between the rITEDB tuple, associated with this rIACDPOSDB tuple, and the corresponding rIDPB tuple has been corrupted.
Possible Error Effect:	The ISDN port associated with this agent may not function properly.
Corrective Action Taken:	If the agent represented by this rIACDPOSDB tuple is currently logged into the switch, the agent will be logged out. Also, the rIACDPOSDB tuple that represents this agent position will be recovered to the idle list defined in the RLDB_HEADC relation. This will also force recovery on the associated rITEDB tuple once the TEDB audit is run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The value of the port field in the rIDPB tuple.
<i>Good Data =</i>	The rIDPB tuple as specified in the rITEDB tuple.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump 1 =</i>	The rIACDPOSDB tuple.
<i>Dump 2 =</i>	The rITEDB tuple.

Table 6 Error Code: IDL_STATE

Error Code:	IDL_STATE
Error Description:	An agent represented by this rIACDPOSDB has been found to be in the available state without its corresponding entry in the position available list, Rlpalist[], being set to DBYES.
Possible Error Effect:	Hunted ACD calls may not be presented to this agent.
Corrective Action Taken:	The corresponding entry in Rlpalist[] is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.

<i>Bad Data</i> =	The value of the corresponding entry in Rlpalist[].
<i>Good Data</i> =	DBYES.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump</i> =	The rIACDPOSDB tuple.

Table 7 Error Code: INTABT

Error Code:	INTABT
Error Description:	The internal operator fields, intopid , did not match between the dynamic rIACDPOSDB tuple and its corresponding static rIOLOGINID tuple for an agent logged into the switch.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The internal operator id field is corrected in the rIACDPOSDB from the value obtained in the rIOLOGINID tuple. The agent represented by this rIACDPOSDB is logged out of the switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of intopid in the rIACDPOSDB tuple before correction.
<i>Good Data</i> =	The value of intopid in the rIACDPOSDB tuple after correction.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump 1</i> =	The rIACDPOSDB tuple.
<i>Dump 2</i> =	The rIOLOGINID tuple.

Table 8 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The pos_state field in the rIACDPOSDB tuple representing a logged in agent was found to be incorrect by Position Manager. In other words, a call to Position Manager with the event AUDIT_SYNC cause a state change for this rIACDPOSDB tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	Position Manager has already corrected the pos_state field in the rIACDPOSDB. Audits is merely reporting the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of pos_state in the rIACDPOSDB tuple before correction.
<i>Good Data</i> =	The value of pos_state in the rIACDPOSDB tuple after correction.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump</i> =	The rIACDPOSDB tuple.

Table 9 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The config field in the rIACDPOSDB has been corrupted. The value in the rIACDPOSDB tuple does not match the value in the corresponding static rIIS_TERM tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The config field in the rIACDPOSDB is updated to the value indicated in the static rIIS_TERM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of config in the rIACDPOSDB tuple before correction.
<i>Good Data</i> =	The value of config in the rIACDPOSDB tuple after correction.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump 1</i> =	The rIACDPOSDB tuple.
<i>Dump 2</i> =	The rIIS_TERM tuple.

Table 10 Error Code: LOGBID

Error Code:	LOGBID
Error Description:	The audit could not read the corresponding static rIOLOGINID tuple for this

	agent position represented by this rIACDPOSDB tuple.
Possible Error Effect:	Call processing for this agent position may be effected.
Corrective Action Taken:	The agent at this position is logged out of the switch. Since this error may be due to the agent_id field of the rIACDPOSDB being corrupt, a manual action error is not reported.
Manual Action Required:	None. However, if the error continues to roll out, it may signify that a rILOGINID tuple for this agent is missing and may have to be manually inserted.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of the serv_tm field in the rIACDPOSDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump</i> =	The rIACDPOSDB tuple.

Table 11 Error Code: LOGTONE

Error Code:	LOGTONE
Error Description:	A position in the logged out state (such as, pos_state in the rIACDPOSDB tuple is FIP_LOGOUT) was found to have serving team field, serv_tm , in the rIACDPOSDB tuple that was not zero.
Possible Error Effect:	The state of this agent position may be corrupted.
Corrective Action Taken:	The agent represented by this rIACDPOSDB is logged out of the switch. The serv_tm field in the rIACDPOSDB is set to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of serv_tm in the rIACDPOSDB tuple before correction.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump</i> =	The rIACDPOSDB tuple.

Table 12 Error Code: LPORT

Error Code:	LPORT
Error Description:	Using the lport field from the corresponding rITEDB as the key, we were not able to read the corresponding rIIS_TERM for this position represented by the rIACDPOSDB tuple.
Possible Error Effect:	This agent position may not function properly.
Corrective Action Taken:	The agent position represented by this rIACDPOSDB is logged out of the switch. The rIACDPOSDB tuple is recovered to the idle list defined in the RLDB_HEADC relation. Once the TEDB audit is run, the corresponding rITEDB will also be recovered.
Manual Action Required:	None if the error is not persistent. If it continues to roll out, it may be due to a missing rIIS_TERM tuple that will need to be inserted.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of the lport field in the corresponding rITEDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump 1</i> =	The rIACDPOSDB tuple.
<i>Dump 2</i> =	The rITEDB tuple.

Table 13 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The audit found that this rIACDPOSDB tuple was properly linked to an rITEDB tuple. However, this rIACDPOSDB tuple was not linked to the busy list defined in the RLDB_HEADC relation. Any rIACDPOSDB tuple that is linked to an rITEDB tuple must also be linked to the busy list.
Possible Error Effect:	This agent position may not function properly. In addition, the ACD system integrity process may not be able to monitor this agent position.
Corrective Action Taken:	The agent position represented by this rIACDPOSDB is logged out of the switch.

	The rIACDPOSDB tuple is recovered to the idle list defined in the RLDB_HEADC relation. Once the TEDB audit is run, the corresponding rITEDB will also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The physical key of the owning rIDB_HEADC tuple.
<i>Good Data =</i>	The physical key of the busy list rIDB_HEADC tuple.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump =</i>	The rIACDPOSDB tuple.

Table 14 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The audit found an rIACDPOSDB tuple that was not linked to the idle list defined in the RLDB_HEADC relation. However, this tuple was also not linked to an rITEDB tuple.
Possible Error Effect:	This may represent the possible loss of rIACDPOSDB tuple resources.
Corrective Action Taken:	The rIACDPOSDB tuple is recovered to the idle list defined in the RLDB_HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The physical key of the owning rIDB_HEADC tuple.
<i>Good Data =</i>	The physical key of the idle list rIDB_HEADC tuple.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump =</i>	The rIACDPOSDB tuple.

Table 15 Error Code: NUCA

Error Code:	NUCA
Error Description:	The hunted call appearance, field acd_hca , for this agent position does not match the value found in the corresponding static rIIS_LINE tuple.
Possible Error Effect:	Hunted call processing may be impacted for this agent position.
Corrective Action Taken:	The call appearance field, acd_hca , is corrected to the value specified in the rIIS_LINE tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the acd_hca field in the rIACDPOSDB tuple.
<i>Good Data =</i>	The value of the frst_ca field in the rIIS_LINE tuple.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump 1 =</i>	The rIACDPOSDB tuple.
<i>Dump 2 =</i>	The rIIS_LINE tuple.

Table 16 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit was not able to read the corresponding static rIFC_LINE tuple for this agent position represented by the rIACDPOSDB tuple.
Possible Error Effect:	Call processing for this agent position may be effected.
Corrective Action Taken:	None.
Manual Action Required:	The missing static rIFC_LINE tuple must be inserted for this agent position represented by the rIACDPOSDB tuple. It should be inserted by Recent Change or ODBE.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the port field in the corresponding rLPT_PT tuple.
<i>Good Data =</i>	The value of the party field in the corresponding rLPT_PT tuple.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump =</i>	The rIACDPOSDB tuple.

Table 17 Error Code: PLS_COMP

Error Code:	PLS_COMP
Error Description:	The cpe_comp field in the rIACDPOSDB has been corrupted. The value does not match that obtained in the corresponding static rIFEATBN tuple.
Possible Error Effect:	The agent's customer premise equipment may not function correctly. There may also be a possible loss of feature button functionality.
Corrective Action Taken:	The value of cpe_comp field in the rIACDPOSDB tuple is set to the value obtained from the rIFEATBN tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the cpe_comp field in the rIACDPOSDB tuple.
<i>Good Data =</i>	The value of the cpe_comp field in the rIFEATBN tuple.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump 1 =</i>	The rIACDPOSDB tuple.
<i>Dump 2 =</i>	The rIFEATBN tuple.

Table 18 Error Code: PORT

Error Code:	PORT
Error Description:	The audit was unable to read the corresponding static rLPT_PT tuple for this rIACDPOSDB tuple.
Possible Error Effect:	Hunted call processing may be impacted for this agent position.
Corrective Action Taken:	None.
Manual Action Required:	The missing rLPT_PT tuple must be inserted for this agent position represented by this rIACDPOSDB tuple. The tuple can be inserted using Recent Change or ODBE.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the lport field in the corresponding rIIS_TERM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump 1 =</i>	The rIACDPOSDB tuple.
<i>Dump 2 =</i>	The rIIS_TERM tuple.

Table 19 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit was unable to read the corresponding static rIIS_LINE tuple for this rIACDPOSDB tuple.
Possible Error Effect:	Hunted call processing may be impacted for this agent position.
Corrective Action Taken:	None.
Manual Action Required:	The missing rIIS_LINE tuple must be inserted for this agent position represented by this rIACDPOSDB tuple. The tuple can be inserted using Recent Change or ODBE.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the ltm_idx field in the rIACDPOSDB tuple.
<i>Good Data =</i>	The value of the global variable GI_pcrd.
<i>Logical Index =</i>	The rIACDPOSDB tuple.
<i>Dump =</i>	The rIACDPOSDB tuple.

Table 20 Error Code: SPCL_LINE

Error Code:	SPCL_LINE
Error Description:	The ltm_idx field in the rIACDPOSDB tuple is corrupt.
Possible Error Effect:	Call processing may not function properly for this agent position represented by this rIACDPOSDB.
Corrective Action Taken:	The ltm_idx field is updated from the rIFC_LINE tuple. The agent represented by this rIACDPOSDB tuple is logged out of the switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACDPOSDB tuple.
<i>Bad Data =</i>	The value of the ltm_idx field in the rIACDPOSDB tuple.
<i>Good Data =</i>	The value of the ltm_idx field in the rIFC_LINE tuple.

<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump 1</i> =	The rIACDPOSDB tuple.
<i>Dump 2</i> =	The rIFC_LINE tuple.

Table 21 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The audit found an rIACDPOSDB tuple that is linked to the busy list defined in the relation RLDB_HAEDC. However, the linkage to its corresponding rITEDB tuple is corrupt.
Possible Error Effect:	Call processing will not function properly for this agent position represented by this rIACDPOSDB.
Corrective Action Taken:	The associated rIACDPOSDB tuple will be recovered to the idle list defined in the RLDB_HEADC relation. Once the TEDB audit is run, the corresponding rITEDB tuple will also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rITEDB tuple obtained (will not be valid).
<i>Bad Data</i> =	The value of the tedb field from the invalid pointer to rITEDB obtained.
<i>Good Data</i> =	The value of the rITEDB tuple we tried to obtain.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump</i> =	The rIACDPOSDB tuple.

Table 22 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The posnum field in the rIACDPOSDB tuple has been corrupted. The value in the rIACDPOSDB tuple does not match the value in the corresponding static rIIS_TERM tuple.
Possible Error Effect:	Call processing will not function properly for this agent position represented by this rIACDPOSDB.
Corrective Action Taken:	The posnum field in the rIACDPOSDB is updated with the value for the rIIS_TERM tuple. The agent represented by the rIACDPOSDB tuple is logged out of the switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of the corrupted posnum field in the rIACDPOSDB tuple.
<i>Good Data</i> =	The value of the posnum field obtained from the rIIS_TERM tuple.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump 1</i> =	The rIACDPOSDB tuple.
<i>Dump 2</i> =	The rIIS_TERM tuple.

Table 23 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The audit has detected a mismatch of the serv_tm field representing the serving team between the rIACDPOSDB tuple and the static rIOLOGINID tuple. The mismatch has occurred for at least two passes of the audit.
Possible Error Effect:	Hunted calls may not be processed correctly at this agent position.
Corrective Action Taken:	The agent represented by the rIACDPOSDB is logged out of the switch. The serv_tm field in the rIACDPOSDB is updated to the value found in the static rIOLOGINID tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIACDPOSDB tuple.
<i>Bad Data</i> =	The value of the corrupted serv_tm field in the rIACDPOSDB tuple.
<i>Good Data</i> =	The value of the serv_tm field obtained from the rIOLOGINID tuple.
<i>Logical Index</i> =	The rIACDPOSDB tuple.
<i>Dump 1</i> =	The rIACDPOSDB tuple.
<i>Dump 2</i> =	The rIOLOGINID tuple.

ACR

Software Release: 5E10 and later

1. AUXILIARY CALL RECORD (ACR) AUDIT

The auxiliary call record (ACR) audit verifies data used by the call record assembler (CRA) subsystem. An auxiliary call record is a fixed-size block of memory associated with a call record. The CRA subsystem uses information in the auxiliary call record only for Operator Services Position System (OSPS) calls. Auxiliary call record resides only in the switching modules (SMs) and is used exclusively by administrative features. An auxiliary call record is assigned when an OSPS operator supplies information for any one of the following fields (bkname, fwdname, or detail). It remains associated with the call record throughout call processing. Call processing data is recorded in the auxiliary call record until the call disconnects. The ACR then returns to an idle state for reassignment. It may remain assigned to a nonexistent call record or contain irrelevant data. These circumstances may result in a loss of resources.

2. DATA STRUCTURES

Auxiliary call records are stored as elements in an array that only the CRA subsystem accesses. There is one array for each SM. Figure 1 illustrates the basic audit structure.

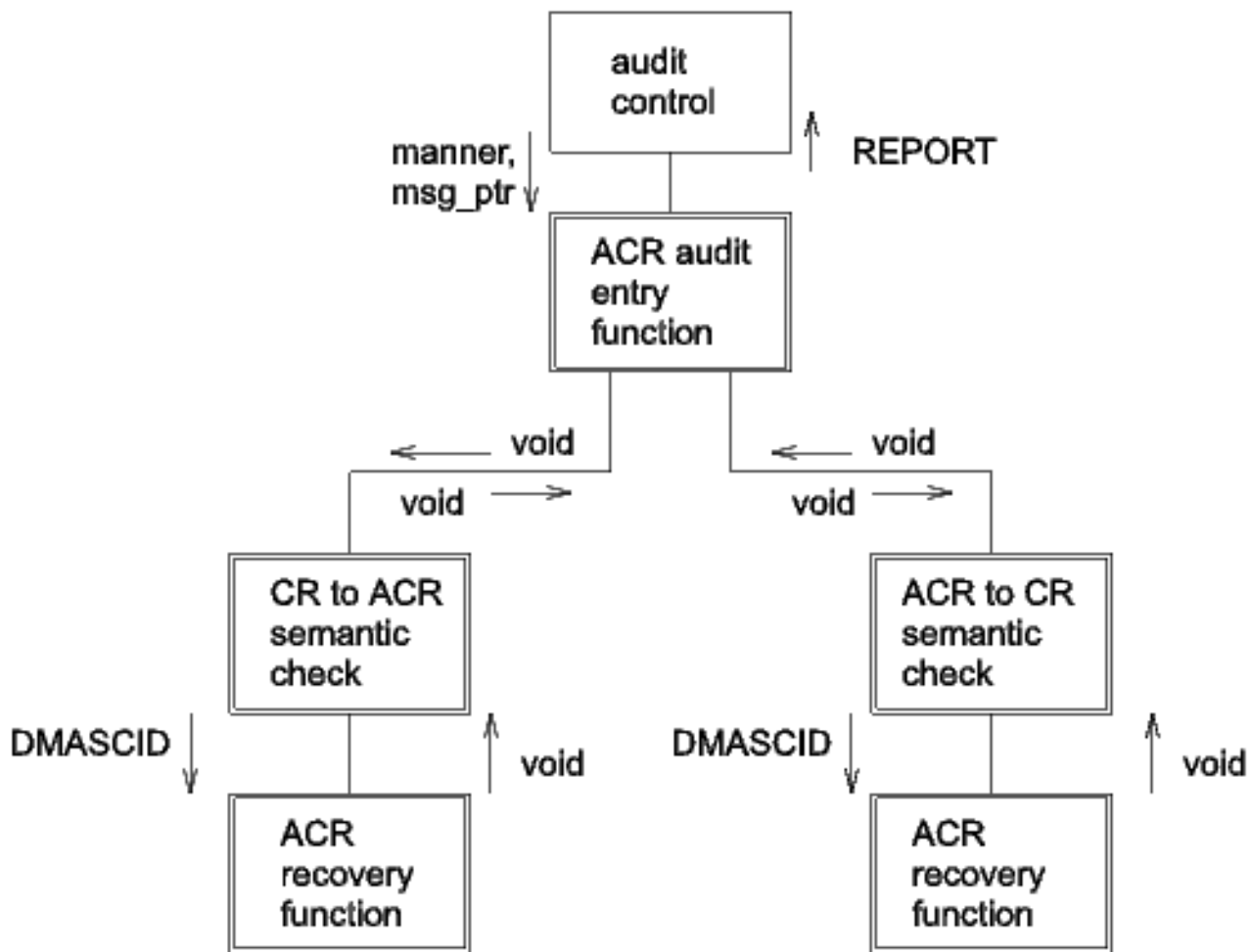


Figure 1 (ACR-1) Audit Structure Chart

3. AUDIT CHECKS

The auxiliary call record audit guards against these errors by identifying and releasing auxiliary call records. The auxiliary call record audit runs in the initialization and segmented modes. The audit performs the same functions in both modes. It does not take segment breaks in the initialization mode.

The audit runs in two phases. The first phase is traversing the call record array and the second phase is traversing the auxiliary call record array.

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The ACR audit range checks the index of the call record (CR), next it traverses the call record structure and reads the index. The audit reads the status and ascid fields of the ACR for active auxiliary call records. An ACR whose status is not busy or ascid that does not contain the correct call identification is reported. The second phase range checks the ascid of the ACR and traverses the auxiliary call record structure reading the status and ascid fields.

An ACR whose status is busy and whose ascid is null is reported. An ACR whose status is idle and has not been placed on the ACR idle queue list is reported. An ACR whose status is busy but has been allocated to a nonexistent call record, the call record's index, or points to a different ACR is reported according to the following guidelines:

- (1) When the value of the index in the CR is out of range and not null, it is set to null.
- (2) When a busy CR has been found to point to a busy ACR which does not "point-back to" that CR, and the ACR has all zero data fields, the ACR is released and the index in the CR is cleared.
- (3) When a busy CR is found to point to a busy ACR which does not "point-back to" that CR, and the ACR has at least one non-zero data field, the CR's cid becomes the ACR's "owner" cid.
- (4) When a busy CR is found to point to an idle ACR, and the ACR has all zero data fields, the ACR is released and the index in the CR is cleared.
- (5) When a busy CR is found to point to an idle ACR, and the ACR has at least one non-zero data field, the ACR is made busy and the CR's cid becomes the ACR's "owner" cid.
- (6) When an idle CR is found to point to a busy ACR, the ACR is released and the indexes in the CR is cleared.
- (7) When an idle CR is found to point to an idle ACR, the index in the CR is cleared.
- (8) When the value of the ascid in the ACR is out of range, and is not null, it is set to null.
- (9) When a busy ACR is found with a null ascid, and it is not pointed to by a CR, the ACR is released.
- (10) When a busy ACR is found to point to an idle CR, the ACR is released.
- (11) When a busy ACR is found to point to a busy CR, which does not "point-back to" that ACR, and the CR index is null, the ACR is connected to the CR.
- (12) When a busy ACR is found to point to a busy CR which does not "point-back to" that ACR, and the CR index is not null, the ACR is released.
- (13) When an idle ACR is found to point to an idle CR, the ACR is released.
- (14) When the address of an ACR which is stored in the ACR address table is out-of-range, it is reset and the data for that ACR is cleared.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACPTRBAD

Error Code:	ACPTRBAD
Error Description:	The linkage to the "master" call record array CRecord[] in the ACRRecord[] auxiliary call record array (ACRRecord[].ascid) was found in a non-null condition (not equal to CRNULLINDX) but is set to an out of range value. This linkage should point to the array index of a valid "master" call record (where ACRRecord[].ascid = the array index X in CRecord[X]) and should be in the range of allowable call records. The allowable range is greater than or equal to CRZERO, but less than CRMAX.
Possible Error Effect:	This auxiliary call record array entry is stuck in a busy state and is lost to the system. This results in a loss of resources to OSPS operations.
Corrective Action Taken:	The ascid linkage is cleared (ascid.crindx marked CRNULLINDX and ascid.crpcrid cleared to 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the auxiliary call record array entry.
<i>Bad Data =</i>	The invalid index portion of the linkage to the "master" call record which was found (ACRRecord[].ascid.crindx).
<i>Good Data =</i>	The value ascid.crindx was corrected to CRNULLINDX.
<i>Logical Index =</i>	The array index of the ACRRecord[] array entry.
<i>Dump =</i>	NA.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	Bad link to an ACR. A CR was found with auxcrindx pointing to an ACR whose ascid did not contain the correct call ID.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD may be returned to the idle list, or attached to the correct call record.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CRECORD.
<i>Bad Data =</i>	ascid of ACRECORD.
<i>Good Data =</i>	auxcrindx of CRECORD.
<i>Logical Index =</i>	Index to CRECORD.
<i>Dump =</i>	NA.

Table 3 Error Code: BDPID

Error Code:	BDPID
Error Description:	Bad process associated with an ACR. An ACR was found with a status of busy, but it is allocated to a busy call record whose auxcrindx did not contain the correct call ID.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD may be returned to the idle list, or attached to the correct call record.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of ACRECORD.
<i>Bad Data =</i>	ascid of ACRECORD.
<i>Good Data =</i>	Auxcrindx of CRECORD.
<i>Logical Index =</i>	Index to ACRECORD.
<i>Dump =</i>	NA.

Table 4 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	Bad process associated with an ACR. An ACR was found with a status of busy, but is allocated to an idle call record.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of ACRECORD.
<i>Bad Data =</i>	ascid of ACRECORD.
<i>Good Data =</i>	CRNULLINDX.
<i>Logical Index =</i>	Index to ACRECORD.
<i>Dump =</i>	NA.

Table 5 Error Code: INDEX

Error Code:	INDEX
Error Description:	Bad index in a CR. An idle CR was found with an auxcrindx pointing to a ACR.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD may be returned to the idle list, or the auxcrindx of the call record may be set to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CRECORD.
<i>Bad Data =</i>	auxcrindx of CRECORD.
<i>Good Data =</i>	CRNULLINDX.
<i>Logical Index =</i>	Index to CRECORD.
<i>Dump =</i>	NA.

Table 6 Error Code: INV

Error Code:	INV
Error Description:	Invalid link from an ACR to a CR. An ACR was found with a status of idle, and is allocated to an idle CR.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The corresponding ACRECORD is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of ACRECORD.
<i>Bad Data =</i>	ascid of ACRECORD.
<i>Good Data =</i>	CRNULLINDX.
<i>Logical Index =</i>	Index to ACRECORD.
<i>Dump =</i>	NA.

Table 7 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	Nonexistent process associated with a busy ACR. An ACR was found with a status of busy, but its ascid is null.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of ACRECORD.
<i>Bad Data =</i>	Status of ACRECORD.
<i>Good Data =</i>	0 for ACRIDLE.
<i>Logical Index =</i>	Index to ACRECORD.
<i>Dump =</i>	NA.

Table 8 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	Bad link to an ACR. A CR was found with an auxcrindx pointing to an auxiliary call record whose STATUS was idle.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD ascid and CRECORD auxcrindx may be set to null, or the ACR is attached to the call record and made busy.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CRECORD.
<i>Bad Data</i> =	auxcrindx of CRECORD.
<i>Good Data</i> =	CRNULLINDX.
<i>Logical Index</i> =	Index to CRECORD.
<i>Dump</i> =	NA.

Table 9 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The value of the ascid in the ACR is out of range. The index was either less than zero or greater than the maximum number of CRs allowed and was not null.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The ACRECORD ascid is set to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of ACRECORD.
<i>Bad Data</i> =	ascid of ACRECORD.
<i>Good Data</i> =	CRNULLINDX.
<i>Logical Index</i> =	Index to ACRECORD.
<i>Dump</i> =	NA.

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The address of the ACR entry in ACRaddrtbl is out of range.
Possible Error Effect:	Access to ACRarray for this ACR entry is broken and may cause write protection errors.
Corrective Action Taken:	Correct the ACRaddrtbl entry with the address of the ACRarray entry. Also reset the status, ascid.crindx, and ascid.crpcrid for this ACR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad entry in ACRaddrtbl.
<i>Bad Data</i> =	Contents of bad entry in ACRaddrtbl.
<i>Good Data</i> =	Address of entry in ACRarray.
<i>Logical Index</i> =	Index to CRECORD.
<i>Dump</i> =	NA.

Table 11 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	The value of the auxcrindx in the CR is out-of-range. The index was either less than zero or greater than the maximum number of ACRs allowed and was not null.
Possible Error Effect:	The ACRECORD will be lost to the system.
Corrective Action Taken:	The CRECORD auxcrindx is set to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CRECORD.
<i>Bad Data</i> =	auxcrindx of CRECORD.
<i>Good Data</i> =	CRNULLINDX.
<i>Logical Index</i> =	Index to CRECORD.
<i>Dump</i> =	NA.

Table 12 Error Code: ZAP

Error Code:	ZAP
Error Description:	Addressing error occurred while the ASacr audit was running.
Possible Error Effect:	Audit was unable to finish.
Corrective Action Taken:	Zap the cursor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CRECORD entry.
<i>Bad Data</i> =	Index to CRECORD.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

ACRECORD

Software Release: 5E10 and later

1. AUXILIARY CALL RECORD (ACRECORD) AUDIT

Auxiliary call records are used in conjunction with the call record (RLCRECORD) and operator services position system (OSPS) call record (OCRECORD) administration and billing relations to maintain call related information for the life of an OSPS call. Auxiliary call records contain detailed information entered by an operator. This information is stored in the RLACRECORD relation and is audited by the ACRECORD audit. The ACRECORD audit runs in the post-initialization and segmented modes in all switching modules (SMs).

2. DATA STRUCTURES

The ACRECORD audit checks rIACRECORD tuples to determine if the tuples are validly marked busy or idle. The audit makes these checks based on the following data:

- acrlink.st_ownk** = Set owner linkage to the RLDB_HEADC head cell idle list. For busy rIACRECORD tuples, this is marked DBACRIDLE.
- acrlink.st_link** = One-to-one linkage with an associated rIOCRECORD tuple. For all busy rIACRECORD tuple, this linkage must be used.

The rIACRECORD tuple and related data structures referenced by the ACRECORD audit are shown in Figure 1 .

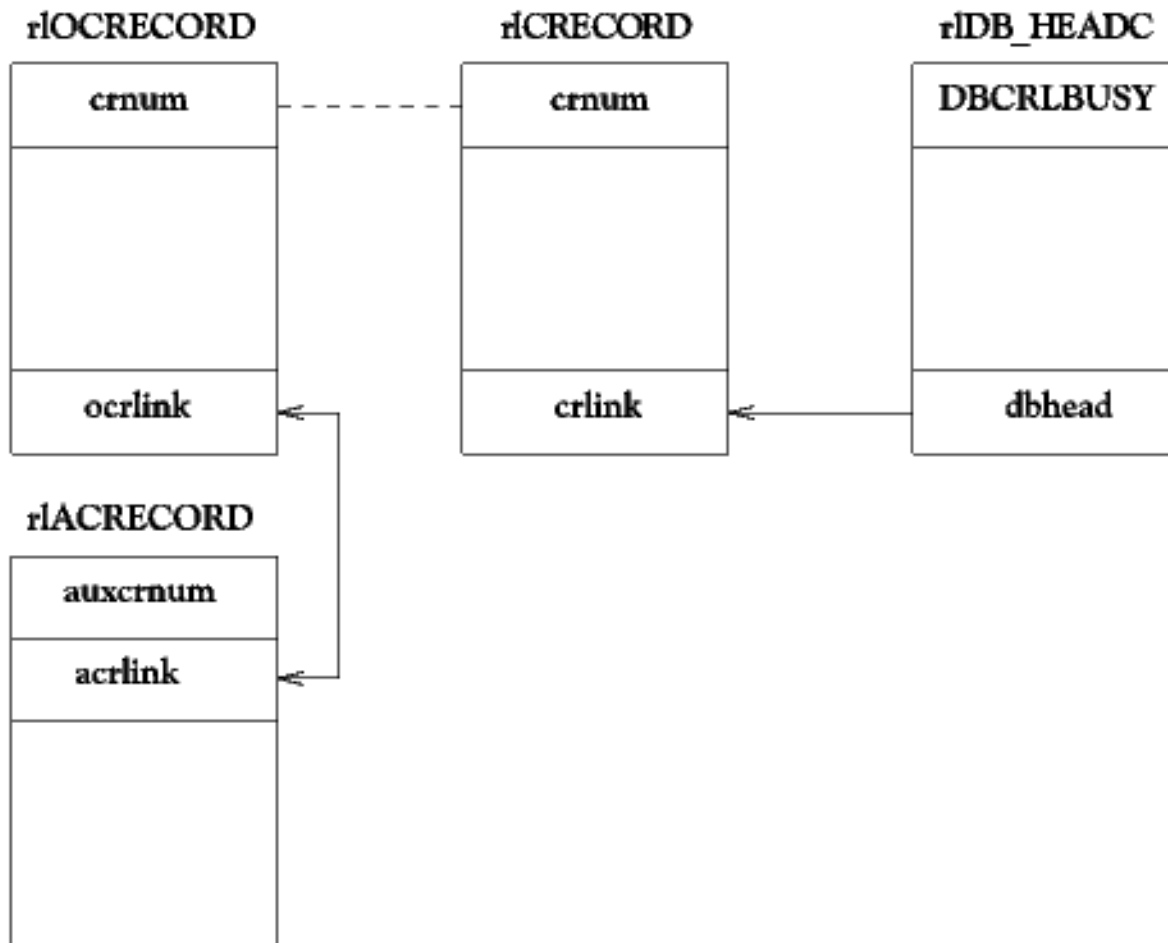


Figure 1 (ACRECORD-1) Data Structures of the ACRECORD Audit

3. AUDIT CHECKS

For each tuple in the RLACRECORD relation, the ACRECORD audit makes the following checks: If the ACRECORD audit gets entered:

- CORRUPT_IDLE** = Check that the one-to-one linkage to RLOCRECORD is not used if the rIACRECORD tuple is linked to the RLDB_HEADC idle list head cell (DBACRECIDLE).
- INVOWNER** = Check that rIACRECORD tuples which are busy (not linked to the RLDB_HEADC idle list head cell) have a corresponding rIOCRECORD OSPA call record tuple linked.
- NOTBUSY** = Check that rIACRECORD tuples which have a rIOCRECORD OSPA call record tuple linked are not linked to the RLDB_HEADC idle list head cell.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	A rIACRECORD tuple was found linked to the RLDB_HEADC idle list head cell, but there was a corresponding rIOCRECORD tuple linked to it. All idle rIACRECORD tuples must not be linked to associated rIOCRECORD tuples (where rIACRECORD.acrlink.st link == DALNKNUL).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rIACRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIACRECORD tuple.
<i>Bad Data =</i>	The contents of the one-to-one linkage to RLOCRECORD (rIACRECORD.acrlink.st link - DBSFOCRECO).
<i>Good Data =</i>	Contents of the linkage to the RLDB_HEADC idle list head cell (rIACRECORD.acrlink.st ownk - DBSFDB HEA).
<i>Logical Index =</i>	Logical key of the rIACRECORD tuple (rIACRECORD.auxcrnum).
<i>Dump =</i>	Contents of the rIACRECORD tuple.

Table 2 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	A busy rIACRECORD tuple is not linked to a rIOCRECORD tuple. All rIACRECORD tuple not linked to the RLDB_HEADC idle list head cell must be linked to rIOCRECORD tuples (where rIACRECORD.acrlink.st ownk == DALNKNUL AND rIACRECORD.acrlink.st link == rIOCRECORD.crnum - DBSFOCRECO).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rIACRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIACRECORD tuple.
<i>Bad Data =</i>	The contents of the one-to-one linkage to RLOCRECORD (rIACRECORD.acrlink.st link).
<i>Good Data =</i>	Logical key of rIOCRECORD tuple obtained from rIACRECORD linkage (rIOCRECORD.crnum).
<i>Logical Index =</i>	Logical key of the rIACRECORD tuple (rIACRECORD.auxcrnum).
<i>Dump =</i>	Contents of the rIACRECORD tuple.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A busy rIACRECORD tuple is linked to a rIOCRECORD tuple, however, its owner linkage is not busy. All busy rIACRECORD tuples must have their owner linkages marked to zero where (rIACRECORD.acrlink.st ownk == DALNKNUL).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rIACRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIACRECORD tuple.
<i>Bad Data =</i>	The contents of the one-to-one linkage to RLOCRECORD (rIACRECORD.acrlink.st link).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIACRECORD tuple (rIACRECORD.auxcrnum).
<i>Dump =</i>	Contents of the rIACRECORD tuple.

ACSRQC

Software Release: 5E10 and later

1. AUTOMATIC CUSTOMER STATION REARRANGEMENT QUEUE COMPLIMENT (ACSRQC) AUDIT

The ACSRQC tuple is used when a new request for ACSR is received by the system, and is created after the static tuple of type rIRCACSRQ is created. The ACSRQC audit verifies the extra tuples in the dynamic relation RLACSR_QC, and also verifies the semantic data in that tuple. The audit runs segmented in operational kernel process (OKP) environment.

2. DATA STRUCTURES

rIACSR_QC.

3. AUDIT CHECKS

The ACSRQC audit steps through the tuples of RLACSR_QC relation, verifying extra tuples in RLACSR_QC relation, and also verifies the semantic data in that tuple. For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

If any extra tuples are found in RLACSR_QC relation, those tuples are deleted. If any field is found to be inconsistent with a static tuple, it is corrected. The related audit, RLACSRQ, is run to recover other data structures.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit did not find a static tuple of type rIRCACSR_Q when it tried to access that tuple from dynamic rIACSR_QC tuple. This means that this dynamic tuple is an extra tuple.
Possible Error Effect:	The result of the status query of an ACSR request may be erroneously replied as if the tuple exists even though it does not.
Corrective Action Taken:	The audit will report the error and will delete the extra dynamic tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the dynamic tuple buffer.
<i>Bad Data =</i>	This field here is used to dump the value of the request number of the extra dynamic tuple.
<i>Good Data =</i>	0. There is no good data.
<i>Logical Index =</i>	The value of the "epdn" field of the extra dynamic tuple.
<i>Dump =</i>	The extra dynamic tuple of type rIACSR_QC.

Table 2 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	When audit found an error in the status field of the rIACSR_QC tuple it tried to get the actual address of the tuple using DB primitives but it failed.
Possible Error Effect:	The result of the status query of an ACSR request may be erroneously replied.
Corrective Action Taken:	The audit will report the error and will return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the dynamic tuple buffer.
<i>Bad Data =</i>	0.

<i>Good Data</i> =	0.
<i>Logical Index</i> =	The value of the "epdn" field of the dynamic tuple.
<i>Dump</i> =	The dynamic tuple of type rIACSR_QC whose status field does not match.

Table 3 Error Code: NUMELEM

Error Code:	NUMELEM
Error Description:	The key field epdn of the current dynamic tuple rIACSR_QC does not match the same field in static tuple of type rIRCACSR_Q. This also means that the req_number field of the dynamic tuple could be wrong.
Possible Error Effect:	The dynamic tuple has wrong information or it is an extra tuple. This can lead to false information given when inquired about the acsr request.
Corrective Action Taken:	The audit will report the error and will delete the dynamic tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the dynamic tuple buffer.
<i>Bad Data</i> =	This field here is used to dump the value of the request number of the static tuple. It may or may not be a bad data. If the req_number of static rIRCACSR_Q tuple and the rIACSR_QC does match then the epdn fields of both will not and they are dumped in the following two fields.
<i>Good Data</i> =	The "epdn" field of the static rIRCACSR_Q tuple.
<i>Logical Index</i> =	The "epdn" field of the dynamic tuple being audited.
<i>Dump</i> =	The dynamic tuple of type rIACSR_QC being audited.

Table 4 Error Code: STATUS

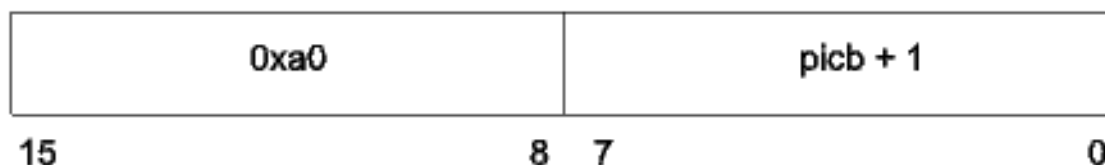
Error Code:	STATUS
Error Description:	The status field of the dynamic rIACSR_QC tuple does not match the same field of the static tuple of type rIRCACSR_Q.
Possible Error Effect:	The result of the status query of an ACSR request will be erroneously replied.
Corrective Action Taken:	The audit will report the error and will correct the status field of the dynamic tuple from the static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the dynamic tuple buffer.
<i>Bad Data</i> =	The status field of the dynamic tuple being audited.
<i>Good Data</i> =	Status field of the static tuple rIRCACSR_Q tuple.
<i>Logical Index</i> =	The value of the "epdn" field of the dynamic tuple.
<i>Dump</i> =	The extra dynamic tuple of type rIACSR_QC.

ACTBASE

Software Release: 5E10 and later

1. ACTIVE SERVICE GROUP BASE ADDRESS (ACTBASE) AUDIT

The active service group base addresses array (SMactbase) contains the base addresses of the active controllers for each peripheral line unit. The switch may have up to eight peripheral line units per SM. The base address for an active controller is derived from the peripheral interface control bus (PICB) number as follows:



The ACTBASE audit ensures that the SMactbase array contains the base address of the active line unit controller. The ACTBASE audit is part of the switch maintenance audits subgroup. It is part of the BASE software package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The ACTBASE audit checks the active service group base address array (SMactbase) against information in the switch maintenance equipment status relation (RLSMEST). It obtains the picb + 1 value from the PClusname array. Figure 1 shows these data structures.



Figure 1 (ACTBASE-1) Data Structures of the ACTBASE AUDIT

3. AUDIT CHECKS

The audit steps through SMactbase array on a per line unit basis and performs the following checks.

- Validates the base address in SMactbase after obtaining the picb + 1 value from PClusname.
- Constructs the key of rLSMEST (ckname) is formed as follows:



- Checks rSMEST to verify that the base address points to the active controller. It does this by checking the bas_stat and qual_2 fields in rSMEST. If the bas_stat is active (SMACT) and qual_2 is null (SMNULL), the entry points to the base address of an active controller. The audit considers all other entries errors.

4. ERROR HANDLING

The audit invokes the peripheral fault recovery (PFR) program to correct inconsistencies in SMactbase. It schedules the circuit data audit (CKTDATA) as a related audit.

The audit produces the following error code(s).

Table 1 Error Code: ACTBASE

Error Code:	ACTBASE
Error Description:	The SMactbase[] array contains an invalid address.
Possible Error Effect:	The SMIM function SMgdcntler will be called to correct the error.
Corrective Action Taken:	Correct the SMactbase[] array to contain the base address of active controller.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMactbase[].
<i>Bad Data =</i>	Contents of SMactbase[].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	swno.
<i>Dump =</i>	NA.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	In case of a line unit controller duplex failure the SMactbase[] array does not contain the base address of the side 0 or side 1 controller service group. SMEST for both controller indicate out-of-service status. (rSMEST.bas_stat = SMOOS).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Base address of service group 0 is placed in the SMactbase[] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Base address of service group 1.
<i>Bad Data =</i>	Base address of current service group in SMactbase[].
<i>Good Data =</i>	Base address of service group 0.
<i>Logical Index =</i>	Line unit number.
<i>Dump 1 =</i>	rSMEST for service group 0 common board.
<i>Dump 2 =</i>	rSMEST for service group 1 common board.

Table 3 Error Code: SMEST

Error Code:	SMEST
Error Description:	The key to retrieve CC rSMEST tuple has error.
Possible Error Effect:	None.
Corrective Action Taken:	The related audits will be scheduled to correct the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Key to retrieve CC rSMEST tuple.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: STATE

Error Code:	STATE
Error Description:	The CC base address in SMactbase[] is not pointing to the active CC.
Possible Error Effect:	None.
Corrective Action Taken:	The SMactbase[] will be corrected by SMIM function SMgdcntler.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMactbase[].
<i>Bad Data</i> =	Contents of SMactbase[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Service group number.
<i>Dump</i> =	Both CC rISMEST tuples.

Table 5 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUSri_otime, used for transient timing may have occurred.
Possible Error Effect:	The audits transient timing has been interrupted. Lost resources may take longer to recover because transient timing must be restarted.
Corrective Action Taken:	AUSri_otime will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the transient timing variable (AUSri_otime).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Content of the transient timing variable (AUSri_otime).
<i>Dump 2</i> =	Content of the current OSDS ITIME (OSnow).

Table 6 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The SMactbase[] contains a non-null value for an unequipped line unit.
Possible Error Effect:	None.
Corrective Action Taken:	The SMactbase[] will be set to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMactbase[].
<i>Bad Data</i> =	Contents of SMactbase[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	swno.
<i>Dump</i> =	NA.

Table 7 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to provide recovery after taking a reset. When a reset occurs (usually triggered by an illegal or out of range memory access), the audit which was running at the time is aborted, and is then rescheduled to run in AUZAP mode to provide recovery of related data left inconsistent as a result of the reset.
Possible Error Effect:	The audit may continue to cause memory access violations, increasing the probability of drastic recovery action and preventing the audit from checking for further errors.
Corrective Action Taken:	The data which caused the reset is initialized by the next entry of the audit.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The number of the line unit under audit at the time of the reset.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

ADB

Software Release: 5E10 and later

1. ANALOG DATA BLOCK (ADB) AUDIT

The ADB audit insures the internal consistency of the RLADB relation and performs linkage checks between the RLADB relation and the RLDB_HEADC and RLPORTLA relations.

2. DATA STRUCTURES

The RLADB relation is a dynamic, indexed sequential relation that is used to indicate the presence of a call on that call appearance for an analog port in a key system group. An rIADB tuple is used when an ISDN port in a key system group is the originator of a call. An rIADB is also used when the analog port in the key system group is bridging onto the call. When the rIADB tuple is not being used, it is linked onto an idle list that is contained in the RLDB_HEADC relation.

3. AUDIT CHECKS

The ADB audit is invoked through audit control which calls the ADB audit entry function. This function invokes the indexed sequential driver (DRINASEK) to step through all tuples in the RLADB relation in a sequential manner. If all the linkage and semantic checks pass, the driver will step to the next tuple and continue another cycle. If any checks fail, recovery is started.

The following is a list of semantic checks that are performed:

- (a) If the rIADB tuple is on the idle link list, then the linkage to the rIPORTLA tuple should be null.

The following checks are only made if the rIADB tuple is on the busy list:

- (a) The rIADB tuple should be linked to a rIPORTLA tuple associated with a shared analog port.
- (b) The shared call model process identification field should be associated with a live process.
- (c) If the rIADB tuple is involved in a bridge connection, then the analog terminal process identification field should be associated with a live process.

4. ERROR HANDLING

When an error is detected either in the semantic function or in the linkage checking routine, a failure return is returned back to the indexed sequential driver. If the rIADB tuple is linked onto the idle or busy list, the rIADB is unlinked from that list. If the rIADB tuple is mutilated in such a way that it can not be unlinked from the list, the indexed sequential driver patches the list. The rIADB tuple that was found in error is recovered and any rIADB tuples that were left hanging as a result of a list being patched.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	A non idle tuple is not linked to the rIPORTLA tuple.
Possible Error Effect:	Incorrect linkage.
Corrective Action Taken:	The rIADB tuple will be put on an idle list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIADB tuple.
<i>Bad Data</i> =	Linkage field from rIADB to rIORTLA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIADB tuple.
<i>Dump</i> =	Contents of the rIADB tuple.

Table 2 Error Code: BDPID

Error Code:	BDPID
Error Description:	The associated rIORTLA owner link to rIPCBLA does not match the <i>atp_pic.procno</i> .
Possible Error Effect:	The rIADB tuple will not have the same process number as the rIPCBLA.
Corrective Action Taken:	The rIADB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIADB tuple.
<i>Bad Data</i> =	OSPID of the TP of the analog terminal.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIADB tuple.
<i>Dump</i> =	Contents of the rIADB tuple.

Table 3 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The OSPID of the TP which contains the SCM in this module is not a live process.
Possible Error Effect:	Call will not be linked correctly.
Corrective Action Taken:	The rIADB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIADB tuple.
<i>Bad Data</i> =	The process number in the rIADB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIADB tuple.
<i>Dump</i> =	The contents of the rIADB tuple.

Table 4 Error Code: FPID_DEAD

Error Code:	FPID_DEAD
Error Description:	The <i>atp_pid</i> is a non-null pid, however, the process is a dead one.
Possible Error Effect:	A dead process is connected to a non-null pid.
Corrective Action Taken:	The rIADB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of three rIADB tuple.
<i>Bad Data</i> =	The process number for the <i>atp_pid</i> .
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rIADB tuple.
<i>Dump</i> =	The contents of the rIADB tuple.

Table 5 Error Code: PID

Error Code:	PID
Error Description:	The OSPID of the TP which contains the SCM is null.
Possible Error Effect:	The pid is idle when it should be busy.
Corrective Action Taken:	The rIADB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIADB tuple.
<i>Bad Data</i> =	The process number of the pid for rIADB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIADB tuple.
<i>Dump</i> =	Contents of the rIADB tuple.

Table 6 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	This error is taken because the rIADB tuple is not on either a idle list or a busy list.
Possible Error Effect:	The tuple is not linked to anything.
Corrective Action Taken:	The rIADB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIADB tuple.
<i>Bad Data =</i>	Current owner link of rIADB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIADB tuple.
<i>Dump =</i>	The contents of the rIADB tuple.

Table 7 Error Code: TPID

Error Code:	TPID
Error Description:	The pid(scm_pid) in the rIADB tuple does not match the pid(tp1 or tp2) in the rISCMDB tuple or rILSCMDB tuple.
Possible Error Effect:	There will be a loss of resources.
Corrective Action Taken:	The rIADB tuple will be put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIADB tuple.
<i>Bad Data =</i>	OSPID of TP.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIADB tuple.
<i>Dump 1 =</i>	The contents of the rIADB tuple.
<i>Dump 2 =</i>	The contents of the rISCMDB or rILSCMDB tuple.

Table 8 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rIADB tuple has a nonzero linkage in one of its link blocks.
Possible Error Effect:	Resource is linked to a rIADB tuple on the idle list and could be unusable by call processing.
Corrective Action Taken:	The audit will place the rIADB tuple on the idle list and zero the linkages.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIADB tuple.
<i>Bad Data =</i>	Nonzero value of linkage.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The rIADB tuple.
<i>Dump =</i>	Contents of the rIADB tuple.

ADCB-A

Software Release: 5E12 only

1. AUTOPLEX® DATA CONTROL BLOCK (ADCB) AUDIT

The AUTOPLEX® data control block (ADCB) audit was developed for the CDMA data services feature. The ADCB array is used to perform the adaption layer processing in the PHV. It includes pointers to the radio link protocol control block (RLPCB) and the adaption layer (ADL) application to interworking queues (PSAIQ).

2. DATA STRUCTURES

For details on the fields contained in the RLADCB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	The ADCB array includes a pointer to the RLPCB which in turn points to several queues. The indexes in these queues are verified by the audit to be within a valid range.
Possible Error Effect:	The indexes to the queues are not within a valid range indicating they are corrupted and need to be restored.
Corrective Action Taken:	The error is printed and a recovery function is called to re-initialize the RLP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RLPCB.
<i>Bad Data =</i>	Contents of the RLPCB's <i>vr</i> field; next to receive.
<i>Good Data =</i>	0 (N/A)
<i>Logical Key =</i>	The ADCB's index
<i>Dump =</i>	The AURLPCBDUMP audit structure.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The ADP application to interworking queue (PSAIQ) includes a header/payload buffer descriptor (BD). The audit will check the BD pointers to verify they are within a valid range and that the next BD pointer is PSNULLBDPTR.
Possible Error Effect:	The pointers are invalid indicating they are corrupted and need to be restored.
Corrective Action Taken:	The error is printed and the pointers are re-initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Contents of the ADCB AIQ header/payload <i>actcount</i> field.
<i>Bad Data =</i>	Contents of the ADCB AIQ header/payload <i>nxtbdptr</i> field.
<i>Good Data =</i>	PSNULLBDPTR; value that <i>nxtbdptr</i> should be set to.
<i>Logical Key =</i>	The ADCB's index.
<i>Dump =</i>	None.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The processor took a non-returning error and the audit control decided that the ADCB audit was running at the time. The ADCB audit will report this as an error.
Possible Error Effect:	The error will be reported.
Corrective Action Taken:	No recovery action will take place.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0
<i>Bad Data =</i>	0
<i>Good Data =</i>	0
<i>Logical Key =</i>	0
<i>Dump =</i>	None.

ADCB-B

Software Release: 5E13 and later

1. AUTOPLEX® DATA CONTROL BLOCK (ADCB) AUDIT

The AUTOPLEX® data control block (ADCB) audit was developed for the CDMA data services feature. The ADCB array is used to perform the adaption layer processing in the PHV. It includes pointers to the radio link protocol control block (RLPCB) and the adaption layer (ADL) application to interworking queues (PSAIQ).

2. DATA STRUCTURES

For details on the fields contained in the RLADCB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	The ADCB array includes a pointer to the RLPCB which in turn points to several queues. The indexes in these queues are verified by the audit to be within a valid range.
Possible Error Effect:	The indexes to the queues are not within a valid range indicating they are corrupted and need to be restored.
Corrective Action Taken:	The error is printed and a recovery function is called to re-initialize the RLP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RLPCB.
<i>Bad Data =</i>	RLPCB→retxq head.
<i>Good Data =</i>	0 (N/A)
<i>Logical Key =</i>	The ADCB's index.
<i>Dump =</i>	The RLPCB.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The ADP application to interworking queue (PSAIQ) includes a header/payload buffer descriptor (BD). The audit will check the BD pointers to verify they are within a valid range and that the next BD pointer is PSNULLBDPTR.
Possible Error Effect:	The pointers are invalid indicating they are corrupted and need to be restored.
Corrective Action Taken:	The error is printed and the pointers are re-initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Contents of the ADCB AIQ header/payload <i>actcount</i> field.
<i>Bad Data =</i>	Contents of the ADCB AIQ header/payload <i>nextbdptr</i> field.
<i>Good Data =</i>	PSNULLBDPTR; value that <i>nextbdptr</i> should be set to.
<i>Logical Key =</i>	The ADCB's index.
<i>Dump =</i>	None.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The processor took a non-returning error and the audit control decided that the ADCB audit was running at the time. The ADCB audit will report this as an error.
Possible Error Effect:	The error will be reported.
Corrective Action Taken:	No recovery action will take place.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0
<i>Bad Data =</i>	0
<i>Good Data =</i>	0
<i>Logical Key =</i>	0
<i>Dump =</i>	None.

AGENTID

Software Release: 5E11 and later

1. AGENT IDENTIFICATION (AGENTID) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. The data which defines each agent is stored in the RAGENTID (agent identification) relation. Data in this relation is audited by the AGENTID audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RAGENTID relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a static rAGENTID tuple that does not have a corresponding dynamic rAGENTPOS tuple.
Possible Error Effect:	This may prevent calls from being presented to the agent represented by this rAGENTID tuple.
Corrective Action Taken:	The audit will insert the missing dynamic rAGENTPOS tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rAGENTPOS tuple.
<i>Bad Data</i> =	The relation ID of the RAGENTPOS relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rAGENTPOS tuple.
<i>Dump</i> =	NA.

AGENTPOS

Software Release: 5E11 and later

1. AGENT POSITION (AGENTPOS) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. Agent serving team position data is stored in the RAGENTPOS (agent position) relation. Data in this relation is audited by the AGENTPOS audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RAGENTPOS relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTESTPOS

Error Code:	BADTESTPOS
Error Description:	The audit was unable to obtain a dynamic rISRVQDB (serving team queue data block) tuple corresponding to the rAGENTPOS tuple which represents an agent position logged into the switch. The agent position number should be consistent between the rAGENTPOS tuple and the rISRVQDB tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The rAGENTPOS tuple is idled to represent an undefined position.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rAGENTPOS tuple.
<i>Bad Data =</i>	The invalid position number (rISRVQDB.posnum).
<i>Good Data =</i>	The tuple's position number (rAGENTPOS.posnum).
<i>Logical Index =</i>	The logical key of the rAGENTPOS tuple.
<i>Dump =</i>	The rAGENTPOS tuple.

Table 2 Error Code: LOGBID

Error Code:	LOGBID
Error Description:	The audit found an inconsistency in the values of the agent identification field, agent_id, between the dynamic rAGENTPOS tuple and its corresponding dynamic rISRVQDB tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The rAGENTPOS tuple is idled to represent an undefined position.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rAGENTPOS tuple.
<i>Bad Data =</i>	The invalid agent identification number (rISRVQDB.agent_id).
<i>Good Data =</i>	The tuple's agent identification number (rAGENTPOS.posnum).

<i>Logical Index</i> =	The logical key of the rAGENTPOS tuple.
<i>Dump 1</i> =	The rAGENTPOS tuple.
<i>Dump 2</i> =	The rSRVQDB tuple.

AIGSDB-A

Software Release: 5E11 - 5E13

1. ACCESS INTERFACE UNIT (AIU) GROUP SCAN DATA BLOCK (AIGSDB) AUDIT

A group scan data block is linked to an active concentrated buffer headcell when a line origination is buffered. Each group scan data block represents a group of eight lines. Each line in a group is represented by one bit. If the bit is set, there is an origination on the line waiting to be processed. The AIU group scan data block audit (AIGSDB) ensures that group scan data block tuples (rIAI_GSDBs) are properly linked to one of the concentrated buffer headcell tuples (rICBHC). The rICBHC tuples represent the idle list or one of several active lists. The audit also checks the name of the first port in the scan group that is stored in an rIAI_GSDB tuple.

2. DATA STRUCTURES

For details on the fields contained in the RLAI_GSDB relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rIAI_GSDB tuple is queued for origination but the unit type attribute (rIAI_GSDB.utype) is not consistent with the unit type attribute (rUNITDATA.utype) from the static rUNITDATA tuple.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the unit type (rIAI_GSDB.utype) attribute of the rIAI_GSDB tuple to be consistent with the unit type (rUNITDATA.utype) attribute of the static rUNITDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIAI_GSDB tuple.
<i>Bad Data =</i>	Unit type value (rIAI_GSDB.utype) from the rIAI_GSDB tuple.
<i>Good Data =</i>	Unit type value (rUNITDATA.utype) from the rUNITDATA tuple.
<i>Logical Index =</i>	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump =</i>	rIAI_GSDB tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIAI_GSDB tuple was found. Either the read of rICKTDATA was successful and the equipage status was SMUNEQ, or the read failed.
Possible Error Effect:	Incorrect data is left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIAI_GSDB tuple.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 3 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Either a dummy rIAI_GSDB tuple contains nonzero data, or the logical key of the rIAI_GSDB tuple is corrupt.
Possible Error Effect:	Invalid attempts to use the rIAI_GSDB tuple.
Corrective Action Taken:	The audit zeros the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Good Data</i> =	Calculate logical key value of the rIAI_GSDB tuple.
<i>Logical Index</i> =	Physical key value of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 4 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rIAI_GSDB tuple is not linked to the idle list and the link to the rIGSLK tuple is null leaving the rIAI_GSDB tuple in limbo.
Possible Error Effect:	The rIAI_GSDB tuple is lost to the system and may reduce call capacity.
Corrective Action Taken:	The audit reinitializes the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Link to the rIGSLK tuple (rIAI_GSDB.gscbm.st_link).
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 5 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The rIAI_GSDB tuple is linked to an invalid list.
Possible Error Effect:	The rIAI_GSDB tuple is lost to the system and may reduce call capacity.
Corrective Action Taken:	The audit reinitializes the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Link to the rICBHC tuple (rIAI_GSDB.gscbm.st_ownk).
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 6 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The rIAI_GSDB tuple contains bad data in the port name (rIAI_GSDB.port[]) attribute that is inconsistent with data from the static rUNITDATA and rICKT_PORT tuples.
Possible Error Effect:	May inhibit origination of a line port.
Corrective Action Taken:	The audit reinitializes the port name (rIAI_GSDB.port[]) attribute in the rIAI_GSDB tuple to be consistent with data from the static rUNITDATA and rICKT_PORT tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Port name (rIAI_GSDB.port[]) from the rIAI_GSDB tuple.
<i>Good Data</i> =	Calculated port name.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump 1</i> =	rUNITDATA tuple.
<i>Dump 2</i> =	rICKT_PORT tuple.

<i>Dump 2=</i>	rIAI_GSDB tuple.

Table 7 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIAI_GSDB tuple is on the idle list, but is still linked to the rIGSLK tuple.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIAI_GSDB tuple.
<i>Bad Data =</i>	Non null data found in the rIAI_GSDB tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump =</i>	rIAI_GSDB tuple.

AIGSDB-B

Software Release: 5E14 and later

1. ACCESS INTERFACE UNIT (AIU) GROUP SCAN DATA BLOCK (AIGSDB) AUDIT

A group scan data block is linked to an active concentrated buffer headcell when a line origination is buffered. Each group scan data block represents a group of eight lines. Each line in a group is represented by one bit. If the bit is set, there is an origination on the line waiting to be processed. The AIU group scan data block audit (AIGSDB) ensures that group scan data block tuples (rIAI_GSDBs) are properly linked to one of the concentrated buffer headcell tuples (rICBHC). The rICBHC tuples represent the idle list or one of several active lists. The audit also checks the name of the first port in the scan group that is stored in an rIAI_GSDB tuple.

2. DATA STRUCTURES

For details on the fields contained in the RLAI_GSDB relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rIAI_GSDB tuple is queued for origination but the unit type attribute (rIAI_GSDB.utype) is not consistent with the unit type attribute (rUNITDATA.utype) from the static rUNITDATA tuple.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the unit type (rIAI_GSDB.utype) attribute of the rIAI_GSDB tuple to be consistent with the unit type (rUNITDATA.utype) attribute of the static rUNITDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIAI_GSDB tuple.
<i>Bad Data =</i>	Unit type value (rIAI_GSDB.utype) from the rIAI_GSDB tuple.
<i>Good Data =</i>	Unit type value (rUNITDATA.utype) from the rUNITDATA tuple.
<i>Logical Index =</i>	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump =</i>	rIAI_GSDB tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIAI_GSDB tuple was found. Either the read of rICKTDATA was successful and the equipage status was SMUNEQ, or the read failed.
Possible Error Effect:	Incorrect data is left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIAI_GSDB tuple.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 3 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Either a dummy rIAI_GSDB tuple contains nonzero data, or the logical key of the rIAI_GSDB tuple is corrupt.
Possible Error Effect:	Invalid attempts to use the rIAI_GSDB tuple.
Corrective Action Taken:	The audit zeros the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Good Data</i> =	Calculate logical key value of the rIAI_GSDB tuple.
<i>Logical Index</i> =	Physical key value of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 4 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rIAI_GSDB tuple is not linked to the idle list and the link to the rIGSLK tuple is null leaving the rIAI_GSDB tuple in limbo.
Possible Error Effect:	The rIAI_GSDB tuple is lost to the system and may reduce call capacity.
Corrective Action Taken:	The audit reinitializes the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Link to the rIGSLK tuple (rIAI_GSDB.gscbm.st_link).
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 5 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The rIAI_GSDB tuple is linked to an invalid list.
Possible Error Effect:	The rIAI_GSDB tuple is lost to the system and may reduce call capacity.
Corrective Action Taken:	The audit reinitializes the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Link to the rICBHC tuple (rIAI_GSDB.gscbm.st_ownk).
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump</i> =	rIAI_GSDB tuple.

Table 6 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The rIAI_GSDB tuple contains bad data in the port name (rIAI_GSDB.port[]) attribute that is inconsistent with data from the static rUNITDATA and rICKT_PORT tuples.
Possible Error Effect:	May inhibit origination of a line port.
Corrective Action Taken:	The audit reinitializes the port name (rIAI_GSDB.port[]) attribute in the rIAI_GSDB tuple to be consistent with data from the static rUNITDATA and rICKT_PORT tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIAI_GSDB tuple.
<i>Bad Data</i> =	Port name (rIAI_GSDB.port[]) from the rIAI_GSDB tuple.
<i>Good Data</i> =	Calculated port name.
<i>Logical Index</i> =	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump 1</i> =	rUNITDATA tuple.
<i>Dump 2</i> =	rICKT_PORT tuple.

<i>Dump 2=</i>	rIAI_GSDB tuple.

Table 7 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIAI_GSDB tuple is on the idle list, but is still linked to the rIGSLK tuple.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rIAI_GSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIAI_GSDB tuple.
<i>Bad Data =</i>	Non null data found in the rIAI_GSDB tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key (rIAI_GSDB.grpname) of the rIAI_GSDB tuple.
<i>Dump =</i>	rIAI_GSDB tuple.

AIUAP-A

Software Release: 5E11 - 5E13

1. ACCESS INTERFACE UNIT APPLICATION PACK (AIUAP) TIME SLOT MAP AUDIT

The access interface unit (AIU) application pack (AP) time slot map (PCaiuap[][]) is used by peripheral control to provide maintenance and call processing status for time slots on the AP CODEC and the pulse code modulation (PCM) highway in an AIU.

2. DATA STRUCTURES

For details on the fields contained in the structures mentioned in the error descriptions below, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The dynamic copy of the DPIDB mask (PCaiuap[][].drtsmask) in the PCaiuap[][] array does not match the static copy (rIAIU2AP.dpidbmask[]) in the corresponding rIAIU2AP tuple.
Possible Error Effect:	D-channel path setup on this DPIDB may be blocked.
Corrective Action Taken:	The DPIDB mask (PCaiuap[][].drtsmask) in the PCaiuap[][] array is made consistent with the DPIDB mask (rIAIU2AP.dpidbmask[]) in the static rIAIU2AP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiuap[][] array entry.
<i>Bad Data</i> =	DPIDB mask (PCaiuap[][].drtsmask) from the PCaiuap[][] array.
<i>Good Data</i> =	DPIDB mask (rIAIU2AP.dpidbmask[]) from the rIAIU2AP tuple.
<i>Logical Index</i> =	Includes the following: • AIU number in bits 8-31. • AP time slot number in bits 0-7.
<i>Dump</i> =	NA.

Table 2 Error Code: BIT2MAP

Error Code:	BIT2MAP
Error Description:	The coder/decoder (CODEC) time slot map (codecmmap[]) is inconsistent with data in the PCM highway map (hwymap[]) and the relative peripheral interface data bus (PIDB) array (compidb[]) in the PCaiuap[][] array.
Possible Error Effect:	Call processing may try to hunt to an invalid AP time slot.
Corrective Action Taken:	Recover the CODEC time slot map (codecmmap[]) to be consistent with data in the PCM highway map (hwymap[=0]) and the relative PIDB array (compidb[]) in the PCaiuap[][] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Contents of the CODEC time slot map (codecmmap[]) from the PCaiuap[][] array.

<i>Bad Data</i> =	Includes the following: • Port associated with the even AP time slot in bits 16-31. • Port associated with the odd AP time slot in bits 0-15.
<i>Good Data</i> =	Includes the following: • Idle status of the even AP time slot in bits 24-31. • Even AP time slot shift indicator in bits 16-23. • Idle status of the odd AP time slot in bits 8-15. • Odd AP time slot shift indicator in bits 0-7.
<i>Logical Index</i> =	Includes the following: • AIU number in bits 15-31. • AP number in bits 8-15. • AP time slot number in bits 0-7.
<i>Dump 1</i> =	PCaiup[][] array entry found to be in error.
<i>Dump 2</i> =	rPTSB tuples associated with the even AP time slot.
<i>Dump 3</i> =	rPTSB tuples associated with the odd AP time slot.

Table 3 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	A time slot on both bus A (PCaiup[][].compidb[0]) and bus B (PCaiup[][].compidb[1]) are busy which is illegal for the AP type being audited.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Includes the following: • The AIU number in bits 16 through 31. • The AIU AP number in bits 0 through 15.
<i>Bad Data</i> =	Contents of the AP highway to PIDB map (PCaiup[][].compidb[]).
<i>Good Data</i> =	AP type.
<i>Logical Index</i> =	Includes the following: • The AP bus number in bits 16 through 31. • The AP time slot number in bits 0 through 15.
<i>Dump</i> =	PCaiup[][] array entry.

Table 4 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The AP time slot is being used by diagnostics, but the PCM highway map field (hwymap[]) in the PCaiup[][] array is not marked busy, the partial map field [] in PCaiup[][] is not marked completely full, or the rSMEST tuple does

	indicate that there is maintenance activity (rSMEST.stat_act) on the associated line card.
Possible Error Effect:	Maintenance activities may erroneously try to use this AP time slot.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCaiup[][] array entry found to be in error.
<i>Bad Data =</i>	Maintenance activity (rSMEST.stat_act) data.
<i>Good Data =</i>	Relative PIDB number.
<i>Logical Index =</i>	Time slot number.
<i>Dump =</i>	PCaiup[][] array entry found to be in error.

Table 5 Error Code: CDCONFIG

Error Code:	CDCONFIG
Error Description:	The AP time slot is connected to side 1, according to the setting of the compidb[] field in the PCaiup[][] array, but this is a simplex configuration, so there should be no connections to side 1.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rUNITDATA tuple containing the AP circuit information.
<i>Bad Data =</i>	The unit number that is under test.
<i>Good Data =</i>	Contents of the compidb[] field.
<i>Logical Index =</i>	Time slot number.
<i>Dump 1 =</i>	PCaiup[][] array entry found to be in error.
<i>Dump 2 =</i>	The rUNITDATA tuple.

Table 6 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	An attempt was made to read the rEQUIPDSL tuple for either a quarter or non-ODB full time slot based on the logical PH stored in the rIPTSB tuple, but the rEQUIPDSL tuple was found not to exist.
Possible Error Effect:	The data in the rIPTSB tuple is either corrupted or invalid, and it cannot be determined what equipage is associated with the tuple.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Includes the following: • The AIU number in bits 16 through 31. • The AIU AP number in bits 0 through 15.
<i>Bad Data =</i>	Logical PH number.
<i>Good Data =</i>	Logical PH member number.
<i>Logical Index =</i>	Includes the following: • The AP bus number in bits 16 through 31. • The AP time slot number in bits 0 through 15.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	PCaiup[][] array entry.

Table 7 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance state (maint) in PCaiup[][] is not consistent with the maintenance state (rSMEST.bas_stat) found in the corresponding rSMEST tuple for the AP circuit.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The maintenance state in PCaiup[][] is corrected to the maintenance state found in the corresponding rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiup[][] .
<i>Bad Data =</i>	Contents of maintenance state.
<i>Good Data =</i>	Correct maintenance state.
<i>Logical Index =</i>	Circuit name from the rUNITDATA tuple.
<i>Dump =</i>	rSMEST tuple.

Table 8 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The relative PIDB field (compidb[]) in the PCaiupdb[][] array for this time slot is marked busy, but the state field (rIPTSB.use) in the corresponding rIPTSB tuple indicates that this time slot is idle or invalid.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiup[][] .
<i>Bad Data =</i>	State field (rIPTSB.use) from the rIPTSB tuple.
<i>Good Data =</i>	Time slot number.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple.
<i>Dump 1=</i>	rIPTSB tuple.
<i>Dump 2=</i>	PCaiupdb[][] array entry found to be in error.

Table 9 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	An ODD inconsistency has been found between the RLUNITDATA (that is, RLCKTDATA and RLUNIT_CKT) and the RLAIUPIDB relations.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rIAIUPIDB tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rIAIUPIDB tuple should be inserted into the database with a key (rIAIUPIDB.picb) or PICB number that matches the value of the <i>Bad Data</i> in the Dump Description .
Dump Description:	
<i>Error Address =</i>	Includes the following: • The AIU number in bits 16 through 31. • The AIU AP number in bits 0 through 15.
<i>Bad Data =</i>	PICB number.
<i>Good Data =</i>	Contents of the AP highway to PIDB map (PCaiup[][].compidb[]).
<i>Logical Index =</i>	Includes the following: • The AP bus number in bits 16 through 31. • The AP time slot number in bits 0 through 15.

<i>Dump</i> =	rUNITDATA tuple.

Table 10 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The maintenance state (maint) in PCaiup[][] is not out-of-service (OOS) when rUNITDATA tuple is not found.
Possible Error Effect:	Call processing may try to hunt to an unequipped AP.
Corrective Action Taken:	The maintenance state in PCaiup[][] is corrected to OOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiup[][].
<i>Bad Data</i> =	Contents of maintenance state.
<i>Good Data</i> =	SMOOS.
<i>Logical Index</i> =	Circuit index from the rUNITDATA tuple.
<i>Dump</i> =	PCaiupdb[][] array entry found to be in error.

Table 11 Error Code: PTSB

Error Code:	PTSB
Error Description:	The relative PIDB field (compidb[]) is corrupt, causing the access of the corresponding rIPTSB tuple to fail.
Possible Error Effect:	The AP time slot is lost to the system, causing call processing to degrade.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple.
<i>Bad Data</i> =	Relative PIDB number.
<i>Good Data</i> =	Base address of the AP circuit.
<i>Logical Index</i> =	Logical key (rIPTSB.ptsb) of the rIPTSB tuple.
<i>Dump 1</i> =	rIPTSB tuple.
<i>Dump 2</i> =	PCaiup[][].

Table 12 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The relative PIDB field (compidb[]) in the PCaiup[][] array for this time slot is marked busy, but the port information data in the rIPTSB tuple does not point back to this AP time slot.
Possible Error Effect:	The AP time slot is lost to the system causing call processing to degrade.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Includes the following: - If <i>Bad Data</i> is 0, return value from the PCpt2ptinfo() primitive. - If <i>Bad Data</i> is 1, return value from reading the rCKTDATA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	AP number.
<i>Logical Index</i> =	Logical key (rIPTSB.ptsb) of the rIPTSB tuple.
<i>Dump 1</i> =	rIPTSB tuple.
<i>Dump 2</i> =	PCaiup[][].
<i>Dump 3</i> =	Port data information of type PCPXCXINFO.

Table 13 Error Code: RELPIDB

Error Code:	RELPIDB
Error Description:	The AP time slot is being used for maintenance activities, but the relative PIDB

	field (compidb[]) does not contain a valid line card circuit index.
Possible Error Effect:	Maintenance activities may erroneously try to use this AP time slot.
Corrective Action Taken:	The relative PIDB field (compidb[]) and the PCM highway map field (hwymap[]) are set to idle, and the partial map field (partmap[]) is set to completely full in the PCaiupap[][] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCaiupap[][] array entry.
<i>Bad Data =</i>	Relative PIDB data (compidb[]) from PCaiupap[][].
<i>Good Data =</i>	Circuit index from the rUNIT_CKT tuple.
<i>Logical Index =</i>	Time slot number.
<i>Dump =</i>	PCaiupap[][] array entry found to be in error.

Table 14 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	A rUNITDATA static tuple exists in the database that contains an AP circuit name (rUNITDATA.ckname), but the corresponding static rAIU2AP tuple for this AIU (rAIU2AP.aiuno) is missing from the database.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rAIU2AP tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rAIU2AP tuple should be inserted into the database with a key (rAIU2AP.aiuno) or AIU unit number that matches the value of <i>Logical Key</i> in the Dump Description .
Dump Description:	
<i>Error Address =</i>	Address of PCaiupap[][] array entry.
<i>Bad Data =</i>	RLAIU2AP relation identification.
<i>Good Data =</i>	RLUNITDATA relation identification.
<i>Logical Index =</i>	AIU unit number.
<i>Dump =</i>	rUNITDATA tuple.

Table 15 Error Code: SMEST

Error Code:	SMEST
Error Description:	Access of the rSMEST tuple for either the AP circuit associated with the PCaiupap[][] array entry has failed.
Possible Error Effect:	The AIUAP time slot map will skip auditing the PCaiupap[][] data associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	If <i>Good Data</i> is 0, then this is the address of the PCaiupap[][] array entry. If <i>Good Data</i> is 1, then this is the AIU unit number shifted up 16 bits or'ed with the AP number.
<i>Bad Data =</i>	Address of the rSMEST tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rSMEST key.
<i>Dump =</i>	rUNITDATA tuple if <i>Good Data</i> is 0 or rUNIT_CKT tuple if <i>Good Data</i> is 1.

Table 16 Error Code: TSBUSYIDLE

Error Code:	TSBUSYIDLE
Error Description:	The relative PIDB field (compidb[]) in the PCaiupap[][] array for this AP time slot is marked busy, but the PCM highway map field (hwymap[]) in the PCaiupap[][] array is not marked busy, or the partial map field [][] in PCaiupap[][] is not marked completely full.
Possible Error Effect:	Call processing or maintenance activities accessing invalid data may assert.
Corrective Action Taken:	The PCM highway map field (hwymap[]) in the PCaiupap[][] array for this AP time slot is marked busy to match the state of the corresponding rPTSB tuple and the partial map field (partmap[]) is marked completely full.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCaiupap[][].

<i>Bad Data</i> =	AP number.
<i>Good Data</i> =	Relative PIDB number.
<i>Logical Index</i> =	Time slot number.
<i>Dump 1</i> =	rIPTSB tuple.
<i>Dump 2</i> =	rIPTSB tuple if <i>Bad Data</i> is 1, 2, or 3. PCaiup[][] array entry if <i>Bad Data</i> is 4.

Table 17 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	The relative PIDB field (compidb[]) in the PCaiup[][] array for this time slot is marked idle, but the corresponding bit in PCM highway field (hwymap) is not marked idle, or the bit in the partial map field (partmap) is not marked as completely full.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle and the bit in the partial map field (partmap) is set to completely full.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCaiup[][].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Relative PIDB data (compidb[]).
<i>Logical Index</i> =	Index into the relative PIDB array (compidb[]).
<i>Dump</i> =	PCaiup[][] array entry found to be in error.

AIUAP-B

Software Release: 5E14 and later

1. ACCESS INTERFACE UNIT APPLICATION PACK (AIUAP) TIME SLOT MAP AUDIT

The access interface unit (AIU) application pack (AP) time slot map (PCaiup[][]) is used by peripheral control to provide maintenance and call processing status for time slots on the AP CODEC and the pulse code modulation (PCM) highway in an AIU.

2. DATA STRUCTURES

For details on the fields contained in the structures mentioned in the error descriptions below, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The dynamic copy of the DPIDB mask (PCaiup[][].drtsmask) in the PCaiup[][] array does not match the static copy (rAIU2AP.dpidbmask[]) in the corresponding rAIU2AP tuple.
Possible Error Effect:	D-channel path setup on this DPIDB may be blocked.
Corrective Action Taken:	The DPIDB mask (PCaiup[][].drtsmask) in the PCaiup[][] array is made consistent with the DPIDB mask (rAIU2AP.dpidbmask[]) in the static rAIU2AP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiup[][] array entry.
<i>Bad Data</i> =	DPIDB mask (PCaiup[][].drtsmask) from the PCaiup[][] array.
<i>Good Data</i> =	DPIDB mask (rAIU2AP.dpidbmask[]) from the rAIU2AP tuple.
<i>Logical Index</i> =	Includes the following: • AIU number in bits 8-31. • AP time slot number in bits 0-7.
<i>Dump</i> =	NA.

Table 2 Error Code: BIT2MAP

Error Code:	BIT2MAP
Error Description:	The coder/decoder (CODEC) time slot map (codecmmap[]) is inconsistent with data in the PCM highway map (hwymap[]) and the relative peripheral interface data bus (PIDB) array (compidb[]) in the PCaiup[][] array.
Possible Error Effect:	Call processing may try to hunt to an invalid AP time slot.
Corrective Action Taken:	Recover the CODEC time slot map (codecmmap[]) to be consistent with data in the PCM highway map (hwymap[0]) and the relative PIDB array (compidb[]) in the PCaiup[][] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Contents of the CODEC time slot map (codecmmap[]) from the PCaiup[][] array.

<i>Bad Data</i> =	Includes the following: - Port associated with the even AP time slot in bits 16-31. - Port associated with the odd AP time slot in bits 0-15.
<i>Good Data</i> =	Includes the following: - Idle status of the even AP time slot in bits 24-31. - Even AP time slot shift indicator in bits 16-23. - Idle status of the odd AP time slot in bits 8-15. - Odd AP time slot shift indicator in bits 0-7.
<i>Logical Index</i> =	Includes the following: - AIU number in bits 15-31. - AP number in bits 8-15. - AP time slot number in bits 0-7.
<i>Dump 1</i> =	PCaiup[][] array entry found to be in error.
<i>Dump 2</i> =	rPTSB tuples associated with the even AP time slot.
<i>Dump 3</i> =	rPTSB tuples associated with the odd AP time slot.

Table 3 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	A time slot on both bus A (PCaiup[][].compidb[0]) and bus B (PCaiup[][].compidb[1]) are busy which is illegal for the AP type being audited.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Includes the following: - The AIU number in bits 16 through 31. - The AIU AP number in bits 0 through 15.
<i>Bad Data</i> =	Contents of the AP highway to PIDB map (PCaiup[][].compidb[]).
<i>Good Data</i> =	AP Type.
<i>Logical Index</i> =	Includes the following: - The AP bus number in bits 16 through 31. - The AP time slot number in bits 0 through 15.
<i>Dump</i> =	PCaiup[][] array entry.

Table 4 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The AP time slot is being used by diagnostics, but the PCM highway map field (hwymap[]) in the PCaiup[][] array is not marked busy, the partial map field [] in PCaiup[][] is not marked completely full, or the rSMEST tuple does

	indicate that there is maintenance activity (rSMEST.stat_act) on the associated line card.
Possible Error Effect:	Maintenance activities may erroneously try to use this AP time slot.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCaiup[][] array entry found to be in error.
<i>Bad Data =</i>	Maintenance activity (rSMEST.stat_act) data.
<i>Good Data =</i>	Relative PIDB number.
<i>Logical Index =</i>	Time slot number.
<i>Dump =</i>	PCaiup[][] array entry found to be in error.

Table 5 Error Code: CDCONFIG

Error Code:	CDCONFIG
Error Description:	The AP time slot is connected to side 1, according to the setting of the compidb[] field in the PCaiup[][] array, but this is a simplex configuration, so there should be no connections to side 1.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rUNITDATA tuple containing the AP circuit information.
<i>Bad Data =</i>	The unit number that is under test.
<i>Good Data =</i>	Contents of the compidb[] field.
<i>Logical Index =</i>	Time slot number.
<i>Dump 1 =</i>	PCaiup[][] array entry found to be in error.
<i>Dump 2 =</i>	The rUNITDATA tuple.

Table 6 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	An attempt was made to read the rEQUIPDSL tuple for either a quarter or non-ODB full time slot based on the logical PH stored in the rIPTSB, but the rEQUIPDSL tuple was found not to exist.
Possible Error Effect:	The data in the rIPTSB tuple is either corrupted or invalid, and it cannot be determined what equipage is associated with the tuple.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Includes the following: • The AIU number in bits 16 through 31. • The AIU AP number in bits 0 through 15.
<i>Bad Data =</i>	Logical PH number.
<i>Good Data =</i>	Logical PH member number.
<i>Logical Index =</i>	Includes the following: • The AP bus number in bits 16 through 31. • The AP time slot number in bits 0 through 15.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	PCaiup[][] array entry.

Table 7 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance state (maint) in PCaiup[][] is not consistent with the maintenance state (rSMEST.bas_stat) found in the corresponding rSMEST tuple for the AP circuit.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The maintenance state in PCaiup[][] is corrected to the maintenance state found in the corresponding rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiup[][] .
<i>Bad Data =</i>	Contents of maintenance state.
<i>Good Data =</i>	Correct maintenance state.
<i>Logical Index =</i>	Circuit name from the rUNITDATA tuple.
<i>Dump =</i>	rSMEST tuple.

Table 8 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The relative PIDB field (compidb[]) in the PCaiupdb[][] array for this time slot is marked busy, but the state field (rIPTSB.use) in the corresponding rIPTSB tuple indicates that this time slot is idle or invalid.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiup[][] .
<i>Bad Data =</i>	State field (rIPTSB.use) from the rIPTSB tuple.
<i>Good Data =</i>	Time slot number.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple.
<i>Dump 1=</i>	rIPTSB tuple.
<i>Dump 2=</i>	PCaiupdb[][] array entry found to be in error.

Table 9 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	An ODD inconsistency has been found between the RLUNITDATA (that is, RLCKTDATA and RLUNIT_CKT) and the RLAIUPIDB relations.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rIAIUPIDB tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rIAIUPIDB tuple should be inserted into the database with a key (rIAIUPIDB.picb) or PICB number that matches the value of the <i>Bad Data</i> in the Dump Description .
Dump Description:	
<i>Error Address =</i>	Includes the following: • The AIU number in bits 16 through 31. • The AIU AP number in bits 0 through 15.
<i>Bad Data =</i>	PICB number.
<i>Good Data =</i>	Contents of the AP highway to PIDB map (PCaiup[][].compidb[]).
<i>Logical Index =</i>	Includes the following: • The AP bus number in bits 16 through 31. • The AP time slot number in bits 0 through 15.

<i>Dump =</i>	rUNITDATA tuple.

Table 10 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The maintenance state (maint) in PCaiup[][] is not out-of-service (OOS) when rUNITDATA tuple is not found.
Possible Error Effect:	Call processing may try to hunt to an unequipped AP.
Corrective Action Taken:	The maintenance state in PCaiup[][] is corrected to OOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiup[][].
<i>Bad Data =</i>	Contents of maintenance state.
<i>Good Data =</i>	SMOOS.
<i>Logical Index =</i>	Circuit index from the rUNITDATA tuple.
<i>Dump =</i>	PCaiupidb[][] array entry found to be in error.

Table 11 Error Code: PTSB

Error Code:	PTSB
Error Description:	The relative PIDB field (compidb[]) is corrupt, causing the access of the corresponding rIPTSB tuple to fail.
Possible Error Effect:	The AP time slot is lost to the system, causing call processing to degrade.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple.
<i>Bad Data =</i>	Relative PIDB number.
<i>Good Data =</i>	Base address of the AP circuit.
<i>Logical Index =</i>	Logical key (rIPTSB.ptsb) of the rIPTSB tuple.
<i>Dump 1=</i>	rIPTSB tuple.
<i>Dump 2=</i>	PCaiup[][].

Table 12 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The relative PIDB field (compidb[]) in the PCaiup[][] array for this time slot is marked busy, but the port information data in the rIPTSB tuple does not point back to this AP time slot.
Possible Error Effect:	The AP time slot is lost to the system causing call processing to degrade.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle, the bit in the partial map field (partmap) is set to completely full, and the relative PIDB field (compidb[]) is set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Includes the following: - If bad data is 0, return value from the PCpt2ptinfo() primitive. - If bad data is 1, return value from reading the rCKTDATA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	AP number.
<i>Logical Index =</i>	Logical key (rIPTSB.ptsb) of the rIPTSB tuple.
<i>Dump 1=</i>	rIPTSB tuple.
<i>Dump 2=</i>	PCaiup[][].
<i>Dump 3=</i>	Port data information of type PCPXCXINFO.

Table 13 Error Code: RELPIDB

Error Code:	RELPIDB
Error Description:	The AP time slot is being used for maintenance activities, but the relative PIDB

	field (compidb[]) does not contain a valid line card circuit index.
Possible Error Effect:	Maintenance activities may erroneously try to use this AP time slot.
Corrective Action Taken:	The relative PIDB field (compidb[]) and the PCM highway map field (hwymap[]) are set to idle, and the partial map field (partmap[]) is set to completely full in the PCaiupap[][] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCaiupap[][] array entry.
<i>Bad Data =</i>	Relative PIDB data (compidb[]) from PCaiupap[][].
<i>Good Data =</i>	Circuit index from the rUNIT_CKT tuple.
<i>Logical Index =</i>	Time slot number.
<i>Dump =</i>	PCaiupap[][] array entry found to be in error.

Table 14 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	A rUNITDATA static tuple exists in the database that contains a AP circuit name (rUNITDATA.cktname), but the corresponding static rAIU2AP tuple for this AIU (rAIU2AP.aiuno) is missing from the database.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rAIU2AP tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rAIU2AP tuple should be inserted into the database with a key (rAIU2AP.aiuno) or AIU unit number that matches the value of <i>Logical Key</i> in the Dump Description .
Dump Description:	
<i>Error Address =</i>	Address of PCaiupap[][] array entry.
<i>Bad Data =</i>	RLAIU2AP relation identification.
<i>Good Data =</i>	RLUNITDATA relation identification.
<i>Logical Index =</i>	AIU unit number.
<i>Dump =</i>	rUNITDATA tuple.

Table 15 Error Code: SMEST

Error Code:	SMEST
Error Description:	Access of the rSMEST tuple for either the AP circuit associated with the PCaiupap[][] array entry has failed.
Possible Error Effect:	The AIUAP time slot map will skip auditing the PCaiupap[][] data associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	If <i>Good Data</i> is 0, then this is the Address of the PCaiupap[][] array entry. If <i>Good Data</i> is 1, then this is the AIU unit number shifted up 16 bits or'ed with the AP number.
<i>Bad Data =</i>	Address of the rSMEST tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rSMEST key.
<i>Dump =</i>	rUNITDATA tuple if <i>Good Data</i> is 0 or rUNIT_CKT tuple if <i>Good Data</i> is 1.

Table 16 Error Code: TSBUSYIDLE

Error Code:	TSBUSYIDLE
Error Description:	The relative PIDB field (compidb[]) in the PCaiupap[][] array for this AP time slot is marked busy, but the PCM highway map field (hwymap[]) in the PCaiupap[][] array is not marked busy, or the partial map field [][] in PCaiupap[][] is not marked completely full.
Possible Error Effect:	Call processing or maintenance activities accessing invalid data may assert.
Corrective Action Taken:	The PCM highway map field (hwymap[]) in the PCaiupap[][] array for this AP time slot is marked busy to match the state of the corresponding rIPTS tuple and the partial map field (partmap[]) is marked completely full.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCaiupap[][].

<i>Bad Data</i> =	AP number.
<i>Good Data</i> =	Relative PIDB number.
<i>Logical Index</i> =	Time slot number.
<i>Dump 1</i> =	rPTSB tuple if <i>Bad Data</i> is 1, 2, or 3. PCaiup[][] array entry if <i>Bad Data</i> is 4.
<i>Dump 2</i> =	PCaiup[][] array if <i>Bad Data</i> is 1, 2, or 3.

Table 17 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	The relative PIDB field (compidb[]) in the PCaiup[][] array for this time slot is marked idle, but the corresponding bit in PCM highway field (hwymap) is not marked idle, or the bit in the partial map field (partmap) is not marked as completely full.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The corresponding bit in the PCM highway field (hwymap) is set to idle and the bit in the partial map field (partmap) is set to completely full.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCaiup[][].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Relative PIDB data (compidb[]).
<i>Logical Index</i> =	Index into the relative PIDB array (compidb[]).
<i>Dump</i> =	PCaiup[][] array entry found to be in error.

AIUPIDB-A

Software Release: 5E11 - 5E13

1. ACCESS INTERFACE UNIT PERIPHERAL INTERFACE DATA BUS (AIUPIDB) TIME SLOT MAP AUDIT

The access interface unit (AIU) peripheral interface data bus (PIDB) time slot map is used by peripheral control to provide maintenance and call processing status for PIDB time slots which are connected to the AIU using the time slot interchanger (TSI).

2. DATA STRUCTURES

For details on the fields contained in the structures mentioned in the error descriptions below, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_PTSB

Error Code:	ASSOC_PTSB
Error Description:	The associate rIPTSb tuple indicates that this PIDB time slot is involved in a on-demand B-channel (ODB) call, however the common data usage (rIPTSb.cduse) attribute in the rIPTSb tuple does not contain a valid common data usage type.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The PIDB busy/idle status map for the time slot under audit will be idle, the associated rIPTSb tuple will be quarantined, and the PTSB audit will scheduled as a related audit to recover the inconsistent rIPTSb tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	0
<i>Good Data =</i>	1
<i>Logical Index =</i>	Logical key (rIPTSb.ptsb) of the rIPTSb tuple.
<i>Dump 1=</i>	PCaiupidb[][] array entry.
<i>Dump 2=</i>	rIPTSb tuple.

Table 2 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	The PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1 does not match the PIDB busy/idle status map (PCaiupidb[][].pidbmap[0]) for service group 0.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1 will be made consistent with the PIDB busy/idle status map (PCaiupidb[][].pidbmap[0]) for service group 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	Contents of the PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1.

<i>Good Data</i> =	Contents of the PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1.
<i>Logical Index</i> =	Includes the following: • The AIU number in bits 16 through 31. • The AIU PIDB number in bits 0 through 15.
<i>Dump</i> =	rUNITDATA tuple.

Table 3 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	This error is used in multiple places and can be distinguished by the dump data. Case 1: The setting of the clear channel (clrchan[]) could not be determined from the data in rUNITDATA. Case 2: The system peripheral interface data bus (PIDB) number (syspidb[]), PIDB type (type[]) or the clear channel (clrchan[]) data in PCaiupidb[] is not correct.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	Includes the following: Case 1: All checks of the PCaiupidb[] are skipped for all additional time slots per PIDB number. Case 2: The system PIDB number, the PIDB type, and the clear channel data in PCaiupidb[][] is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	Includes the following: • 0 for case 1. • The clear channel in PCaiupidb[] for case 2.
<i>Good Data</i> =	The clear channel (clrchan[]) attribute.
<i>Logical Index</i> =	Circuit name of rUNITDATA tuple.
<i>Dump 1</i> =	Includes the following: • PCaiupidb[][] for case 1. • rUNITDATA tuple for case 2.
<i>Dump 2</i> =	Includes the following: • The rUNITDATA tuple for case 1. • PCaiupidb[][] for case 2.

Table 4 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance state (maint[]) in PCaiupidb[][] is not consistent with the maintenance state (rSMEST.bas_stat) found in the corresponding rSMEST tuple for the PIDB circuit.

Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The maintenance state in PCaiupdb[][] is corrected to the maintenance state found in the corresponding rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupdb[][].
<i>Bad Data</i> =	Contents of maintenance state.
<i>Good Data</i> =	Correct maintenance state.
<i>Logical Index</i> =	Circuit name from the rUNITDATA tuple.
<i>Dump 1</i> =	rSMEST tuple associated with the circuit.
<i>Dump 2</i> =	rSMEST tuple associated with the common data and control circuit (COMDAC).

Table 5 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static rDPIDB_PT tuples was found missing from the ODD when a corresponding static rUNITDATA tuple exists for an AIU DPIDB circuit.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rDPIDB_PT tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rDPIDB_PT tuple should be inserted into the database with a DPIDB key (rDPIDB_PT.dpidb) value that matches the value of <i>Bad Data</i> in the Dump Description and a key slot (rDPIDB_PT.key_slot) value of 0 and 16.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupdb[][].
<i>Bad Data</i> =	DPIDB value.
<i>Good Data</i> =	1
<i>Logical Index</i> =	Circuit name from the rUNITDATA tuple.
<i>Dump 1</i> =	PCaiupdb[][] array entry.
<i>Dump 2</i> =	rUNITDATA tuple.

Table 6 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The maintenance state (maint) in PCaiupdb[][] is not out-of-service (OOS) or the PIDB type is not PCNULLPIDB when rUNITDATA tuple is not found.
Possible Error Effect:	Call processing may try to hunt to an unequipped PIDB time slot.
Corrective Action Taken:	The maintenance state in PCaiupdb[][] is corrected to OOS and the PIDB type is corrected to PCNULLPIDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupdb[][].
<i>Bad Data</i> =	Contents of maintenance state (PCaiupdb[][].maint).
<i>Good Data</i> =	Contents of the PIDB type (PCaiupdb[][].type).
<i>Logical Index</i> =	Circuit index.
<i>Dump</i> =	PCaiupdb[][].

Table 7 Error Code: PTSB

Error Code:	PTSB
Error Description:	The logical key of the rIPTSB tuples associated with this PCaiupdb[][] is corrupt.
Possible Error Effect:	The AIUPIDB time slot map will skip auditing the PCaiupdb[][] data associated with the corrupt rIPTSB tuple. Also, calls trying to use this PIDB may fail.
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple.
<i>Bad Data</i> =	Relative PIDB number.
<i>Good Data</i> =	Time slot number.
<i>Logical Index</i> =	Logical key (rIPTSB.ptsb) of the rIPTSB tuple.
<i>Dump 1</i> =	rIPTSB tuple.
<i>Dump 2</i> =	PCaiupdb[][].

Table 8 Error Code: SMEST

Error Code:	SMEST
Error Description:	Access of the rSMEST tuple for either the PIDB or COMDAC associated with the PCaiupidb[][] array entry has failed.
Possible Error Effect:	The AIUPIDB time slot map will skip auditing the PCaiupidb[][] data associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	COMDAC circuit name.
<i>Good Data =</i>	Base address of the PIDB circuit.
<i>Logical Index =</i>	rSMEST key.
<i>Dump =</i>	rUNITDATA tuple.

Table 9 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rIPTSB tuple contains an invalid usage (rIPTSB.use) type.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The PIDB time slot usage (pidbmap[]) in PCaiupidb[][] is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	rIPTSB tuple usage type (rIPTSB.use).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIPTSB tuple (rIPTSB.ptsb).
<i>Dump 1=</i>	rIPTSB tuple.
<i>Dump 2=</i>	PCaiupidb[][].

Table 10 Error Code: TSBUSYIDLE

Error Code:	TSBUSYIDLE
Error Description:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked busy, but the state field (rIPTSB.use) in the corresponding rIPTSB tuple indicates this time slot is idle.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked idle to match the state of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	0.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	Logical key of the rIPTSB tuple (rIPTSB.ptsb).
<i>Dump 1=</i>	PCaiupidb[][].
<i>Dump 2=</i>	rIPTSB tuple.

Table 11 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked idle, but the state field (rIPTSB.use) in the corresponding rIPTSB tuple indicates that this time slot is busy.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked busy to match the state (rIPTSB.use) of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	1.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the rIPTSB tuple (rIPTSB.ptsb).

<i>Dump 1</i> =	PCaiupidb[][].
<i>Dump 2</i> =	rIPTSB tuple.

AIUPIDB-B

Software Release: 5E14 and later

1. ACCESS INTERFACE UNIT PERIPHERAL INTERFACE DATA BUS (AIUPIDB) TIME SLOT MAP AUDIT

The access interface unit (AIU) peripheral interface data bus (PIDB) time slot map is used by peripheral control to provide maintenance and call processing status for PIDB time slots which are connected to the AIU using the time slot interchanger (TSI).

2. DATA STRUCTURES

For details on the fields contained in the structures mentioned in the error descriptions below, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_PTSB

Error Code:	ASSOC_PTSB
Error Description:	The associate rIPTSb tuple indicates that this PIDB time slot is involved in a on-demand B-channel (ODB) call, however the common data usage (rIPTSb.cduse) attribute in the rIPTSb tuple does not contain a valid common data usage type.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The PIDB busy/idle status map for the time slot under audit will be idle, the associated rIPTSb tuple will be quarantined, and the PTSB audit will be scheduled as a related audit to recover the inconsistent rIPTSb tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	0
<i>Good Data =</i>	1
<i>Logical Index =</i>	Logical key (rIPTSb.ptsb) of the rIPTSb tuple.
<i>Dump 1=</i>	PCaiupidb[][] array entry.
<i>Dump 2=</i>	rIPTSb tuple.

Table 2 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	The PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1 does not match the PIDB busy/idle status map (PCaiupidb[][].pidbmap[0]) for service group 0.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1 will be made consistent with the PIDB busy/idle status map (PCaiupidb[][].pidbmap[0]) for service group 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCaiupidb[][].
<i>Bad Data =</i>	Contents of the PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1.

<i>Good Data</i> =	Contents of the PIDB busy/idle status map (PCaiupidb[][].pidbmap[1]) for service group 1.
<i>Logical Index</i> =	Includes the following: • The AIU number in bits 16 through 31. • The AIU PIDB number in bits 0 through 15.
<i>Dump</i> =	rlUNITDATA tuple.

Table 3 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	This error is used in multiple places and can be distinguished by the dump data. Case 1: The setting of the clear channel (clrchan[]) could not be determined from the data in rlUNITDATA. Case 2: The system PIDB number (syspidb[]), PIDB type (type[]) or the clear channel (clrchan[]) data in PCaiupidb[] is not correct.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	Includes the following: Case 1: All checks of the PCaiupidb[] are skipped for all additional time slots per PIDB number. Case 2: The system PIDB number, the PIDB type, and the clear channel data in PCaiupidb[][] is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	Includes the following: • 0 for case 1. • The clear channel in PCaiupidb[] for case 2.
<i>Good Data</i> =	The clear channel (clrchan[]) attribute.
<i>Logical Index</i> =	Circuit name of rlUNITDATA tuple.
<i>Dump 1</i> =	Includes the following: • PCaiupidb[][] for case 1. • rlUNITDATA tuple for case 2.
<i>Dump 2</i> =	Includes the following: • The rlUNITDATA tuple for case 1. • PCaiupidb[][] for case 2.

Table 4 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	A rlAPP2TSP static tuple exists in the database that contains a transport circuit name (rlAPP2TSP.tsp_name), but the corresponding static rlCKTDATA tuple is missing from the database.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rlCKTDATA tuple.

Corrective Action Taken:	None.
Manual Action Required:	A rCKTDATA tuple should be inserted into the database with a key (rCKTDATA.name) or circuit name that matches the value of <i>Bad Data</i> in the Dump Description .
Dump Description:	
<i>Error Address</i> =	Address of PCaiupdb[][].
<i>Bad Data</i> =	Circuit name of the application circuit.
<i>Good Data</i> =	Circuit name of the transport circuit.
<i>Logical Index</i> =	Circuit name of the PIDB circuit.
<i>Dump</i> =	rUNITDATA tuple.

Table 5 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance state (maint[]) in PCaiupdb[][] is not consistent with the maintenance state (rSMEST.bas_stat) found in the corresponding rSMEST tuple for the PIDB circuit.
Possible Error Effect:	May cause calls through the AIU to fail.
Corrective Action Taken:	The maintenance state in PCaiupdb[][] is corrected to the maintenance state found in the corresponding rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupdb[][].
<i>Bad Data</i> =	Contents of maintenance state.
<i>Good Data</i> =	Correct maintenance state.
<i>Logical Index</i> =	Circuit name from the rUNITDATA tuple.
<i>Dump 1</i> =	rSMEST tuple associated with the PIDB circuit.
<i>Dump 2</i> =	rSMEST tuple associated with the common data and control circuit (COMDAC).

Table 6 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	If <i>Good Data</i> is 0, then a rUNITDATA tuple exists in the database that is associate with a AIU PIDB circuit (rUNITDATA.name), but the corresponding static rAPP2TSP tuple is missing from the database. If <i>Good Data</i> is 1, then a static rDPIDB_PT tuples was found missing from the ODD when a corresponding static rUNITDATA tuple exists for an AIU DPIDB circuit.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rAPP2TSP or rDPIDB_PT tuple.
Corrective Action Taken:	None.
Manual Action Required:	If <i>Good Data</i> is 0, then a rAPP2TSP tuple should be inserted into the database with a key (rAPP2TSP.app_name) or circuit name that matches the value of <i>Logical Key</i> in the Dump Description . If <i>Good Data</i> is 1, then a rDPIDB_PT tuple should be inserted into the database with a DPIDB key (rDPIDB_PT.dpidb) value that matches the value of <i>Bad Data</i> in the Dump Description and a key slot (rDPIDB_PT.key_slot) value of 0 and 16.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupdb[][].
<i>Bad Data</i> =	Circuit name of the PIDB circuit.
<i>Good Data</i> =	Includes the following: 0 if a rAPP2TSP tuple is missing from the database. 1 if a rDPIDB_PT tuple is missing from the database.
<i>Logical Index</i> =	Circuit name of the PIDB circuit.
<i>Dump 1</i> =	Includes the following: If <i>Good Data</i> is 0, rUNITDATA tuple.

	If <i>Good Data</i> is 1, PCaiupidb[][] array entry.
<i>Dump 2</i> =	If <i>Good data</i> is 1, rUNITDATA tuple.

Table 7 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The maintenance state (maint) in PCaiupidb[][] is not out-of-service (OOS) or the PIDB type is not PCNULLPIDB when rUNITDATA tuple is not found.
Possible Error Effect:	Call processing may try to hunt to an unequipped PIDB time slot.
Corrective Action Taken:	The maintenance state in PCaiupidb[][] is corrected to OOS and the PIDB type is corrected to PCNULLPIDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	Contents of maintenance state (PCaiupidb[][].maint).
<i>Good Data</i> =	Contents of the PIDB type (PCaiupidb[][].type).
<i>Logical Index</i> =	Circuit index.
<i>Dump</i> =	PCaiupidb[][].

Table 8 Error Code: PTSB

Error Code:	PTSB
Error Description:	The logical key of the rIPTSB tuples associated with this PCaiupidb[][] is corrupt.
Possible Error Effect:	The AIUPIDB time slot map will skip auditing the PCaiupidb[][] data associated with the corrupt rIPTSB tuple. Also, calls trying to use this PIDB may fail.
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple.
<i>Bad Data</i> =	Relative PIDB number.
<i>Good Data</i> =	Time slot number.
<i>Logical Index</i> =	Logical key (rIPTSB.ptsb) of the rIPTSB tuple.
<i>Dump 1</i> =	rIPTSB tuple.
<i>Dump 2</i> =	PCaiupidb[][].

Table 9 Error Code: SMEST

Error Code:	SMEST
Error Description:	Access of the rSMEST tuple for either the PIDB or COMDAC associated with the PCaiupidb[][] array entry has failed.
Possible Error Effect:	The AIUPIDB time slot map will skip auditing the PCaiupidb[][] data associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	COMDAC circuit name.
<i>Good Data</i> =	Base address of the PIDB circuit.
<i>Logical Index</i> =	rSMEST key.
<i>Dump</i> =	rUNITDATA tuple.

Table 10 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rIPTSB tuple contains an invalid usage (rIPTSB.use) type.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The PIDB time slot usage (pidbmap[]) in PCaiupidb[][] is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	rIPTSB tuple usage type (rIPTSB.use).
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Logical key of the rIPTSB tuple (rIPTSB.ptsb).
<i>Dump 1</i> =	rIPTSB tuple.
<i>Dump 2</i> =	PCaiupidb[][].

Table 11 Error Code: TSBUSYIDLE

Error Code:	TSBUSYIDLE
Error Description:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked busy, but the state field (rIPTSB.use) in the corresponding rIPTSB tuple indicates this time slot is idle.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked idle to match the state of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	0.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Logical key of the rIPTSB tuple (rIPTSB.ptsb).
<i>Dump 1</i> =	PCaiupidb[][].
<i>Dump 2</i> =	rIPTSB tuple.

Table 12 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked idle, but the state field (rIPTSB.use) in the corresponding rIPTSB tuple indicates that this time slot is busy.
Possible Error Effect:	Call processing using this invalid data may cause asserts or single process purges.
Corrective Action Taken:	The state field (pidbmap[]) in the PCaiupidb[][] array for this time slot is marked busy to match the state (rIPTSB.use) of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCaiupidb[][].
<i>Bad Data</i> =	1.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the rIPTSB tuple (rIPTSB.ptsb).
<i>Dump 1</i> =	PCaiupidb[][].
<i>Dump 2</i> =	rIPTSB tuple.

ALCB-A

Software Release: 5E10 - 5E11

1. ACCESS LINE CONTROL BLOCK (ALCB) AUDIT

The ALCB audit is part of the Switch Maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The ALCB audit uses the packet switching access line control block array (PSalcb), the logical control block array (PSlcb), and the logical channel control block array (PSlccb).

3. AUDIT CHECKS

The audit makes linkage checks within the PSalcb array with the logical link control block (PSlcb) and the logical channel control block array (PSlccb). It invokes its semantic check function to verify the correctness of data in the PSalcb. For on-demand B-channel (ODB) packet services, the audit will turn on the t3xx timer if there is no packet activity, the timer is not currently running, and the user is a subscriber to the t3xx timer feature.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ACCPTR

Error Code:	ACCPTR
Error Description:	ALCB access pointer is not 0. The audit found an ALCB with access pointer not set to 0.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit reports the error and sets the ALCBs al_sid field to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCBs al_sid field. Access pointer should be 0.
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: ACPTRBAD

Error Code:	ACPTRBAD
Error Description:	Action pointer bad. The audit found an ALCB with an invalid action field "al_action".
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCBs action (al_action) field.
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 3 Error Code: ACTION

Error Code:	ACTION
Error Description:	An ALCB appears to be in use for modem-pool testing (ALCB.al_action =

	PSXMPTM) but either its state is incorrect (ALCB.al_rst_state != PS_R255_MPTM) or it has more than one virtual call up (ALCB.al_actvc > 1).
Possible Error Effect:	If calls or modem pool tests really are up on this ALCB, they will fail.
Corrective Action Taken:	The audit will report the error and recover the ALCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ALCB (& PSalcb[key given]).
<i>Bad Data</i> =	ALCB's state (ALCB.al_rst_state).
<i>Good Data</i> =	PS_R255_MPTM.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 4 Error Code: ALDBRANGE

Error Code:	ALDBRANGE
Error Description:	The audit has found a LAPD ALCB whose "aldb_index" field is out of range (greater than null).
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's aldb_index field.
<i>Good Data</i> =	0. The value chosen here is 0 but any value between 0 and 128 which gives proper linkage between DCCB, LLCB and ALCB is a good value.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 5 Error Code: ALINK

Error Code:	ALINK
Error Description:	The audit found an ALCB in the R255 modem pool terminal maintenance state which has more than one LCCB attached to it.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Count of the number of LCCB's linked to the ALCB.
<i>Good Data</i> =	ALCB index.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 6 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The virtual call throughput class for the sending direction of an LCCB linked to an ALCB is out of range (PSTPC75 > PSlccb[].vc_tpc[PSCED] > PSTPC19200).
Possible Error Effect:	Calls associated with this ALCB could be blocked when they shouldn't be.
Corrective Action Taken:	The audit will set the quarantine bit in the LCCB. The LCCB audit will run as a related audit and will recover the LCCB to the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSlccb[].vc_tpc[PSCED].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 7 Error Code: BADOWNER

Error Code:	BADOWNER

Error Description:	The audit found an ALCB with the transient state bit "al_notlinked" set. The ALCB has a non-null LLCB index field and should not be attached to any LLCB when this bit is set.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB LLCB value (licb_index).
<i>Good Data =</i>	PSLLCBNULL - Valid null value for this field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	PH channel number, al_phcn.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The virtual call throughput class for the receiving direction of an LCCB linked to an ALCB is out of range (PSTPC75 > PSlccb[].vc_tpc[PSCING] > PSTPC19200).
Possible Error Effect:	Calls associated with this ALCB could be blocked when they shouldn't be.
Corrective Action Taken:	The audit will set the quarantine bit in the LCCB. The LCCB audit will run as a related audit and will recover the LCCB to the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSlccb[].vc_tpc[PSCING].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The incoming logical channel number count for multi-line hunt group (MLHG) members of an ALCB is incorrect (PSalcb[].al_inlcn_mlhg). This field should be equal to the number of LCCBs, linked to this ALCB, that meet the following conditions: - The LCCB Busy Limit Parameter flag is set (PSlccb[].blp_incr == DBYES). - The LCCB is associated with an incoming call (PSlccb[].aphend == PSCED).
Possible Error Effect:	MLHG calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will update the PSalcb[].al_inlcn_mlhg field to agree with the number of LCCBs that meet the previous conditions.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_inlcn_mlhg.
<i>Good Data =</i>	The number of LCCBs that meet the conditions above.
<i>Logical Index =</i>	ALCB index.
<i>Dump 1 =</i>	The ALCB fields declared before the PSalcb[].lccb_index[] fields.
<i>Dump 2 =</i>	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit found an ALCB in an inter-SM PH that is linked to a valid LLCB, but there is also a non-null LCCB field in the ALCB data structure. If the ALCB is for an inter-SM PH, there should NOT be any LCCBs linked.

Possible Error Effect:	Prints a report and corrects the erroneous fields.
Corrective Action Taken:	The audit prints an error report and sets the ALCBs lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into the ALCB's lccb_index[].
<i>Bad Data =</i>	ALCB's LCCB value in lccb_index[] array that is non-null.
<i>Good Data =</i>	PSLCCBNULL - Valid null value for this field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 11 Error Code: BDPCR

Error Code:	BDPCR
Error Description:	BD pointer attached to ALCB is corrupted. The audit found an ALCB with the pvcsetup field enabled but there was a null BD field (al_pvcmap). If the pvcsetup field is set, there must be a BD attached.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit reports the error and sets the ALCBs al_pvcsetup BD field to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCBs al_pvcmap. The ALCBs PVC BD field. This field should be non-null when pvcsetup is set.
<i>Good Data =</i>	PSNULLOFF - Valid null value for the pvcmap field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 12 Error Code: CAMPSTATE

Error Code:	CAMPSTATE
Error Description:	When al_campon field indicated that a campon is going on (PSENABLED), audit found invalid value of the al_rst_state field in ALCB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field is used to dump the value of al_campon field of ALCB.
<i>Bad Data =</i>	ALCB's al_rst_state field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 13 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found an ALCB that should have multi-line hunt group information, since the ALDB's huntflag indicates this is a multi-line hunt group channel, but the count of LCCBs linked to this ALCB with their "huntflag" set does not match the value of the ALCB's "al_usage".
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's "al_usage".
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Highest LCN value of ALCB.
<i>Bad Data =</i>	ALCB's count of LCCBs with the huntflag enabled.
<i>Good Data =</i>	Audits count of the total number of LCCB's with the huntflag enabled.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 14 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The audit found an ALCB linked to a valid LLCB, but the ALCB's PH channel

	number doesn't match the LLCB's D channel number.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit prints an error report and sets the ALCB's PH channel number (al_phcn) to the LLCB's D channel number (dchan).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's LLCB field (llcb-index).
<i>Bad Data =</i>	ALCB's channel number (al_phcn) the value of the current ALCB. This should equal the LLCB's dchan.
<i>Good Data =</i>	ALCB's LLCB's D channel number (dchan).
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 15 Error Code: DFLG

Error Code:	DFLG
Error Description:	MLHG flags should be DISABLED The audit found an ALCB with PVC channels that have their LCCB's huntflag enabled and should be null. PVC's cannot be of the huntable type.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the ALCB's LCCB's huntflag to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into the ALCB's lccb_index[] array.
<i>Bad Data =</i>	ALCB's LCCB (indicated by the lccb_index[] array mentioned in error address) whose huntflag is enabled.
<i>Good Data =</i>	PSDISABLE - Valid null value that the LCCB's huntflag should have.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 16 Error Code: DISP

Error Code:	DISP
Error Description:	Should be a disabled pointer. The audit found an ALCB with the action field "al_action" set to NULL action, but either the 13changed, pvcsetup or the pvcmap fields is not disabled or null. None of these fields can be set when in the NULL action condition.
Possible Error Effect:	Prints a report and corrects the ALCB fields.
Corrective Action Taken:	The audit reports the error and sets the "al_13changed" and "al_pvcsetup" fields to Disabled and nulls the pvcmap BD field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's 13changed field (al_13chnegd). This should be PSDISABLED.
<i>Bad Data =</i>	ALCB's pvcsetup (al_pvcsetup) field. This should be PSDISABLED.
<i>Good Data =</i>	PSDISABLED - valid null field for the al_13changed and al_pvcsetup fields.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	(PSOFFSET size) ALCB's PVC BD (al_pvcmap).

Table 17 Error Code: EVEN_COUNT

Error Code:	EVEN_COUNT
Error Description:	The combined throughput sum for the sending direction of an ALCB is incorrect (PSalcb[].al_sum_snd). This sum should be equal to the throughput class for the sending direction (PSlccb[].vc_tpc[PSCED]) of all LCCBs, linked to this ALCB, that have the maximum combined throughput increment flag set (PSlccb[].mct_incr == DBYES).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will update the PSalcb[].al_sum_snd to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_sum_snd.
<i>Good Data =</i>	The correct value for the PSalcb[].al_sum_snd field.

<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB fields declared before the PSalcb[].lccb_index[] fields.
<i>Dump 2</i> =	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 18 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Multi-line hunt group (MLHG). The audit found an ALCB that should have multi-line huntgroup information, since the ALDB's huntflag indicates this is a multi-line hunt group channel, but the information does not match what is in the ALDB, or the al_usage field is not set to indicate that this is a multi-line hunt group channel. The audit totals the number of ALDB LCN-in and the 2-way LCNs (which is the count of hunttable LCNs) and compares it with the ALCB's "al_lcnmlhg" field finding them inconsistent.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's al_lcnmlhg.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's count of MLHG LCN's (al_lcnmlhg).
<i>Good Data</i> =	Audits count of the total of 2way plus the lcn-in.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 19 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	PVC group type number. The audit found an ALCB with an invalid value for the highest PVC lcn value. When comparing the ALCB's "al_hipvc" against the access line data block's (ALDB's) write protected area, pvc_h_lcn, which is the highest LCN value for a PVC, the two values did not match.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the PVC high logical channel number (al_hipvc) field to that of the DCDB write protected data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's highest PVC LCN number (al_pvclcn).
<i>Good Data</i> =	ALDB's highest LCN value for a PVC.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB up to the beginning of field ALCB.lccb_index[].
<i>Dump 2</i> =	The ALCB from the end of field ALCB.lccb_index[] on.

Table 20 Error Code: HIM_NUM

Error Code:	HIM_NUM
Error Description:	The audit found an ALCB with an invalid high logical channel number (al_hilcn) field. This should match the first non-null value in the write protected data ALDB when looking at the h_lcn_out fields down to pvc_h_lcn.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the ALCB high LCN (al_hilcn) field to that of the ALDB write protected data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's high LCN number (al_hilcn).
<i>Good Data</i> =	ALDB's highest value for an LCN.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB up to the beginning of field ALCB.lccb_index[].
<i>Dump 2</i> =	The ALCB from the end of field ALCB.lccb_index[] on.

Table 21 Error Code: IDLE_LINK

Error Code:	IDLE_LINK

Error Description:	The audit has found a LAPB ALCB whose "aldb_index" field is out of range (either null or greater than null).
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's aldb_index field.
<i>Good Data</i> =	0. The value chosen here is 0 but any value between 0 and 127 which gives proper linkage between DCCB, LLCB and ALCB is good value.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 22 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	LCCB linked to ALCB when it should be idle. The audit found an ALCB with a non-null LCCB value in a lccb_index[] array slot which is a slot value greater than the highest logical channel number (LCN) value that the write protected data block (ALDB) indicates is valid for this channel. Basically, all slot values greater than what the ALDB indicates is the highest LCN value, must be null and the audit found a non-null value.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's lccb_index[] element.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index into the ALCB's lccb_index[] array element.
<i>Bad Data</i> =	ALCB's LCCB value in the lccb_index[] array element.
<i>Good Data</i> =	ALCB's highest LCN field - al_hilcn.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	PH channel number (ALCB → al_phcn) of PSCHNUM type.

Table 23 Error Code: INDEX

Error Code:	INDEX
Error Description:	The audit found an ALCB that is linked to a valid LLCB, but a LCCB field is out of range.
Possible Error Effect:	Prints a report and corrects the ALCB's LCCB field.
Corrective Action Taken:	The audit prints an error report and set the ALCB's lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's index into the lccb_index[] array.
<i>Bad Data</i> =	ALCB's LCCB value that is out of range.
<i>Good Data</i> =	PSHMAXLCCB - One more than the maximum value the LCCB field can hold.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 24 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	When al_action field indicated that a campon is going on, audit found invalid value of the al_rst_state field in ALCB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field is used to dump the value of al_action field of ALCB.
<i>Bad Data</i> =	ALCB's al_rst_state field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 25 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The audit found an ALCB on the free linked list with a non-null LCCB index field. The free linked list should not have any LCCBs attached to the ALCB.
Possible Error Effect:	Prints a report and corrects the erroneous field.
Corrective Action Taken:	The audit prints an error report and null, the LCCB field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Element number of the lccb_index array that is non-null.
<i>Bad Data =</i>	ALCB's LCCB value.
<i>Good Data =</i>	PSLCCBNULL - Valid null value for this field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 26 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The audit found an ALCB with the al_notlinked bit set indicating that the ALCB is in a transient state of being torn down but the state is not R4. This is the only valid state for an ALCB when this bit is set.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	al_phcn field of ALCB PH channel number.
<i>Bad Data =</i>	ALCB's state (al_rst_state). This should be R4.
<i>Good Data =</i>	al_action field of ALCB.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 27 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The audit found an ALCB on the free linked list and either there is a timer tag record in the al_tim array or the PVC map field indicating a buffer descriptor (BD) is not null.
Possible Error Effect:	Prints a report and corrects the fields.
Corrective Action Taken:	The audit reports the error and sets the al_tim timer array fields and the pvcmap field to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's al_pvcmap (BD) field. This should be PSNULLOFF.
<i>Bad Data =</i>	ALCB's link timer field (al_tim[PSXT3LNK]).
<i>Good Data =</i>	PSNULLOFF - Valid null value for the al_pvcmap field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	(long) ALCB's link map field (al_tim[PSXT3MAP]).

Table 28 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Number of links to active VCs. The audit found an ALCB with an invalid value for the number of active virtual calls (VC) up on this ALCB (al_actvc). Audits have computed the number of non-null LCCBs on this ALCB and this value does not equal "al_actvc".
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the number of active VCs (al_actvc) to that which the audit found as non-null LCCBs linked to this ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB's number of active VCs (al_actvc).
<i>Good Data =</i>	Audits count of non-null LCCBs attached to ALCB.

<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 29 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The audit found an ALCB linked to a valid LLCB, but the ALCB's forward or back ALCB fields are not null. When the ALCB is linked to a LLCB, it cannot be linked to other ALCB's too.
Possible Error Effect:	Print a report and correct the erroneous fields.
Corrective Action Taken:	The audit will print an error report and set the ALCB's forward and back ALCB fields (fdalb_ind and bkalcb_ind) to null values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's back ALCB field (bkalcb_ind).
<i>Bad Data</i> =	ALCB's forward ALCB field (fdalcb_ind).
<i>Good Data</i> =	PSALCBNULL - Valid null value for these fields.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 30 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The audit found an ALCB with a valid "establish" action field (al_action) but the ALCB's state was not R0 or R3.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit report the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PH channel number.
<i>Bad Data</i> =	ALCB's state (al_rst_state) field.
<i>Good Data</i> =	ALCB's action (al_action) field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 31 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit found a member error in the linked list of the specified audit.
Possible Error Effect:	Prints an error report and corrects the linked list.
Corrective Action Taken:	Returns AUFALSE to calling function which rebuilds the linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Linked list null value.
<i>Logical Index</i> =	AUcount, the current element number.
<i>Dump</i> =	AULNK defined in au/AUaps/AUextern.h (all size unsigned short): • "back" element's back field, • "back" struct.'s owner field, • "back" struct.'s next field, • current element being audited, • current element on this linked list, • current element's back field, • current element's owner,

	• data element null value, • head cell null value, • head cell's first element, • head cell's last element, • "next" element on this linked list, • "next" element's back field, • "next" element's owner field.

Table 32 Error Code: MLNKGRP

Error Code:	MLNKGRP
Error Description:	MLHG link group error in the ALCB. The audit found an ALCB with non-null MLHG (al_lcnmlhg) or usage information of the MLHG (al_usage) when the ALDB write protected data indicates that no MLHG channels can be on this ALCB.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's al_lcnmlhg and al_usage fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's multi-line hunt group flag (al_lcnmlhg). This should be PSNOHUNT_GRP.
<i>Bad Data</i> =	ALCB's usage field (al_usage). This should be 0.
<i>Good Data</i> =	ALCB's action field (al_action).
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 33 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	The maximum combined throughput for the sending direction for an ALCB is incorrect (PSalcb[].al_mct_snd). This field should agree with the data in the ALDB (PSalcb[].al_mct_snd == PSaldb[].l3info.snd_mct).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will set PSalcb[].al_mct_snd to the value of PSaldb[].l3info.snd_mct.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSalcb[].al_mct_snd.
<i>Good Data</i> =	PSaldb[].l3info.snd_mct.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB fields declared before the PSalcb[].lccb_index[] fields.
<i>Dump 2</i> =	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 34 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	When al_action field indicated that a campon is going on (PSXCAMPED) and the value of al_rst_state is either PS_R4_R2OOS or PS_R255_MPTM, the value of al_actvc or the virtual call count is not zero.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field is used to dump the value of al_action field of ALCB.

<i>Bad Data</i> =	ALCB's al_actvc field.
<i>Good Data</i> =	0. The value of al_actvc field should be zero.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 35 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The audit found an ALCB in an R0 (NOTINIT) state and should not have any LCCBs linked to it, but this ALCB has a non-null LCCB connected to it.
Possible Error Effect:	Prints a report and corrects the fields.
Corrective Action Taken:	The audit reports the error and sets the lccb_index[] array field to PSLCCBNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index into the ALCB's lccb_index[] array that is non-null.
<i>Bad Data</i> =	ALCB's LCCB field.
<i>Good Data</i> =	PSLCCBNULL - Valid null value for this field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 36 Error Code: ODD_COUNT

Error Code:	ODD_COUNT
Error Description:	The combined throughput sum for the receiving direction of an ALCB is incorrect (PSalcb[].al_sum_rcv). This sum should be equal to the throughput class for the receiving direction (PSlccb[].vc_tpc[PSCING]) of all LCCBs, linked to this ALCB, that have the maximum combined throughput increment flag set (PSlccb[].mct_incr == DBYES).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will update the PSalcb[].al_sum_rcv to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSalcb[].al_sum_rcv.
<i>Good Data</i> =	The correct value for the PSalcb[].al_sum_rcv field.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB fields declared before the PSalcb[].lccb_index[] fields.
<i>Dump 2</i> =	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 37 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	The audit found an ALCB linked to a LAPB or LAPD LLCB, but the LLCB does not point back to the ALCB.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and return AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's LLCB field (llcb_index).
<i>Bad Data</i> =	ALCB's LLCB's ALCB index. This should be the value of the current ALCB.
<i>Good Data</i> =	ALCB index.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 38 Error Code: OSLNULL

Error Code:	OSLNULL
Error Description:	The audit found an ALCB linked to a valid LLCB, but the LCCB index array's (lccb_index[]) 0 slot is not null. There should not be any LCCB marked in this field.
Possible Error Effect:	Prints a report and corrects the erroneous fields.
Corrective Action Taken:	The audit prints an error report and sets the ALCB's lccb_index[] element to

	null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's lccb_index[] field. This should be null.
<i>Good Data</i> =	PSLCCBNULL - Valid null value for this field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 39 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit found an ALCB on the free linked list with either the next ALCB or previous ALCB out of range.
Possible Error Effect:	Prints a report and recovers the linked list.
Corrective Action Taken:	The audit prints an error report and calls the head cell recovery which nulls the head cell and links up all ALCBs with the owner of this head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSALCBNULL - One more than maximum value the next ALCB and previous ALCB can hold.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	AULNK defined in au/AUps/AUextern.h (all size unsigned short): • "back" struct's owner field, • "back" struct's next field, • "back" element's back field, • current element being audited, • current element's owner, • current element on this linked list, • current element's back field, • data element null value, • head cell's first element, • head cell's last element, • head cell null value, • "next" element's owner field, • "next" element on this linked list, • "next" element's back field.

Table 40 Error Code: OWNER

Error Code:	OWNER
Error Description:	The audit found an ALCB with an out of range LLCB index or "owner" field.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and returns AUFALSE which recovers the ALCB

	to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's LLCB index field (llcb_index). This is treated as the owner field.
<i>Good Data</i> =	PSLLCBNULL - Null value for this LLCB_index field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 41 Error Code: P_UNKQUAL

Error Code:	P_UNKQUAL
Error Description:	The maximum combined throughput for the receiving direction for an ALCB is incorrect (PSalcb[].al_mct_rcv). This field should agree with the data in the ALDB (PSalcb[].al_mct_rcv == PSaldb[].l3info.rcv_mct).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will set PSalcb[].al_mct_rcv to the value of PSaldb[].l3info.rcv_mct.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSalcb[].al_mct_rcv.
<i>Good Data</i> =	PSaldb[].l3info.rcv_mct.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB fields declared before the PSalcb[].lccb_index[] fields.
<i>Dump 2</i> =	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 42 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	The audit has found that the value of al_party found in this ALCB does not match the value of pix found in the ALDB located at the index equal to aldb_index of ALCB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	The al_party value of this ALCB.
<i>Good Data</i> =	The value of pix found in ALDB accessed by aldb_index of this ALCB.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 43 Error Code: PIXBAD

Error Code:	PIXBAD
Error Description:	An ALCB has been found linked to a LAPD LLCB but has a null ALDB linkage (ALCB.aldb_index = PSALDBNULL) and a non null party (ALCB.al_party != FCP_NULL).
Possible Error Effect:	Varies according to conditions (that is, is the al_party field wrong or is the aldb_index wrong?).
Corrective Action Taken:	The link will be torn down and all associated resources (ALCB etc.) will be returned to the appropriate idle list(s).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ALCB in error.
<i>Bad Data</i> =	ALCB's al_party field.
<i>Good Data</i> =	FCP_NULL (The correct value of the ALCB's al_party field).
<i>Logical Index</i> =	The ALCB index.
<i>Dump 1</i> =	The LLCB linked to this ALCB (PSllcb[ALCB.llcbindex]).
<i>Dump 2</i> =	The DCCB associated with this ALCB (PSdccb[ALCB.ph_cn]).
<i>Dump 3</i> =	The ALCB up to but not including its lccb_index[] field.
<i>Dump 4</i> =	The ALCB from just after its lccb_index[] field.

Table 44 Error Code: PIXOUTOFRNG

Error Code:	PIXOUTOFRNG
Error Description:	The audit has found an ALCB whose "al_party" field is out of range (that is not between 0 and 7 inclusive) when we have a valid aldb_index value.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the ALCB in which error was found.
<i>Bad Data =</i>	ALCB's al_party field.
<i>Good Data =</i>	0. The value chosen here is 0. The value can really be any one between 0 and 7 with proper linkage between DCCB, LLCB and ALCB.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 45 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The audit has found an ALCB whose "aldb_index" field is not correct. It failed the point to point back check with ALDB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the ALCB in which error was found.
<i>Bad Data =</i>	ALCB's aldb_index field.
<i>Good Data =</i>	0. The value chosen here is 0 but any value between 0 and 127 which gives proper linkage between DCCB, LLCB and ALCB.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 46 Error Code: QUAR

Error Code:	QUAR
Error Description:	The audit found an ALCB with the quarantine bit set.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB's quarantine bit value (quar_bit).
<i>Good Data =</i>	DBNO - valid value for the quarantine bit indicating no problems.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 47 Error Code: PQMAXCNT

Error Code:	PQMAXCNT
Error Description:	The Multi-Line Hunt Group busy limit parameter of an ALCB is incorrect (PSalcb[].al_busy_limit). This field should agree with the data in the ALDB (PSalcb[].al_busy_limit == PSaldb[].l3info.busy_limit).
Possible Error Effect:	Calls terminated to MLHG members associated with this ALCB could be blocked.
Corrective Action Taken:	The audit will set PSalcb[].al_busy_limit to the value of PSaldb[].l3info.busy_limit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_busy_limit.
<i>Good Data =</i>	PSalcb[].l3info.busy_limit.
<i>Logical Index =</i>	ALCB index.
<i>Dump 1 =</i>	The ALCB fields declared before the PSalcb[].lccb_index[] fields.

<i>Dump 2 =</i>	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 48 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The hunt notify field of an ALCB is incorrect (PSalcb[].al_hnt_nfy). This field should agree with the data in the ALDB (PSalcb[].al_hnt_nfy == PSaldb[].l3info.hnt_nfy).
Possible Error Effect:	Calls terminated to MLHG members could fail.
Corrective Action Taken:	The audit will set the PSalcb[].al_hnt_nfy to the value of PSaldb[].l3info.hnt_nfy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_hnt_nfy.
<i>Good Data =</i>	PSaldb[].l3info.hnt_nfy.
<i>Logical Index =</i>	ALCB index.
<i>Dump 1 =</i>	The ALCB fields declared before the PSalcb[].lccb_index[] fields.
<i>Dump 2 =</i>	The ALCB fields declared after the PSalcb[].lccb_index[] fields.

Table 49 Error Code: SETMAP

Error Code:	SETMAP
Error Description:	PVC map set incorrectly. When audit finds an ALCB with the pvcmap field non-null, the pvcsetup field should be enabled or the action field should be PVCMAP. Audits found a non-null pvcmap field, but the pvcset was not enabled and the action field was not PVCMAP.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit reports the error and sets the ALCB's al_pvcmap BD field to PSNULLOFF.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's al_pvcsetup. This should be 1 if the al_action field is not PVCMAP.
<i>Bad Data =</i>	ALCB's action (al_action) field. This should be PSXPVCMAP if the al_pvcsetup field is 0.
<i>Good Data =</i>	PSNULLOFF - valid null value for the pvcmap field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	ALCB's al_pvcmap. The ALCB's PVC BD field. This should be null if the other fields are not set.

Table 50 Error Code: SLAVE_LINK

Error Code:	SLAVE_LINK
Error Description:	The audit found an ALCB that is linked to a valid LCB, but a LCCB linked to the ALCB is not pointing back to the ALCB. In this case, either the LCCB's lcn field is not the correct index into the ALCB's lccb_index[] array, or the LCCB's ALCB index is not the same value as the current ALCB.
Possible Error Effect:	Prints a report and corrects the ALCB's LCCB field.
Corrective Action Taken:	The audit prints an error report and sets the ALCB's lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's LCCB value.
<i>Bad Data =</i>	ALCB's LCCB logical channel number (lcn) field which should equal the index into the ALCB's lccb_index[] array.
<i>Good Data =</i>	Index into the ALCB's lccb_index[].
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	(unsigned short) The ALCB's LCCB's alcb_index field. This should equal the current ALCB value.

Table 51 Error Code: STATE

Error Code:	STATE
Error Description:	The audit found an ALCB with an invalid state (al_rst_state).

Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB's state (al_rst_state).
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 52 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The audit found an ALCB with a valid action field (al_action) of either CAMPED, REMOVE or L2DOWN but the ALCB's state is not R4.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's action field (al_action).
<i>Bad Data =</i>	ALCB's state (al_rst_state) field.
<i>Good Data =</i>	PS_R4_R2OOS - valid state field for the al_rst_state field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 53 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	When al_campon field indicated that a campon is going on (PSENABLED) and the value of the al_rst_state field is PS_R1_PKTRDY audit found that the value of al_actvc field has zero value. This value should be greater than 0 as there must be some virtual calls up in that situation.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB. The ALCB will be returned to the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field is used to dump the value of al_campon field of ALCB.
<i>Bad Data =</i>	ALCB's al_actvc field.
<i>Good Data =</i>	ALCB's al_rst_state field. This is not a good data but this value is dumped to get more information. Good data is that value of al_actvc is greater than zero.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

Table 54 Error Code: TIMERLINK

Error Code:	TIMERLINK
Error Description:	The audit found an ALCB in an R3 (DCE RESTART) state and should have a timer running, but the audit did not find any timer pending.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Return from Pst3_query which indicates if there is a recorded timer running.
<i>Bad Data =</i>	ALCB's al_tim[] link timer element.
<i>Good Data =</i>	PSNOTIMER - Null value for the Pst3_query return.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	(long) ALCB's al_tim[] MAP timer element (al_tim[PSXT3MAP]).

Table 55 Error Code: TIMERLST

Error Code:	TIMERLST

Error Description:	The audit found an ALCB linked to a LLCB. This ALCB has no timers pending; however, it has data indicating there should be a timer: the al_action is PSXL3CHNG, the al_13changed bit is set, or there is a BD attached to this ALCB as indicated in al_pvcmap.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's 13changed bit (In error if set).
<i>Bad Data</i> =	ALCB's al_action (In error is PSL3CHNG).
<i>Good Data</i> =	ALCB's state, "al_rst_state".
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	ALCB's pvc map information indicating BDs (size PSOFFSET).

Table 56 Error Code: TIX

Error Code:	TIX
Error Description:	The audit has found an ALCB whose "tix" field is out of range.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's tix field.
<i>Good Data</i> =	Value of tix field of the LLCB found using the llcb_index field of this ALCB.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	NA.

Table 57 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The audit found an ALCB in an R2 or R4 (DTE RESTART or R2OOS) transient state and the al_chng_state bit is set. The audit has previously seen this bit set for the ALCB for at least 5 minutes; therefore, the ALCB will be torn down.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's al_action field.
<i>Bad Data</i> =	ALCB's al_rst_state.
<i>Good Data</i> =	PH channel number.
<i>Logical Index</i> =	Logical PH number.
<i>Dump</i> =	NA.

Table 58 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, Audit ID that was running at the time of the interrupt.
<i>Dump</i> =	NA.

Table 59 Error Code: ZAPTRN

Error Code:	ZAPTRN
Error Description:	The audit found an ALCB with the al_notlinked bit set indicating that the ALCB is

	in a transient state of being torn down and the al_chng_state bit is set since the audit has seen the al_notlinked bit set before. Since audits have seen this transient state twice, the ALCB will be torn down.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB's al_chng_state bit that the audit has set before for this ALCB and now sees the transient state again, the ALCB will be torn down.
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	NA.

ALCB-B

Software Release: 5E12 and later

1. ACCESS LINE CONTROL BLOCK (ALCB) AUDIT

The ALCB audit is part of the switch maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The ALCB audit uses the packet switching access line control block array (PSalcb), the logical control block array (PSllcb), and the logical channel control block array (PSlccb).

3. AUDIT CHECKS

The audit makes linkage checks within the PSalcb array with the logical link control block (PSllcb) and the logical channel control block array (PSlccb). It invokes its semantic check function to verify the correctness of data in the PSalcb. For on-demand B-channel (ODB) packet services, the audit will turn on the t3xx timer if there is no packet activity, the timer is not currently running, and the user is a subscriber to the t3xx timer feature.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ACCPTR

Error Code:	ACCPTR
Error Description:	ALCB access pointer is not 0. The audit found an ALCB with access pointer not set to 0.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit reports the error and sets the ALCBs al_sid field to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCBs al_sid field. Access pointer should be 0.
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 2 Error Code: ACPTRBAD

Error Code:	ACPTRBAD
Error Description:	Action pointer bad. The audit found an ALCB with an invalid action field "al_action".
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCBs action (al_action) field.
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 3 Error Code: ACTION

Error Code:	ACTION
Error Description:	An ALCB appears to be in use for modem-pool testing (ALCB.al_action =

	PSXMPTM) but either its state is incorrect (ALCB.al_rst_state != PS_R255_MPTM) or it has more than one virtual call up (ALCB.al_actvc > 1).
Possible Error Effect:	If calls or modem pool tests really are up on this ALCB, they will fail.
Corrective Action Taken:	The audit will report the error and recover the ALCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ALCB (&PSalcb[key given]).
<i>Bad Data</i> =	ALCB's state (ALCB.al_rst_state).
<i>Good Data</i> =	PS_R255_MPTM.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 4 Error Code: ALDBRANGE

Error Code:	ALDBRANGE
Error Description:	The audit has found a LAPD ALCB whose "aldb_index" field is out of range (greater than null).
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's aldb_index field.
<i>Good Data</i> =	0. The value chosen here is 0 but any value between 0 and 128 which gives proper linkage between DCCB, LLCB and ALCB is a good value.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 5 Error Code: ALINK

Error Code:	ALINK
Error Description:	The audit found an ALCB in the R255 modem pool terminal maintenance state which has more than one LCCB attached to it.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Count of the number of LCCB's linked to the ALCB.
<i>Good Data</i> =	ALCB index.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 6 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The virtual call throughput class for the sending direction of an LCCB linked to an ALCB is out of range (PSTPC75 > PSlccb[].vc_tpc[PSCED] > PSTPC19200).
Possible Error Effect:	Calls associated with this ALCB could be blocked when they shouldn't be.
Corrective Action Taken:	The audit will set the quarantine bit in the LCCB. The LCCB audit will run as a related audit and will recover the LCCB to the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSlccb[].vc_tpc[PSCED].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 7 Error Code: BADOWNER

Error Code:	BADOWNER

Error Description:	The audit found an ALCB with the transient state bit "al_notlinked" set. The ALCB has a non-null LLCB index field and should not be attached to any LLCB when this bit is set.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB LLCB value (licb_index).
<i>Good Data =</i>	PSLLCBNULL - Valid null value for this field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The virtual call throughput class for the receiving direction of an LCCB linked to an ALCB is out of range (PSTPC75 > PSlccb[].vc_tpc[PSCING] > PSTPC19200).
Possible Error Effect:	Calls associated with this ALCB could be blocked when they shouldn't be.
Corrective Action Taken:	The audit will set the quarantine bit in the LCCB. The LCCB audit will run as a related audit and will recover the LCCB to the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSlccb[].vc_tpc[PSCING].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The incoming logical channel number count for multi-line hunt group (MLHG) members of an ALCB is incorrect (PSalcb[].al_inlcn_mlhg). This field should be equal to the number of LCCBs, linked to this ALCB, that meet the following conditions: - The LCCB Busy Limit Parameter flag is set (PSlccb[].blp_incr == DBYES). - The LCCB is associated with an incoming call (PSlccb[].aphend == PSCED).
Possible Error Effect:	Multi-line hunt group calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will update the PSalcb[].al_inlcn_mlhg field to agree with the number of LCCBs that meet the conditions above.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_inlcn_mlhg.
<i>Good Data =</i>	The number of LCCBs that meet the conditions above.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit found an ALCB in an inter-SM PH that is linked to a valid LLCB, but there is also a non-null LCCB field in the ALCB data structure. If the ALCB is for an inter-SM PH, there should NOT be any LCCBs linked.

Possible Error Effect:	Prints a report and corrects the erroneous fields.
Corrective Action Taken:	The audit prints an error report and sets the ALCBs lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into the ALCB's lccb_index[].
<i>Bad Data =</i>	ALCB's LCCB value in lccb_index[] array that is non-null.
<i>Good Data =</i>	PSLCCBNULL - Valid null value for this field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 11 Error Code: BDPCR

Error Code:	BDPCR
Error Description:	BD pointer attached to ALCB is corrupted. The audit found an ALCB with the pvcsetup field enabled but there was a null BD field (al_pvcmap). If the pvcsetup field is set, there must be a BD attached.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit reports the error and sets the ALCBs al_pvcsetup BD field to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCBs al_pvcmap. The ALCBs PVC BD field. This field should be non-null when pvcsetup is set.
<i>Good Data =</i>	PSNULLOFF - Valid null value for the pvcmap field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 12 Error Code: CAMPSTATE

Error Code:	CAMPSTATE
Error Description:	When al_campon field indicated that a campon is going on (PSENABLED), audit found invalid value of the al_rst_state field in ALCB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field is used to dump the value of al_campon field of ALCB.
<i>Bad Data =</i>	ALCB's al_rst_state field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 13 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found an ALCB that should have multi-line hunt group information, since the ALDB's huntflag indicates this is a multi-line hunt group channel, but the count of LCCBs linked to this ALCB with their "huntflag" set does not match the value of the ALCB's "al_usage".
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's "al_usage".
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Highest LCN value of ALCB.
<i>Bad Data =</i>	ALCB's count of LCCBs with the huntflag enabled.
<i>Good Data =</i>	Audits count of the total number of LCCB's with the huntflag enabled.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 14 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The audit found an ALCB linked to a valid LLCB, but the ALCB's PH channel

	number doesn't match the LLCB's D-channel number.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit prints an error report and sets the ALCB's PH channel number (al_phcn) to the LLCB's D-channel number (dchan).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's LLCB field (llcb-index).
<i>Bad Data =</i>	ALCB's channel number (al_phcn) the value of the current ALCB. This should equal the LLCB's dchan.
<i>Good Data =</i>	ALCB's LLCB's D-channel number (dchan).
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 15 Error Code: DFLG

Error Code:	DFLG
Error Description:	MLHG flags should be DISABLED The audit found an ALCB with PVC channels that have their LCCB's huntflag enabled and should be null. PVC's cannot be of the huntable type.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the ALCB's LCCB's huntflag to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into the ALCB's lccb_index[] array.
<i>Bad Data =</i>	ALCB's LCCB (indicated by the lccb_index[] array mentioned in error address) whose huntflag is enabled.
<i>Good Data =</i>	PSDISABLE - Valid null value that the LCCB's huntflag should have.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 16 Error Code: DISP

Error Code:	DISP
Error Description:	Should be a disabled pointer. The audit found an ALCB with the action field "al_action" set to NULL action, but either the 13changed, pvcsetup or the pvcmap fields is not disabled or null. None of these fields can be set when in the NULL action condition.
Possible Error Effect:	Prints a report and corrects the ALCB fields.
Corrective Action Taken:	The audit reports the error and sets the "al_13changed" and "al_pvcsetup" fields to Disabled and nulls the pvcmap BD field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's 13changed field (al_13changed). This should be PSDISABLED.
<i>Bad Data =</i>	ALCB's pvcsetup (al_pvcsetup) field. This should be PSDISABLED.
<i>Good Data =</i>	PSDISABLED - valid null field for the al_13changed and al_pvcsetup fields.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 17 Error Code: EVEN_COUNT

Error Code:	EVEN_COUNT
Error Description:	The combined throughput sum for the sending direction of an ALCB is incorrect (PSalcb[].al_sum_snd). This sum should be equal to the throughput class for the sending direction (PSlccb[].vc_tpc[PSCED]) of all LCCBs, linked to this ALCB, that have the maximum combined throughput increment flag set (PSlccb[].mct_incr == DBYES).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will update the PSalcb[].al_sum_snd to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_sum_snd.
<i>Good Data =</i>	The correct value for the PSalcb[].al_sum_snd field.

<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 18 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Multi-line hunt group (MLHG). The audit found an ALCB that should have multi-line hunt group information, since the ALDB's huntflag indicates this is a multi-line hunt group channel, but the information does not match what is in the ALDB, or the al_usage field is not set to indicate that this is a multi-line hunt group channel. The audit totals the number of ALDB LCN-in and the 2-way LCNs (which is the count of huntable LCNs) and compares it with the ALCB's "al_lcnmlhg" field finding them inconsistent.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's al_lcnmlhg.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's count of MLHG LCN's (al_lcnmlhg).
<i>Good Data</i> =	Audits count of the total of 2way plus the lcn-in.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 19 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	PVC group type number. The audit found an ALCB with an invalid value for the highest PVC lcn value. When comparing the ALCB's "al_hipvc" against the ALDB's (access line data block write protected area) pvc_h_lcn, which is the highest LCN value for a PVC, the two values did not match.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the PVC high logical channel number (al_hipvc) field to that of the DCDB write protected data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's highest PVC LCN number (al_pvclcn).
<i>Good Data</i> =	ALDB's highest LCN value for a PVC.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB.

Table 20 Error Code: HIM_NUM

Error Code:	HIM_NUM
Error Description:	The audit found an ALCB with an invalid high logical channel number (al_hilcn) field. This should match the first non-null value in the write protected data access line data block (ALDB) when looking at the h_lcn_out fields down to pvc_h_lcn.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the ALCB high LCN (al_hilcn) field to that of the ALDB write protected data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's high LCN number (al_hilcn).
<i>Good Data</i> =	ALDB's highest value for an LCN.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB.

Table 21 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The audit has found a LAPB ALCB whose "aldb_index" field is out of range

	(either null or greater than null).
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's aldb_index field.
<i>Good Data</i> =	0. The value chosen here is 0 but any value between 0 and 127 which gives proper linkage between DCCB, LLCB and ALCB is good value.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 22 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	LCCB linked to ALCB when it should be idle. The audit found an ALCB with a non-null LCCB value in a lccb_index[] array slot which is a slot value greater than the highest logical channel number (LCN) value that the write protected data block (ALDB) indicates is valid for this channel. Basically, all slot values greater than what the ALDB indicates is the highest LCN value, must be null and the audit found a non-null value.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's lccb_index[] element.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index into the ALCB's lccb_index[] array element.
<i>Bad Data</i> =	ALCB's LCCB value in the lccb_index[] array element.
<i>Good Data</i> =	ALCB's highest LCN field - al_hilcn.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 23 Error Code: INDEX

Error Code:	INDEX
Error Description:	The audit found an ALCB that is linked to a valid LLCB, but a LCCB field is out of range.
Possible Error Effect:	Prints a report and corrects the ALCB's LCCB field.
Corrective Action Taken:	The audit prints an error report and set the ALCB's lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's index into the lccb_index[] array.
<i>Bad Data</i> =	ALCB's LCCB value that is out of range.
<i>Good Data</i> =	PSHMAXLCCB - One more than the maximum value the LCCB field can hold.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 24 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	When al_action field indicated that a campon is going on, audit found invalid value of the al_rst_state field in ALCB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field is used to dump the value of al_action field of ALCB.
<i>Bad Data</i> =	ALCB's al_rst_state field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 25 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The audit found an ALCB on the free linked list with a non-null LCCB index field. The free linked list should not have any LCCBs attached to the ALCB.
Possible Error Effect:	Prints a report and corrects the erroneous field.
Corrective Action Taken:	The audit prints an error report and null, the LCCB field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Element number of the lccb_index array that is non-null.
<i>Bad Data =</i>	ALCB's LCCB value.
<i>Good Data =</i>	PSLCCBNULL - Valid null value for this field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 26 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The audit found an ALCB with the al_notlinked bit set indicating that the ALCB is in a transient state of being torn down but the state is not R4. This is the only valid state for an ALCB when this bit is set.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	al_phcn field of ALCB PH channel number.
<i>Bad Data =</i>	ALCB's state (al_rst_state). This should be R4.
<i>Good Data =</i>	al_action field of ALCB.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 27 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The audit found an ALCB on the free linked list and either there is a timer tag record in the al_tim array or the PVC map field indicating a buffer descriptor (BD) is not null.
Possible Error Effect:	Prints a report and corrects the fields.
Corrective Action Taken:	The audit reports the error and sets the al_tim timer array fields and the pvcmap field to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's al_pvcmap (BD) field. This should be PSNULLOFF.
<i>Bad Data =</i>	Return value from the PS timer check function, used for debugging.
<i>Good Data =</i>	PSNULLOFF - Valid null value for the al_pvcmap field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB

Table 28 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Number of links 2 active VCs. The audit found an ALCB with an invalid value for the number of active virtual calls up on this ALCB (al_actvc). Audits have computed the number of non-null LCCBs on this ALCB and this value does not equal "al_actvc".
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the number of active VCs (al_actvc) to that which the audit found as non-null LCCBs linked to this ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB's number of active VCs (al_actvc).
<i>Good Data =</i>	Audits count of non-null LCCBs attached to ALCB.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 29 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The audit found an ALCB linked to a valid LLCB, but the ALCB's forward or back ALCB fields are not null. When the ALCB is linked to a LLCB, it cannot be linked to other ALCB's too.
Possible Error Effect:	Print a report and correct the erroneous fields.
Corrective Action Taken:	The audit will print an error report and set the ALCB's forward and back ALCB fields (fdalb_ind and bkalcb_ind) to null values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's back ALCB field (bkalcb_ind).
<i>Bad Data =</i>	ALCB's forward ALCB field (fdalcb_ind).
<i>Good Data =</i>	PSALCBNULL - Valid null value for these fields.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 30 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The audit found an ALCB with a valid "establish" action field (al_action) but the ALCB's state was not R0 or R3.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit report the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PH channel number.
<i>Bad Data =</i>	ALCB's state (al_rst_state) field.
<i>Good Data =</i>	ALCB's action (al_action) field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 31 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit found a member error in the linked list of the specified audit.
Possible Error Effect:	Prints an error report and corrects the linked list.
Corrective Action Taken:	Returns AUFALSE to calling function which rebuilds the linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Linked list null value.
<i>Logical Index =</i>	AUcount, the current element number.
<i>Dump =</i>	AULNK defined in au/AUups/AUextern.h (all size unsigned short): • "back" element's back field, • "back" struct's owner field, • "back" struct's next field, • current element being audited, • current element on this linked list, • current element's back field, • current element's owner, • data element null value,

	• head cell null value, • head cell's first element, • head cell's last element, • "next" element on this linked list, • "next" element's back field, • "next" element's owner field.

Table 32 Error Code: MLNKGRP

Error Code:	MLNKGRP
Error Description:	MLHG link group error in the ALCB. The audit found an ALCB with non-null Multi-line hunt group MLHG (al_lcnmlhg) or usage information of the MLHG (al_usage) when the access line data block (ALDB) write protected data indicates that no MLHG channels can be on this ALCB.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and nulls the ALCB's al_lcnmlhg and al_usage fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's multi-line hunt group flag (al_lcnmlhg). This should be PSNOHUNT_GRP.
<i>Bad Data =</i>	ALCB's usage field (al_usage). This should be 0.
<i>Good Data =</i>	ALCB's action field (al_action).
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 33 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	The maximum combined throughput for the sending direction for an ALCB is incorrect (PSalcb[].al_mct_snd). This field should agree with the data in the ALDB (PSalcb[].al_mct_snd == PSaldb[].l3info.snd_mct).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will set PSalcb[].al_mct_snd to the value of PSaldb[].l3info.snd_mct.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_mct_snd.
<i>Good Data =</i>	PSaldb[].l3info.snd_mct.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 34 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	When al_action field indicated that a campon is going on (PSXCAMPED) and the value of al_rst_state is either PS_R4_R2OOS or PS_R255_MPTM, the value of al_actvc or the virtual call count is not zero.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field is used to dump the value of al_action field of ALCB.
<i>Bad Data =</i>	ALCB's al_actvc field.
<i>Good Data =</i>	0. The value of al_actvc field should be zero.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 35 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The audit found an ALCB in an R0 (NOTINIT) state and should not have any LCCBs linked to it, but this ALCB has a non-null LCCB connected to it.
Possible Error Effect:	Prints a report and corrects the fields.
Corrective Action Taken:	The audit reports the error and sets the lccb_index[] array field to PSLCCBNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index into the ALCB's lccb_index[] array that is non-null.
<i>Bad Data</i> =	ALCB's LCCB field.
<i>Good Data</i> =	PSLCCBNULL - Valid null value for this field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 36 Error Code: ODD_COUNT

Error Code:	ODD_COUNT
Error Description:	The combined throughput sum for the receiving direction of an ALCB is incorrect (PSalcb[].al_sum_rcv). This sum should be equal to the throughput class for the receiving direction (PSlccb[].vc_tpc[PSCING]) of all LCCBs, linked to this ALCB, that have the maximum combined throughput increment flag set (PSlccb[].mct_incr == DBYES).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will update the PSalcb[].al_sum_rcv to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSalcb[].al_sum_rcv.
<i>Good Data</i> =	The correct value for the PSalcb[].al_sum_rcv field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 37 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	The audit found an ALCB linked to a LAPB or LAPD LLCB, but the LLCB does not point back to the ALCB.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and return AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's LLCB field (llcb_index).
<i>Bad Data</i> =	ALCB's LLCB's ALCB index. This should be the value of the current ALCB.
<i>Good Data</i> =	ALCB index.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 38 Error Code: OSLNULL

Error Code:	OSLNULL
Error Description:	The audit found an ALCB linked to a valid LLCB, but the LCCB index array's (lccb_index[]) 0 slot is not null. There should not be any LCCB marked in this field.
Possible Error Effect:	Prints a report and corrects the erroneous fields.
Corrective Action Taken:	The audit prints an error report and sets the ALCB's lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's lccb_index[] field. This should be null.
<i>Good Data</i> =	PSLCCBNULL - Valid null value for this field.

<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 39 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit found an ALCB on the free linked list with either the next ALCB or previous ALCB out of range.
Possible Error Effect:	Prints a report and recovers the linked list.
Corrective Action Taken:	The audit prints an error report and calls the head cell recovery which nulls the head cell and links up all ALCBs with the owner of this head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSALCBNULL - One more than maximum value the next ALCB and previous ALCB can hold.
<i>Logical Index</i> =	ALCB index.
<i>Dump 1</i> =	The ALCB.
<i>Dump 2</i> =	AULNK defined in au/AUps/AUextern.h (all size unsigned short):
	• "back" struct.'s owner field, • "back" struct.'s next field, • "back" element's back field, • current element being audited, • current element's owner, • current element on this linked list, • current element's back field, • data element null value, • head cell's first element, • head cell's last element, • head cell null value, • "next" element's owner field, • "next" element on this linked list, • "next" element's back field.

Table 40 Error Code: OWNER

Error Code:	OWNER
Error Description:	The audit found an ALCB with an out of range LLCB index or "owner" field.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit prints an error report and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's LLCB index field (llcb_index). This is treated as the owner field.

<i>Good Data</i> =	PSLLCBNULL - Null value for this LLCB_index field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 41 Error Code: P_UNKQUAL

Error Code:	P_UNKQUAL
Error Description:	The maximum combined throughput for the receiving direction for an ALCB is incorrect (PSalcb[].al_mct_rcv). This field should agree with the data in the ALDB (PSalcb[].al_mct_rcv == PSaldb[].l3info.rcv_mct).
Possible Error Effect:	Calls could be blocked or calls could be allowed when they shouldn't be.
Corrective Action Taken:	The audit will set PSalcb[].al_mct_rcv to the value of PSaldb[].l3info.rcv_mct.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSalcb[].al_mct_rcv.
<i>Good Data</i> =	PSaldb[].l3info.rcv_mct.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 42 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	The audit has found that the value of al_party found in this ALCB does not match the value of pix found in the ALDB located at the index equal to aldb_index of ALCB.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	The al_party value of this ALCB.
<i>Good Data</i> =	The value of pix found in ALDB accessed by aldb_index of this ALCB.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 43 Error Code: PIXBAD

Error Code:	PIXBAD
Error Description:	An ALCB has been found linked to a LAPD LLCB but has a null ALDB linkage (ALCB.aldb_index = PSALDBNULL) and a non null party (ALCB.al_party != FCP_NULL).
Possible Error Effect:	Varies according to conditions (that is, is the al_party field wrong or is the aldb_index wrong?).
Corrective Action Taken:	The link will be torn down and all associated resources (ALCB etc.) will be returned to the appropriate idle list(s).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ALCB in error.
<i>Bad Data</i> =	ALCB's al_party field.
<i>Good Data</i> =	FCP_NULL (The correct value of the ALCB's al_party field).
<i>Logical Index</i> =	The ALCB index.
<i>Dump 1</i> =	The LLCB linked to this ALCB (PSllcb[ALCB.llcbindex]).
<i>Dump 2</i> =	The DCCB associated with this ALCB (PSdccb[ALCB.ph_cn]).
<i>Dump 3</i> =	The ALCB.

Table 44 Error Code: PIXOUTOFRNG

Error Code:	PIXOUTOFRNG
Error Description:	The audit has found an ALCB whose "al_party" field is out of range (that is not between 0 and 7 inclusive) when we have a valid aldb_index value.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between

	DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's al_party field.
<i>Good Data</i> =	0. The value chosen here is 0. The value can really be any one between 0 and 7 with proper linkage between DCCB, LLCB and ALCB.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 45 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The audit has found an ALCB whose "aldb_index" field is not correct. It failed the point to point back check with access line data block (ALDB).
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's aldb_index field.
<i>Good Data</i> =	0. The value chosen here is 0 but any value between 0 and 127 which gives proper linkage between DCCB, LLCB and ALCB.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 46 Error Code: QUAR

Error Code:	QUAR
Error Description:	The audit found an ALCB with the quarantine bit set.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's quarantine bit value (quar_bit).
<i>Good Data</i> =	DBNO - valid value for the quarantine bit indicating no problems.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 47 Error Code: PQMAXCNT

Error Code:	PQMAXCNT
Error Description:	The multi-line hunt group busy limit parameter of an ALCB is incorrect (PSalcb[].al_busy_limit). This field should agree with the data in the ALDB (PSalcb[].al_busy_limit == PSaldb[].l3info.busy_limit).
Possible Error Effect:	Calls terminated to multi-line hunt group members associated with this ALCB could be blocked.
Corrective Action Taken:	The audit will set PSalcb[].al_busy_limit to the value of PSaldb[].l3info.busy_limit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data</i> =	PSalcb[].al_busy_limit.
<i>Good Data</i> =	PSalcb[].l3info.busy_limit.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 48 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The hunt notify field of an ALCB is incorrect (PSalcb[].al_hnt_nfy). This field should agree with the data in the ALDB (PSalcb[].al_hnt_nfy ==

	PSaldb[].l3info.hnt_nfy).
Possible Error Effect:	Calls terminated to multi-line hunt group members could fail.
Corrective Action Taken:	The audit will set the PSalcb[].al_hnt_nfy to the value of PSaldb[].l3info.hnt_nfy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel number of the ALCB (PSalcb[].al_phcn).
<i>Bad Data =</i>	PSalcb[].al_hnt_nfy.
<i>Good Data =</i>	PSaldb[].l3info.hnt_nfy.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 49 Error Code: SETMAP

Error Code:	SETMAP
Error Description:	PVC map set incorrectly. When audit finds an ALCB with the pvcmap field non-null, the pvcsetup field should be enabled or the action field should be PVCMAP. Audits found a non-null pvcmap field, but the pvcset was not enabled and the action field was not PVCMAP.
Possible Error Effect:	Prints a report and corrects the ALCB field.
Corrective Action Taken:	The audit reports the error and sets the ALCB's al_pvcmap BD field to PSNULLOFF.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's al_pvcsetup. This should be 1 if the al_action field is not PVCMAP.
<i>Bad Data =</i>	ALCB's action (al_action) field. This should be PSXPVCMAP if the al_pvcsetup field is 0.
<i>Good Data =</i>	PSNULLOFF - valid null value for the pvcmap field.
<i>Logical Index =</i>	ALCB index.
<i>Dump =</i>	The ALCB.

Table 50 Error Code: SLAVE_LINK

Error Code:	SLAVE LINK
Error Description:	The audit found an ALCB that is linked to a valid LCB, but a LCCB linked to the ALCB is not pointing back to the ALCB. In this case, either the LCCB's lcn field is not the correct index into the ALCB's lccb_index[] array, or the LCCB's ALCB index is not the same value as the current ALCB.
Possible Error Effect:	Prints a report and corrects the ALCB's LCCB field.
Corrective Action Taken:	The audit prints an error report and sets the ALCB's lccb_index[] element to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	ALCB's LCCB value.
<i>Bad Data =</i>	ALCB's LCCB logical channel number (lcn) field which should equal the index into the ALCB's lccb_index[] array.
<i>Good Data =</i>	Index into the ALCB's lccb_index[].
<i>Logical Index =</i>	ALCB index.
<i>Dump 1 =</i>	The ALCB
<i>Dump 2 =</i>	(unsigned short) The ALCB's LCCB's alcb_index field. This should equal the current ALCB value.

Table 51 Error Code: STATE

Error Code:	STATE
Error Description:	The audit found an ALCB with an invalid state (al_rst_state).
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	ALCB's state (al_rst_state).
<i>Good Data =</i>	ALCB index.
<i>Logical Index =</i>	ALCB index.

<i>Dump</i> =	The ALCB.

Table 52 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The audit found an ALCB with a valid action field (al_action) of either CAMPED, REMOVE or L2DOWN but the ALCB's state is not R4.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's action field (al_action).
<i>Bad Data</i> =	ALCB's state (al_rst_state) field.
<i>Good Data</i> =	PS_R4_R2OOS - valid state field for the al_rst_state field.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 53 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	When al_campon field indicated that a campon is going on (PSENABLED) and the value of the al_rst_state field is PS_R1_PKTRDY audit found that the value of al_actvc field has zero value. This value should be greater than 0 as there must be some virtual calls up in that situation.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB. The ALCB will be returned to the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field is used to dump the value of al_campon field of ALCB.
<i>Bad Data</i> =	ALCB's al_actvc field.
<i>Good Data</i> =	ALCB's al_rst_state field. This is not a good data but this value is dumped to get more information. Good data is that value of al_actvc is greater than zero.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 54 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The audit found an ALCB in an R3 (DCE RESTART) state and should have a timer running, but the audit did not find any timer pending.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Return from PSt3_query which indicates if there is a recorded timer running.
<i>Bad Data</i> =	ALCB's al_timf link timer element.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 55 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The audit found an ALCB linked to a LLCB. This ALCB has no timers pending however, it has data indicating there should be a timer: the al_action is PSXL3CHNG, the al_13changed bit is set, or there is a BD attached to this ALCB as indicated in al_pvcmap.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	ALCB's 13changed bit (in error if set).
<i>Bad Data</i> =	ALCB's al_action (in error is PSL3CHNG).
<i>Good Data</i> =	ALCB's state, "al_rst_state".
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 56 Error Code: TIX

Error Code:	TIX
Error Description:	The audit has found an ALCB whose "tix" field is out of range.
Possible Error Effect:	Print out a report and return false.
Corrective Action Taken:	The audit will report the error and return false which will recover the link between DCCB, LLCB and ALCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the ALCB in which error was found.
<i>Bad Data</i> =	ALCB's tix field.
<i>Good Data</i> =	Value of tix field of the LLCB found using the llcb_index field of this ALCB.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

Table 57 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The audit found an ALCB in an R2 or R4 (DTE RESTART or R2OOS) transient state and the al_chng_state bit is set. The audit has previously seen this bit set for the ALCB for at least 5 minutes therefore, the ALCB will be torn down.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	ALCB's al_action field.
<i>Bad Data</i> =	ALCB's al_rst_state.
<i>Good Data</i> =	PH channel number.
<i>Logical Index</i> =	Logical PH number.
<i>Dump</i> =	The ALCB.

Table 58 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, audit ID that was running at the time of the interrupt.
<i>Dump</i> =	NA.

Table 59 Error Code: ZAPTRN

Error Code:	ZAPTRN
Error Description:	The audit found an ALCB with the al_notlinked bit set indicating that the ALCB is in a transient state of being torn down and the al_chng_state bit is set since the audit has seen the al_notlinked bit set before. Since audits have seen this transient state twice, the ALCB will be torn down.
Possible Error Effect:	Prints a report and recovers the ALCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the ALCB to the free linked list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	ALCB's al_chng_state bit that the audit has set before for this ALCB and now sees the transient state again, the ALCB will be torn down.
<i>Good Data</i> =	ALCB index.
<i>Logical Index</i> =	ALCB index.
<i>Dump</i> =	The ALCB.

ALDB

Software Release: 5E10 and later

1. ACCESS LINE DATA BLOCK (ALDB) AUDIT

The ALDB is a write protected data and is used to hold layer 3 related data of a packet user. The ALDB is assigned at the initialization time for each packet user and to any new packet user that is added to the channel. The audit checks the ALDB when it is either on the free list or is attached with the D-channel data block (DCDB) for a packet user. It makes only linkage checks. No semantic checks are made because the ALDB is a write protected data. This is because the write protection is removed temporarily when ALDB is assigned to a packet user after initialization, and during which the linkages can be corrupted. The same thing occurs when a packet user is removed from a channel and ALDB is put back to the free list.

2. DATA STRUCTURES

PSALDB.

3. AUDIT CHECKS

The ALDB audit performs audits of the ALDB free list, and the linkage between ALDB and the D-channel data block (DCDB). For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

If the audit finds an error in the free list linkages, it constructs the free list again. If the audit finds errors in the linkages with the DCDB, it clears the linkages, puts the ALDB involved back to free list, and calls for routine port conditioning.

The audit produces the following error code(s).

Table 1 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	We found the value of channel not in the valid range or not equal to the null value (free list owner) for this ALDB.
Possible Error Effect:	The channel field of this ALDB has bad value. This could either be because of wild write or because of the channel field was not cleared/set properly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the ALDB will be put back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The starting address of the ALDB in which we found this error.
<i>Bad Data =</i>	The channel value of this ALDB.
<i>Good Data =</i>	0. The value chosen here is zero but any value between 0-127 and null value is good data.
<i>Logical Index =</i>	ALDB index.
<i>Dump =</i>	NA.

Table 2 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	For any ALDB attached to DCDB (that is, it is assigned for packet switching services), the value of fdaldb_ind and bkaldb_ind should be null. But we found either or both of the values to be not null.
Possible Error Effect:	The fdaldb_ind or bkaldb_ind (or both) fields(s) of this ALDB has (have) bad value. This could either be because of wild write or because the field(s) was (were) not set/cleared properly.

Corrective Action Taken:	The audit will report the error and will correct both fields to be null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Normally the starting address of the ALDB in which we found this error is printed here. But here this field is used to dump one of the two fields, fdaldb_ind, that could be bad.
<i>Bad Data =</i>	The other field of the ALDB, bkald_b_ind that could be bad.
<i>Good Data =</i>	PSALDBNULL, null value of ALDB.
<i>Logical Index =</i>	ALDB index.
<i>Dump =</i>	NA.

Table 3 Error Code: NIDLE

Error Code:	NIDLE
Error Description:	The ALDB being audited is on the free list (channel field equal to PSMAXCHN) but the pix (party member number) field of this ALDB has a non-null value (not equal to FCP_NULL).
Possible Error Effect:	If the ALDB is in use and is incorrectly marked as being on the free list, we may lose any call or data activity on this channel.
Corrective Action Taken:	The audit will report the error, will recover the PSrldbhd headcell (type PSRHEADCELL) and will reconstruct the PSALDB free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The starting address of the ALDB in which we found this error.
<i>Bad Data =</i>	The pix field of the ALDB being audited.
<i>Good Data =</i>	The null value of pix field, FCP_NULL.
<i>Logical Index =</i>	ALDB index (aldbind) of the ALDB being audited.
<i>Dump =</i>	NA.

Table 4 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of either fdaldb_ind or bkald_b_ind or both have bad value for this ALDB which is on the ALDB free list.
Possible Error Effect:	We have an ALDB which is not linked properly to the free list. This could either be because of wild write or because both fields were not cleared properly.
Corrective Action Taken:	The audit will report the error and will call the recovery function to recover the ALDB free link list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Starting address of the ALDB in which we found this error.
<i>Bad Data =</i>	0. The value chosen here is zero but the bad data could be any or both of the fdaldb_ind or bkald_b_ind fields of this ALDB. Both these fields are dumped in the AULNK structure which is dumped after this error message in the form of AULnk.cur_next and AULnk.cur_back respectively.
<i>Good Data =</i>	PSALDBNULL. The value chosen here is null but any value between 0-127 that gives a valid link to ALDB free list is good.
<i>Logical Index =</i>	ALDB index.
<i>Dump =</i>	AULnk structure.

Table 5 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	When we use the channel field (of ALDB) to go to DCDB and pix field (of ALDB) to go to the index equal to the pix field in p2aldb array, the value of aldb index found there does not match the aldb index of the ALDB being audited.
Possible Error Effect:	The channel or pix (or both) field(s) of this ALDB has (have) bad value. This could either be because of wild write or because of the field(s) was not cleared/set properly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the ALDB will be put back to free list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The starting address of the ALDB in which we found this error.
<i>Bad Data</i> =	The pix value of this ALDB. Although the value of pix field is dumped here as bad value, it could be that the channel field had bad value or both have bad values. The value of channel field is dumped in "good" data.
<i>Good Data</i> =	For convenience the value dumped here is the value of the channel. But good data is the combination of channel and pix values which give good link between ALDB and DCDB.
<i>Logical Index</i> =	ALDB index.
<i>Dump</i> =	NA.

Table 6 Error Code: PIXBAD

Error Code:	PIXBAD
Error Description:	We found the value of pix is not in the valid range (less than FCP_NULL) for this ALDB.
Possible Error Effect:	The pix field of this ALDB has bad value. This could either be because of wild write or because of the channel field was not cleared/set properly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the ALDB will be put back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The starting address of the ALDB in which we found this error.
<i>Bad Data</i> =	The pix value of this ALDB.
<i>Good Data</i> =	0. The value chosen here is zero but any value between 0-7 is good data.
<i>Logical Index</i> =	ALDB index.
<i>Dump</i> =	NA.

ALMRDT

Software Release: 5E12 and later

1. ALARM REMOTE DIGITAL TERMINAL (ALMRDT) AUDIT

The TR-303 interface on digital network unit-SONET (DNU-S) feature provides the capability to assign a TR-303 remote digital terminal (RDT) to digital signal level 1 (DS1) facilities terminating on a DNU-S. The data related to TR-303 RDT system alarm levels and RT alarm levels is stored in the RLALM_RDT relation. Data in this relation is audited by the ALMRDT audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLALM_RDT relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The general audit error handling mechanism is to print the error code, consider the rIALM_RDT tuple extra, and delete the tuple.

The audit produces the following error code(s).

Table 1 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit could not find an rICKTDATA tuple in the database with the TR303 RDT circuit name (rname).
Possible Error Effect:	The TR303 RDT will not be usable by call processing or maintenance.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should investigate the missing static rICKTDATA tuple in the database.
Dump Description:	
<i>Error Address =</i>	The address of the rIALM_RDT tuple.
<i>Bad Data =</i>	The return from the database read function. 0 means a successful read, any other value implies a failure in the database read.
<i>Good Data =</i>	The TR303 RDT circuit name (rname). This is the key used in reading the missing rICKTDATA tuple.
<i>Logical Index =</i>	The TR303 RDT unit number (rdt_unit). This is the key used in reading the rIALM_RDT tuple.
<i>Dump 1 =</i>	The contents of the rIALM_RDT tuple.
<i>Dump 2 =</i>	The contents of the rUNIT2RDT tuple.

AML

Software Release: 5E10 and later

1. AUTOMATIC MAINTENANCE LIMIT (AML) AUDIT

An automatic maintenance limit defines the maximum number of trunks in a trunk group allowed to be out-of-service at one time during automatic testing. The automatic maintenance limit prevents too many two-way and outgoing trunks in a trunk group from being removed from service. The data keeping track of which trunks are out-of-service is stored in the switching modules. The data is audited by the automatic maintenance limit (AML) audit. The AML audit is part of the terminal maintenance audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The AML audit checks three dynamic data structures in the switching modules: the terminal maintenance automatic maintenance limit array (TMaml), the pending history relation (RLPHISTORY), and the port linkage area relation (RLPORTLA). The TMaml array contains five fields: the out-of-service count of the number of trunks within the trunk group that are out-of-service and have an automatic maintenance limit status (oos_cnt), a check count field for the out-of-service AML count (chk_cnt), the out-of-service count of the number of trunks within the trunk group that are out-of-service (tgoos_cnt), a check count field for the out-of-service trunks count (tgooschk_cnt), and a transient state audit flag (aud). The TMaml array is indexed by trunk group number. The field the audit checks in RLPORTLA is the port status field; in RLPHISTORY, it checks the pending status of trunks in the current SM. The AML audit also uses the feature data for trunks relation (RLFC_TRUNK) to determine which trunk groups should be audited. Figure 1 shows these data structures.

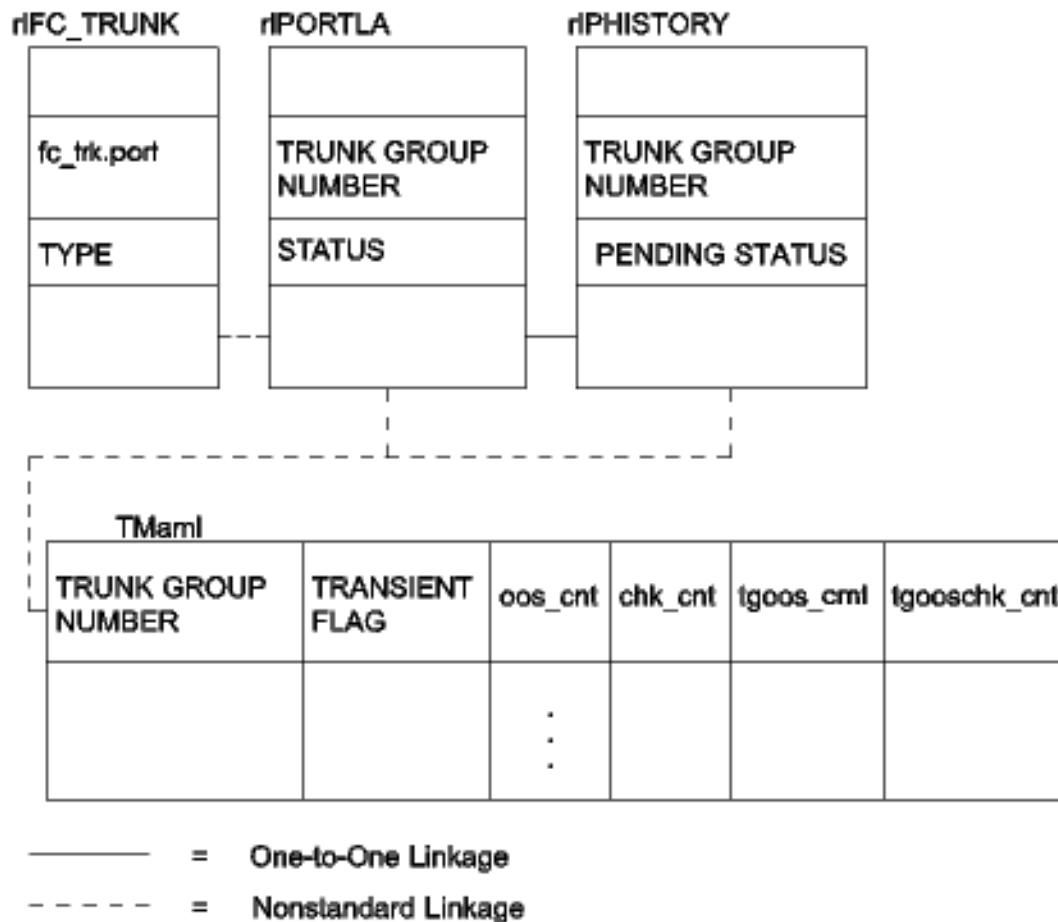


Figure 1 (AML-1) Data Structures of the AML Audit

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) to read RLFC_TRUNK to obtain two-way and outgoing trunk groups.

- (1) If the trunk group is two-way or outgoing, the audit reads RLPORTLA using the fc_trk.port field in RLFC_TRUNK as the key and obtains the port status.
- (2) If the port status is counted toward the automatic maintenance limit, the chk_cnt field of the TMaml array is incremented.
- (3) If the status is not an automatic maintenance limit status, the rPHISTORY tuple for the trunk is read.
- (4) If there is a pending status for the trunk in rPHISTORY and the pending status applies toward the automatic maintenance limit count, the chk_cnt field for the appropriate trunk group in TMaml is incremented.

Once all the tuples have been read, the audit compares the chk_cnt field in TMaml with the oos_cnt field, which is set by the Port Status Administrator program (PSA). Whether the audit reports an error if the counts do not match depends on the transient state audit flag bit. If the bit is set, the audit may be detecting a transient state. Either terminal maintenance (TM) or the AML audit sets the bit whenever a transient state occurs or if the audit is unable to verify the port status. If the bit is not set, the audit assumes the data is correct and reports the inconsistency with an audit error. The audit initiates error recovery action.

4. ERROR HANDLING

The audit sets the out-of-service count in the TMaml array equal to the chk_cnt field and sends a message to the port status administrator in the AM to have the AM automatic maintenance limit status table updated. See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	The AML audit could not read the corresponding rPORTLA tuple using the port.member and port.module field from the rFC_TRUNK tuple.
Possible Error Effect:	OOS counts can not be verified for this trunk group.
Corrective Action Taken:	Report the error and clear the AML audit flag so that OOS errors will not be reported. PORTLA audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMaml array tuple with the TG OOS counts - & TMaml[tg]
<i>Bad Data</i> =	Port member which could not be found in FC_PORTLA.
<i>Good Data</i> =	Trunk group type
<i>Logical Index</i> =	Trunk group with error
<i>Dump</i> =	NA.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The AML audit could not read the corresponding rFC_PORTTYP tuple using the port.member and port.module field from the rFC_TRUNK tuple. The rFC_PORTTYP key for both port.member and port.module should be the same as the port in the rFC_TRUNK tuple.
Possible Error Effect:	OOS counts can not be verified for this trunk group.
Corrective Action Taken:	Report the error and clear the AML audit flag so that OOS errors will not be reported.
Manual Action Required:	Office personnel should determine if the port printed in the audit report as <i>Bad Data</i> is valid. Then insert the valid tuple in FC_PORTTYP using the <i>Bad Data</i> as the member field and SM # as the module field. If the port is not valid, the entry should be removed from FC_TRUNK relation for the trunk group reported in the audit report as <i>Logical Key</i> .
Dump Description:	
<i>Error Address</i> =	Address of the TMaml array tuple with the TG OOS counts - & TMaml[tg].
<i>Bad Data</i> =	Port member which could not be found in FC_PORTTYP.
<i>Good Data</i> =	Trunk group type.
<i>Logical Index</i> =	Trunk group with error.
<i>Dump</i> =	NA.

Table 3 Error Code: OOSCNT

Error Code:	OOSCNT
Error Description:	The oos_cnt in TMaml[] did not match the chk_cnt for the number of AML OOS trunks.
Possible Error Effect:	Incorrect AML count for the OOS AML trunks used by maintenance for automatic removes.
Corrective Action Taken:	Make oos_cnt equal to chk_cnt and send message to PSA in AM in order for the AM status table to be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMaml array tuple with the wrong Out-of-service count - & TMaml[tg].
<i>Bad Data</i> =	Out-of-service count which was determined by PSA TMaml[tg].oos_cnt.
<i>Good Data</i> =	Check count for the Trunk Group which was determined by the audit -

	TMaml[tg].chk cnt.
<i>Logical Index</i> =	Trunk Group in error.
<i>Dump</i> =	NA.

Table 4 Error Code: TGOOSCNT

Error Code:	TGOOSCNT
Error Description:	The tgoos_cnt in TMaml[] did not match the tgooschk_cnt for the number of OOS trunks.
Possible Error Effect:	Incorrect OOS count for the trunks used by maintenance for automatic removes.
Corrective Action Taken:	Make tgoos_cnt equal to tgooschk_cnt and send message to PSA in AM in order for the AM status table to be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMaml array tuple with the wrong TG Out-of-service count - & TMaml[tg].
<i>Bad Data</i> =	Out-of-service count which was determined by PSA TMaml[tg].tgoos_cnt.
<i>Good Data</i> =	Check count for the Trunk group which was determined by the audit - TMaml[tg].tgooschk cnt.
<i>Logical Index</i> =	Trunk group in error.
<i>Dump</i> =	NA.

AMLINFO-A

Software Release: 5E10 only

1. AUTOMATIC MAINTENANCE LIMIT INFORMATION (AMLINFO) AUDIT

An automatic maintenance limit defines the maximum number of trunks in a trunk group allowed to be out-of-service at one time. The automatic maintenance limit prevents too many two-way and outgoing trunks in a trunk group from being automatically removed from service. The number of trunks removed from service can exceed the automatic maintenance limit due to manual removes or critical problems.

2. DATA STRUCTURES

The AMLINFO audit references the trunk group information in RLRT_TRKG and TMamlnfo[]. Other data referenced include TMamlwork[] which contains an audit bit. Information for all the trunk groups is maintained in the administrative module in TMamlnfo[]. The trunk group information is audited by the automatic maintenance limit information (AMLINFO) audit by comparing each entry in TMamlnfo[] with data in RLRT_TRKG relation.

A related audit (AML) maintains the out-of-service data in each switching module.

3. AUDIT CHECKS

The AMLINFO audit invokes the DRSTATIC audit driver to step through the RLRT_TRKG relation. For each tuple in RLRT_TRKG, the AMLINFO audit makes the following checks:

AML_LIMIT = Check the AML limit.
OOSINV = Check the out-of-service counts.
TRKG_SIZE = Check the trunk group size.

For each entry in TMamlnfo[] that does not have an entry in RLRT_TRKG the AMLINFO audit makes the following check:

NOTZERO = Checks for empty entries.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AML_LIMIT

Error Code:	AML_LIMIT
Error Description:	TMamlnfo[].amllimit does not match the value calculated using the size of the trunk group (act_size) and the AML percent (amlpcnt) in the RLRT_TRKG tuple.
Possible Error Effect:	A trunk member could be automatically removed above the correct AML limit.
Corrective Action Taken:	TMamlnfo[].amllimit is corrected and a refresh of the trunk group out-of-service counts is scheduled.
Manual Action Required:	None.
Dump Description:	
Error Address =	Address of TMamlnfo[] entry.
Bad Data =	TMamlnfo[].amllimit for this trunk group.
Good Data =	AML limit as calculated from size of trunk group and percent.
Logical Index =	Trunk group member with invalid entry.

<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 2 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A non-zero entry in TMamlnfo[] has been found without a corresponding entry in RLRT_TRKG. This entry in TMamlnfo[] table should be empty.
Possible Error Effect:	This could indicate that data has not been cleaned up after removal of a trunk group or invalid data has been entered into the TMamlnfo[] table. This error could also indicate a problem with the TM AML process accessing and writing the data. Possible invalid data could be used to report and maintain AML status.
Corrective Action Taken:	This entry will be zeroed out and a refresh of the TMamlnfo[] data will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of this entry in TMamlnfo[].
<i>Bad Data</i> =	Next valid trunk group or maximum trunk group (MXTRKGRP).
<i>Good Data</i> =	Last valid trunk group.
<i>Logical Index</i> =	Trunk group number for this entry.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 3 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	Either the TMamlnfo[].tgooscnt (trunk group out-of-service count) has exceeded the TMamlnfo[].act_size (size of the trunk group) or the TMamlnfo[].ooscnt (AML out-of-service count) has exceeded the TMamlnfo[].tgooscnt.
Possible Error Effect:	Possible invalid out-of-service counts are used to determine if the AML limit has been reached.
Corrective Action Taken:	The TMamlnfo[].tgooscnt and TMamlnfo[].ooscnt counts will be reset to zero and the TMamlnfo[].valid bit cleared. A refresh will be scheduled to recalculate the out-of-service counts.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TMamlnfo[].act_size (Size of trunk group).
<i>Bad Data</i> =	TMamlnfo[].ooscnt (AML out-of-service count).
<i>Good Data</i> =	TMamlnfo[].tgooscnt (Trunk group out-of-service count).
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 4 Error Code: TRKG_SIZE

Error Code:	TRKG_SIZE
Error Description:	TMamlnfo[].act_size does not match the size of the trunk group in rIRT_TRKG tuple.
Possible Error Effect:	Possible invalid AML limit could be used to determine AML status.
Corrective Action Taken:	TMamlnfo[].act_size is reset to equal act_size in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlnfo[].
<i>Bad Data</i> =	TMamlnfo[].act_size (Size of trunk group).
<i>Good Data</i> =	Trunk group size (act_size) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

AMLINFO-B

Software Release: 5E11 - 5E12

1. AUTOMATIC MAINTENANCE LIMIT INFORMATION (AMLINFO) AUDIT

An automatic maintenance limit defines the maximum number of trunks in a trunk group allowed to be out-of-service at one time. The automatic maintenance limit prevents too many two-way and outgoing trunks in a trunk group from being automatically removed from service. The number of trunks removed from service can exceed the automatic maintenance limit due to manual removes or critical problems.

2. DATA STRUCTURES

The AMLINFO audit references the trunk group information in RLRT_TRKG and TMamlnfo[]. Other data referenced include TMamlwork[] which contains an audit bit.

Information for all the trunk groups is maintained in the administrative module in TMamlnfo[]. The trunk group information is audited by the automatic maintenance limit information (AMLINFO) audit by comparing each entry in TMamlnfo[] with data in RLRT_TRKG relation.

A related audit (AML) maintains the out-of-service data in each switching module.

3. AUDIT CHECKS

The AMLINFO audit invokes the DRSTATIC audit driver to step through the RLRT_TRKG relation. For each tuple in RLRT_TRKG, the AMLINFO audit makes the following checks:

ALARM	= Check the alarm level.
AML_LIMIT	= Check the AML limit.
BADFLAG	= Check the outage report flag.
OOSINV	= Check the out-of-service counts.
TRKG_SIZE	= Check the trunk group size.

For each entry in TMamlnfo[] that does not have an entry in RLRT_TRKG the AMLINFO audit makes the following check:

NOTZERO	= Checks for empty entries.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AML_LIMIT

Error Code:	AML_LIMIT
Error Description:	TMamlnfo[].amllimit does not match the value calculated using the size of the trunk group (act_size) and the AML percent (amlpcnt) in the RLRT_TRKG tuple.
Possible Error Effect:	A trunk member could be automatically removed above the correct AML limit.
Corrective Action Taken:	TMamlnfo[].amllimit is corrected and a refresh of the trunk group out-of-service

	counts is scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of TMamlnfo[] entry.
<i>Bad Data</i> =	TMamlnfo[].amlimit for this trunk group.
<i>Good Data</i> =	AML limit as calculated from size of trunk group and percent.
<i>Logical Index</i> =	Trunk group member with invalid entry.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An E911 trunk failure feature is activated and TMamlnfo[].alm_lvl does not match the alarm level in rIRT_TRKG tuple.
Possible Error Effect:	Possible invalid alarm level being used, may result in TM asserts.
Corrective Action Taken:	TMamlnfo[].alm_lvl is reset to equal alm_lvl in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlnfo[].
<i>Bad Data</i> =	TMamlnfo[].alm_lvl (alarm level of trunk).
<i>Good Data</i> =	Alarm level value (alm_lvl) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 3 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The E911 call error detection and alarm feature is activated or the route outage AML feature is activated and TMamlnfo[].outage_rpt does not match the outage report flag in rIRT_TRKG tuple.
Possible Error Effect:	Valid REPT AML REACHED messages may not be printed. Invalid REPT AML REACHED messages may be printed.
Corrective Action Taken:	TMamlnfo[].outage_rpt is reset to equal ffcf2 in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlnfo[].
<i>Bad Data</i> =	TMamlnfo[].outage_rpt (outage report flag of trunk).
<i>Good Data</i> =	Outage report flag (ffcf2) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 4 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A non-zero entry in TMamlnfo[] has been found without a corresponding entry in RLRT_TRKG. This entry in TMamlnfo[] table should be empty.
Possible Error Effect:	This could indicate that data has not been cleaned up after removal of a trunk group or invalid data has been entered into the TMamlnfo[] table. This error could also indicate a problem with the TM AML process accessing and writing the data. Possible invalid data could be used to report and maintain AML status.
Corrective Action Taken:	This entry will be zeroed out and a refresh of the TMamlnfo[] data will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of this entry in TMamlnfo[].
<i>Bad Data</i> =	Next valid trunk group or maximum trunk group (MXTRKGRP).
<i>Good Data</i> =	Last valid trunk group.
<i>Logical Index</i> =	Trunk group number for this entry.
<i>Dump</i> =	Entry for this trunk group in TMamlnfo[].

Table 5 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	Either the TMamlinfo[].tgooscnt (trunk group out-of-service count) has exceeded the TMamlinfo[].act_size (size of the trunk group) or the TMamlinfo[].ooscnt (AML out-of-service count) has exceeded the TMamlinfo[].tgooscnt.
Possible Error Effect:	Possible invalid out-of-service counts are used to determine if the AML limit has been reached.
Corrective Action Taken:	The TMamlinfo[].tgooscnt and TMamlinfo[].ooscnt counts will be reset to zero and the TMamlinfo[].valid bit cleared. A refresh will be scheduled to recalculate the out-of-service counts.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TMamlinfo[].act_size (Size of trunk group).
<i>Bad Data</i> =	TMamlinfo[].ooscnt (AML out-of-service count).
<i>Good Data</i> =	TMamlinfo[].tgooscnt (trunk group out-of-service count).
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

Table 6 Error Code: TRKG_SIZE

Error Code:	TRKG_SIZE
Error Description:	TMamlinfo[].act_size does not match the size of the trunk group in rIRT_TRKG tuple.
Possible Error Effect:	Possible invalid AML limit could be used to determine AML status.
Corrective Action Taken:	TMamlinfo[].act_size is reset to equal act_size in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlinfo[].
<i>Bad Data</i> =	TMamlinfo[].act_size (Size of trunk group).
<i>Good Data</i> =	Trunk group size (act_size) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

AMLINFO-C

Software Release: 5E13 and later

1. AUTOMATIC MAINTENANCE LIMIT INFORMATION (AMLINFO) AUDIT

An automatic maintenance limit defines the maximum number of trunks in a trunk group allowed to be out-of-service at one time. The automatic maintenance limit prevents too many two-way and outgoing trunks in a trunk group from being automatically removed from service. The number of trunks removed from service can exceed the automatic maintenance limit due to manual removes or critical problems.

2. DATA STRUCTURES

The AMLINFO audit references the trunk group information in RLRT_TRKG and TMamlinfo[]. Other data referenced include TMamlwork[] which contains an audit bit. Information for all the trunk groups is maintained in the administrative module in TMamlinfo[]. The trunk group information is audited by the automatic maintenance limit information (AMLINFO) audit by comparing each entry in TMamlinfo[] with data in RLRT_TRKG relation.

A related audit (AML) maintains the out-of-service data in each switching module.

3. AUDIT CHECKS

The AMLINFO audit invokes the DRSTATIC audit driver to step through the RLRT_TRKG relation.

For each tuple in RLRT_TRKG, the AMLINFO audit makes the following checks:

ALARM	= Check the alarm level.
AML_LIMIT	= Check the AML limit.
BADFLAG	= Check the outage report flag.
OOSINV	= Check the out-of-service counts.
TRKG_SIZE	= Check the trunk group size.

For each entry in TMamlinfo[] that does not have an entry in RLRT_TRKG the AMLINFO audit makes the following check:

NOTZERO	= Checks for empty entries.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AML_LIMIT

Error Code:	AML_LIMIT
Error Description:	TMamlinfo[].amllimit does not match the value calculated using the size of the trunk group (act_size) and the AML percent (amlpcnt) in the RLRT_TRKG tuple.
Possible Error Effect:	A trunk member could be automatically removed above the correct AML limit.
Corrective Action Taken:	TMamlinfo[].amllimit is corrected and a refresh of the trunk group out-of-service counts is scheduled.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of TMamlinfo[] entry.
<i>Bad Data</i> =	TMamlinfo[].amlimit for this trunk group.
<i>Good Data</i> =	AML limit as calculated from size of trunk group and percent.
<i>Logical Index</i> =	Trunk group member with invalid entry.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An E911 trunk failure feature is activated and TMamlinfo[].alm_lvl does not match the alarm level in rIRT_TRKG tuple.
Possible Error Effect:	Possible invalid alarm level being used, may result in TM asserts.
Corrective Action Taken:	TMamlinfo[].alm_lvl is reset to equal alm_lvl in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlinfo[].
<i>Bad Data</i> =	TMamlinfo[].alm_lvl (alarm level of trunk).
<i>Good Data</i> =	Alarm level value (alm_lvl) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

Table 3 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The E911 call error detection and alarm feature is activated or the route outage AML feature is activated and TMamlinfo[].outage_rpt does not match the outage report flag in rIRT_TRKG tuple.
Possible Error Effect:	Valid REPT AML REACHED messages may not be printed. Invalid REPT AML REACHED messages may be printed.
Corrective Action Taken:	TMamlinfo[].outage_rpt is reset to equal outage_rpt in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlinfo[].
<i>Bad Data</i> =	TMamlinfo[].outage_rpt (outage report flag of trunk).
<i>Good Data</i> =	Outage report flag (outage_rpt) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

Table 4 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A non-zero entry in TMamlinfo[] has been found without a corresponding entry in RLRT_TRKG. This entry in TMamlinfo[] table should be empty.
Possible Error Effect:	This could indicate that data has not been cleaned up after removal of a trunk group or invalid data has been entered into the TMamlinfo[] table. This error could also indicate a problem with the TM AML process accessing and writing the data. Possible invalid data could be used to report and maintain AML status.
Corrective Action Taken:	This entry will be zeroed out and a refresh of the TMamlinfo[] data will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of this entry in TMamlinfo[].
<i>Bad Data</i> =	Next valid trunk group or maximum trunk group (MXTRKGRP).
<i>Good Data</i> =	Last valid trunk group.
<i>Logical Index</i> =	Trunk group number for this entry.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

Table 5 Error Code: OOSINV

Error Code:	OOSINV

Error Description:	Either the TMamlinfo[].tgooscnt (trunk group out-of-service count) has exceeded the TMamlinfo[].act_size (size of the trunk group) or the TMamlinfo[].ooscnt (AML out-of-service count) has exceeded the TMamlinfo[].tgooscnt.
Possible Error Effect:	Possible invalid out-of-service counts are used to determine if the AML limit has been reached.
Corrective Action Taken:	The TMamlinfo[].tgooscnt and TMamlinfo[].ooscnt counts will be reset to zero and the TMamlinfo[].valid bit cleared. A refresh will be scheduled to recalculate the out-of-service counts.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TMamlinfo[].act_size (Size of trunk group).
<i>Bad Data</i> =	TMamlinfo[].ooscnt (AML out-of-service count).
<i>Good Data</i> =	TMamlinfo[].tgooscnt (trunk group out-of-service count).
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

Table 6 Error Code: TRKG_SIZE

Error Code:	TRKG_SIZE
Error Description:	TMamlinfo[].act_size does not match the size of the trunk group in rIRT_TRKG tuple.
Possible Error Effect:	Possible invalid AML limit could be used to determine AML status.
Corrective Action Taken:	TMamlinfo[].act_size is reset to equal act_size in rIRT_TRKG tuple for this trunk group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in TMamlinfo[].
<i>Bad Data</i> =	TMamlinfo[].act_size (Size of trunk group).
<i>Good Data</i> =	Trunk group size (act_size) in rIRT_TRKG tuple.
<i>Logical Index</i> =	Trunk group number.
<i>Dump</i> =	Entry for this trunk group in TMamlinfo[].

ANILHB

Software Release: 5E10 and later

1. INCOMING LINE HISTORY BLOCK FOR UNSHARED ANALOG LINES (ANILHB) AUDIT

Analog incoming line history blocks are used by several of the LASS features to store incoming call data associated with unshared analog lines. The ANILHB audit performs dynamic to static existence checks of the dynamic RLAN_ILHB relation. This audit runs segmented in the operating system for distributed switching-switching module (OSDS-M) environment. The audit is also scheduled as a post initialization audit as part of a selective initialization.

2. DATA STRUCTURES

The RLAN_ILHB relation is a dynamic indirect access nonsequential keyed relation. The ANILHB audit checks several pieces of static data to verify the proper existence of rLAN_ILHB tuples. The static data, referenced, is shown.

3. AUDIT CHECKS

The ANILHB audit steps through the RLAN_ILHB relation and verifies whether or not a tuple in the relation should exist. For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

If a rLAN_ILHB tuple exists that should not, the audit removes the tuple from the RLAN_ILHB relation. The ANILHB audit produces these error codes.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rANILHB dynamic tuple without a corresponding static tuple or global parameter in the system. rANILHB.lport NOTZERO= Checks for empty entries. Glac_access Glcot_access is set to unavailable. rIFC_LINE.ac_alw DBNO (with rIFC_LINE.port = rLPT_PT.port) rIFC_LINE.cot_alw DBNO (with rIFC_LINE.port = rLPT_PT.port).
Possible Error Effect:	If call processing attempts to access this tuple, services/operations may be disrupted.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rANILHB tuple in question.
<i>Dump =</i>	rANILHB tuple in question.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has found an inconsistency between the static relations rLPT_PT and rFC_LINE. The audit is using the port and party fields of a rLPT_PT tuple to access a tuple in rFC_LINE, however it is not able to access the rFC_LINE tuple so the audit cannot continue to do checks.
Possible Error Effect:	The ANILHB audit will go on to the next tuple.
Corrective Action Taken:	The audit will report an error and continue on to next tuple.
Manual Action Required:	Check the rLPT_PT and rFC_LINE relations in the ODD. The rLPT_PT key, lport, found in the "GOOD DATA" field of the audit report, should be used to access the rLPT_PT tuple. The rLPT_PT tuple should have a port and party field which serve as the keys to the rFC_LINE relation. There should exist a corresponding tuple in rFC_LINE with the port and party found in rLPT_PT.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the port field of the buffer used to access the rFC_LINE tuple.
<i>Bad Data =</i>	Party field of the buffer used to access the rFC_LINE tuple.
<i>Good Data =</i>	lport field of the rLPT_PT tuple used access the associated rFC_LINE tuple.
<i>Logical Index =</i>	The key of the rANILHB tuple in question.
<i>Dump 1 =</i>	rANILHB tuple in question.
<i>Dump 2 =</i>	Associated rLPT_PT tuple.
<i>Dump 3 =</i>	Buffer used to access rFC_LINE.

ANOLHB

Software Release: 5E10 and later

1. OUTGOING LINE HISTORY BLOCK FOR UNSHARED ANALOG LINES (ANOLHB) AUDIT

Analog outgoing line history blocks are used by several of the LASS features to store outgoing call data associated with unshared analog lines. The ANOLHB audit performs dynamic to static existence checks of the dynamic RLAN_OLHB relation. This audit runs segmented in the operating system for distributed switching-switching module (OSDS-SM) environment. It is also scheduled as a post initialization audit as part of a selective initialization.

2. DATA STRUCTURES

The RLAN_OLHB relation is a dynamic indirect access nonsequential keyed relation. The ANOLHB audit checks several pieces of static data to verify the proper existence of rLAN_OLHB tuples. The static data, referenced, is shown.

3. AUDIT CHECKS

The ANOLHB audit steps through the RLAN_OLHB relation, and verifies whether or not a tuple in the relation should exist. For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

If a rLAN_OLHB tuple exists that should not, the audit removes the tuple from the RLAN_OLHB relation. The ANOLHB audit produces these error codes.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rLANOLHB dynamic tuple without a corresponding static tuple or global parameter in the system. rLANOLHB.lport rLPT_PT.lport (with lpt_type = LPT_OTHER). Glar_access Glar_access is set to unavailable. rIFC_LINE.ar_alw DBNO (with rIFC_LINE.port = rLPT_PT.port).
Possible Error Effect:	If call processing attempts to access this tuple, services/operations may be disrupted.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of AN_OLHB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rLAN_OLHB tuple in question.
<i>Dump</i> =	rLAN_OLHB tuple in question.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has found an inconsistency between the static relations rLPT_PT and

	rIFC_LINE. The audit is using the port and party fields of a rILPT_PT tuple to access a tuple in rIFC_LINE, however it is not able to access the rIFC_LINE tuple so the audit cannot continue to do checks on this tuple.
Possible Error Effect:	The ANOLHB audit will not complete its checks on this tuple.
Corrective Action Taken:	The audit will print an error and continue on to the next tuple.
Manual Action Required:	Check the rILPT_PT and rIFC_LINE relations in the ODD. The rILPT_PT key, lport, found in the "GOOD DATA" field of the audit report, should be used to access the rILPT_PT tuple. The rILPT_PT tuple should have a port and party field which serve as the keys to the rIFC_LINE relation. There should exist a corresponding tuple in rIFC_LINE with the port and party found in rILPT_PT.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the port field of the buffer used to access the rIFC_LINE tuple.
<i>Bad Data =</i>	Party field of the buffer used to access the rIFC_LINE tuple.
<i>Good Data =</i>	lport field of the rILPT_PT tuple used access the associated rIFC_LINE tuple.
<i>Logical Index =</i>	The key of the rIAN_OLHB tuple in question.
<i>Dump 1 =</i>	rIAN_OLHB tuple in question.
<i>Dump 2 =</i>	Associated rILPT_PT tuple.

ANQ

Software Release: 5E10 and later

1. AUTOMATIC TERMINAL SETUP NOTIFICATION QUEUE (ANQ) AUDIT

The automatic terminal setup (ATS) feature downloads certain parameters which are essential for terminal operation on a National ISDN standard basic rate interface (BRI) upon terminal request. A terminal can request a parameter download after initialization of layer 3 or after it has been notified by the switch that a service order change has been completed. Parameters which are downloaded include directory numbers (DNs), bearer capabilities per DN, the call appearance (CA) map per DN, DN appearances (DNA) per DN, feature activators (FAs), feature indicators (FIs), and service description parameters. In this feature, the terms "customer premise equipment (CPE)" and "terminal" are interchangeable. A terminal is a logical entity with a single terminal endpoint identifier (TEI) used for signaling. Single physical terminals can communicate as one or more logical terminals (such as, one or more TEIs). Terminal downloading denotes downloading of the above mentioned parameters for a logical terminal associated with the physical equipment.

The RLANQ relation is used to represent entries on the ATS notification queue. The queue consists of up to 2047 entries and is kept in phase-protected memory to be preserved through an SM selective initialization. The relation is re-initialized during a SM full initialization with or without pump and an SM power up full initialization. This relation is preserved over a software release transition (retrofit). This relation is used in the downloading of CPEs. These tuples may be linked to an in service queue, an out of service queue, both at the same time, or not linked to any queue.

The purpose of the ANQ audit is to ensure the integrity of the rLANQ tuples in the RLANQ relation. The ANQ audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in segmented mode. All linkages are verified by the audit.

2. DATA STRUCTURES

The RLANQ dynamic relation contains information used to determine what ISDN CPEs need notification about download processes. Each tuple contains the **port** number of the standard terminal needing a notification, **is_party** which is a bitmap representing in service state of the 8 parties associated with this port, **oos_party** which is a bit map representing the out of service state of the 8 parties associated with this port, the **quarstat** field which represents the quarantine status for auditing purposes, **andbism** which is the member linkage to the in service queue, **andboosm** which is the member linkage to the out of service queue, and some spare bits. Linkage to the in service or out of service queue is defined in the RLDB_HEADC relation.

3. AUDIT CHECKS

The ANQ audit uses the DRINDEX audit driver to traverse through each tuple in the RLANQ relation. For each tuple, the ANQ audit will make the following checks: If the ANQ audit is entered:

- E_BIT = Check to make sure corresponding bits are not both set to 1 in the is_party and oos_party fields.
- LLMZERO = Check to make sure that a tuple linked to the In Service Queue has at least one bit set in is_party.
- MRL_ZERO = Check to make sure that a tuple linked to the Out of Service Queue has at least one bit set in oos_party.
- NOTZERO = Check to make sure that an rLANQ tuple that is not linked to the in service or out of service queue has the corresponding bit field (is_party or oos_party) not all zeros.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	At least one bit is set in both the in service bit field, is_party , and the out of service bit field, oos_party , for the rIANQ tuple under audit. A party can not be both in service and out of service at the same time on the same port.
Possible Error Effect:	The affected party or parties may not be notified of a service profile by the ATS feature.
Corrective Action Taken:	The bits representing the parties marked as both in service and out of service are cleared in both the is_party and oos_party fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIANQ tuple in error.
<i>Bad Data =</i>	The value of the oos_party field found in the rIANQ tuple.
<i>Good Data =</i>	The value of the is_party field found in the rIANQ tuple.
<i>Logical Index =</i>	The logical key of the rIANQ tuple in error.
<i>Dump =</i>	The rIANQ tuple in error.

Table 2 Error Code: LLMZERO

Error Code:	LLMZERO
Error Description:	The audit found an rIANQ tuple linked to the in service queue. However, no bits were set in the is_party field of the tuple. There must be at least one bit set in the is_party field representing an in service party.
Possible Error Effect:	The rIANQ tuple resource may be lost.
Corrective Action Taken:	The rIANQ tuple is unlinked from the in service queue and the corresponding linkage field is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIANQ tuple in error.
<i>Bad Data =</i>	The value of the is_party field in the rIANQ tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The value of the logical key of the rIANQ tuple under audit.
<i>Dump =</i>	The rIANQ tuple under audit.

Table 3 Error Code: MRL_SERO

Error Code:	MRL_SERO
Error Description:	The audit found an rIANQ tuple that was linked to the out of service list. However, the oos_party field had no bits set.
Possible Error Effect:	Possible loss of the rIANQ tuple resource.
Corrective Action Taken:	The rIANQ tuple is unlinked from the out of service list, and the linkage field is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIANQ tuple in error.
<i>Bad Data =</i>	The value of the oos_party field in the rIANQ tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The value of the logical key of the rIANQ tuple under audit.
<i>Dump =</i>	The rIANQ tuple under audit.

Table 4 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The audit found an rIANQ tuple with a nonzero is_party field but not linked to the in service queue, or it found an rIANQ tuple with a nonzero oos_party field but not linked to the out of service queue. The value of the good data field tells which error occurred. If good data is 0, the is_party field was found to be in

	error. If good data is 1, the oos_party was found to be in error.
Possible Error Effect:	The rIANQ tuple resource may be lost.
Corrective Action Taken:	The is_party or oos_party field is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIANQ tuple in error.
<i>Bad Data =</i>	The value of the is_party or oos_party field in the rIANQ.
<i>Good Data =</i>	0 or 1.
<i>Logical Index =</i>	The value of the logical key of the rIANQ tuple under audit.
<i>Dump =</i>	The rIANQ tuple under audit.

APB-A

Software Release: 5E10 - 5E13

1. ASSOCIATED PROCESS BLOCK (APB) AUDIT

The feature execution (FEX) system acts as the operating system for business and residence custom services (BRCS) features. Each terminal process within FEX has at least three parts: a near-end subprocess, a call coordinating subprocess, and a plain old telephone service (POTS) feature mode subprocess. Other subprocesses may occur in feature calls, such as those defining the type of feature. Associated process blocks hold the programs and store data for each subprocess; they keep track of the state of the call and the calling features being used. An associated process block is needed for each feature. The associated process block audit (APB) ensures that the information kept in the associated process block relation (RLAPB) is consistent. APBs are only used by OSPS FEX. Non-OSPS fex uses RLCLDBs and RLMDDBs.

The APB audit is part of the feature control audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The APB audit reads the tuples of the associated process block relation (RLAPB), a dynamic relation in the SMS. The associated process block relation is owned by the process control block linkage area relation (RLPCBLA), a dynamic relation in the SMS. The rAPB tuples are linked by an owner-member linkage to rPCBLA for each terminal process. The rAPB tuples contain subprocess information such as its ID, state, and a resource field that includes process ID information. The APB relation also has owner-member linkage with RLPOSDB, RLTAPOSDB and set owner linkage to RLBUFMSG.

3. AUDIT CHECKS

The audit calls DRINASEK driver to check rAPB's linkages. The driver then invokes a semantic check function to perform the following checks.

- (a) The audit verifies that the idle tuples of rAPB are owned by the correct head cell in rPCBLA.
- (b) The audit reads the cldb_flag in the rPCBLA, which indicates usage of the shared linkage in rPCBLA with rCLDB. If the cldb_flag for this rAPBs owner is DBYES, recovery is initiated.
- (c) The audit reads the apb_tag field, which specifies the resources the subprocess is controlling, and ensures that it is valid by checking the appropriate list of enumerated values.
- (d) The audit verifies the apb_tag against loop-no.

4. ERROR HANDLING

For both syntactic (linkage) and semantic (apb_tag field) errors, the audit purges the owning process and releases the associated process block tuples to the idle list.

The audit schedules the buffer message audit (BUFMSG) as a related audit.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: APB_TAG

Error Code:	APB_TAG
Error Description:	The APB does not represent a valid physical resource.
Possible Error Effect:	The process would be purged next time it received a stimulus.
Corrective Action Taken:	The corresponding process is purged from the system and the APB is returned to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of APB.
<i>Bad Data</i> =	apbtag.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	APB number.
<i>Dump</i> =	rlAPB tuple.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An Associated Process Block is linked to a Process Control Block Linkage Area tuple, but the PCBLA indicates that it should not be ("cldb_flag").
Possible Error Effect:	Asserts, SPPs.
Corrective Action Taken:	The process involved in the call is purged, and data is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rlAPB.
<i>Bad Data</i> =	Value of cldb_flag in rIPCBLA.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key field of rlAPB.
<i>Dump 1</i> =	rlAPB.
<i>Dump 2</i> =	rIPCBLA.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The APB loopno does not match the FEX type of APB_TAG (should be 1).
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rlAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rlAPB.
<i>Dump</i> =	rlAPB tuple.

Table 4 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The APB is owned by an invalid head cell.
Possible Error Effect:	The APB will be lost to the system.
Corrective Action Taken:	The corresponding APB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of APB.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	APB number.
<i>Dump</i> =	rlAPB tuple.

Table 5 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The position loop count does not match the FEX type of APB_TAG.
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB. The corresponding position terminal process will purge the call from the system.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rIAPB.
<i>Dump</i> =	rIAPB tuple.

Table 6 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The APB loopno does not match the FEX type of APB_TAG (should be 2).
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rIAPB.
<i>Dump</i> =	rIAPB tuple.

Table 7 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The APB loopno does not match the FEX type of APB_TAG (should be 3).
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rIAPB.
<i>Dump</i> =	rIAPB tuple.

APB-B

Software Release: 5E14 and later

1. ASSOCIATED PROCESS BLOCK (APB) AUDIT

The feature execution (FEX) system acts as the operating system for business and residence custom services (BRCS) features. Each terminal process within FEX has at least three parts: a near-end subprocess, a call coordinating subprocess, and a plain old telephone service (POTS) feature mode subprocess. Other subprocesses may occur in feature calls, such as those defining the type of feature. Associated process blocks hold the programs and store data for each subprocess; they keep track of the state of the call and the calling features being used. An associated process block is needed for each feature. The associated process block audit (APB) ensures that the information kept in the associated process block relation (RLAPB) is consistent. APBs are only used by OSPS FEX. Non-OSPS fex uses RLCLDBs and RLMDDBs.

The APB audit is part of the feature control audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The APB audit reads the tuples of the associated process block relation (RLAPB), a dynamic relation in the SMS. The associated process block relation is owned by the process control block linkage area relation (RLPCBLA), a dynamic relation in the SMS. The rIAPB tuples are linked by an owner-member linkage to rIPCBLA for each terminal process. The rIAPB tuples contain subprocess information such as its ID, state, and a resource field that includes process ID information. The APB relation also has owner-member linkage with RLPOSDB, RLTAPOSDB and set owner linkage to RLBUFMSG.

3. AUDIT CHECKS

The audit calls DRINASEK driver to check rIAPB's linkages. The driver then invokes a semantic check function to perform the following checks.

- (a) The audit verifies that the idle tuples of rIAPB are owned by the correct head cell in rIPCBLA.
- (b) The audit reads the cldb_flag in the rIPCBLA, which indicates usage of the shared linkage in rIPCBLA with rICLDB. If the cldb_flag for this rIAPB's owner is DBYES, recovery is initiated.
- (c) The audit reads the apb_tag field, which specifies the resources the subprocess is controlling, and ensures that it is valid by checking the appropriate list of enumerated values.
- (d) The audit verifies the apb_tag against loop-no.

4. ERROR HANDLING

For both syntactic (linkage) and semantic (apb_tag field) errors, the audit purges the owning process and releases the associated process block tuples to the idle list.

The audit schedules the buffer message audit (BUFMSG) as a related audit.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: APB_TAG

Error Code:	APB_TAG
Error Description:	The APB does not represent a valid physical resource.
Possible Error Effect:	The process would be purged next time it received a stimulus.
Corrective Action Taken:	The corresponding process is purged from the system and the APB is returned to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of APB.
<i>Bad Data</i> =	apbttag.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	APB number.
<i>Dump</i> =	rlAPB tuple.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An Associated Process Block is linked to a Process Control Block Linkage Area tuple, but the PCBLA indicates that it should not be ("cldb_flag").
Possible Error Effect:	Asserts, SPPs.
Corrective Action Taken:	The process involved in the call is purged, and data is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rlAPB.
<i>Bad Data</i> =	Value of cldb_flag in rlPCBLA.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key field of rlAPB.
<i>Dump 1</i> =	rlAPB.
<i>Dump 2</i> =	rlPCBLA.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The APB loopno does not match the FEX type of APB_TAG (should be 1).
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rlAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rlAPB.
<i>Dump</i> =	rlAPB tuple.

Table 4 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The APB is owned by an invalid head cell.
Possible Error Effect:	The APB will be lost to the system.
Corrective Action Taken:	The corresponding APB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of APB.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	APB number.
<i>Dump</i> =	rlAPB tuple.

Table 5 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The position loop count does not match the FEX type of APB_TAG.
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB. The corresponding position terminal process will purge the call from the system.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rIAPB.
<i>Dump</i> =	rIAPB tuple.

Table 6 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The APB loopno does not match the FEX type of APB_TAG (should be 2).
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rIAPB.
<i>Dump</i> =	rIAPB tuple.

Table 7 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The APB loopno does not match the FEX type of APB_TAG (should be 3).
Possible Error Effect:	Calls that don't exist may be serviced and calls that do exist may not be serviced.
Corrective Action Taken:	The TAPOSDB and PCBLA are unlinked from the APB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPB.
<i>Bad Data</i> =	loop_no.
<i>Good Data</i> =	model_id.
<i>Logical Index</i> =	Key of rIAPB.
<i>Dump</i> =	rIAPB tuple.

APBUF

Software Release: 5E10 and later

1. APPLICATIONS PROCESSOR MESSAGE BUFFER (APBUF) AUDIT

The relation RLAPBUF contains super message tuple buffers. Each buffer holds one or more operating system for distributed switching (OSDS) messages received from the business and residential customer services (BRCS) client. Super message tuples (rAPBUF) are temporarily queued up on a priority basis per applications processor (AP), before being sent to the AP. Each queue consists of a headcell (RLAP_HEADC), and zero or more super message buffers (RLAPBUF), linked together with a standard relational definition language (RDL) own-member linkage. The relation RLAP_HEADC contains the owner tuples for the idle list or rAPBUF tuples and the busy super message queues. The indirect access sequential key driver (DRINASEK) is used to step through rAPBUF tuple by tuple. The APBUF audit verifies the linkages between the RLAPBUF and RLAP_HEADC relations for each rAPBUF tuple. If an invalid linkage is found, the rAPBUF tuple is recovered to the idle list. For the invalid linkage case, an attempt is made to patch the list on which the bad tuple was found.

2. DATA STRUCTURES

The RLAPBUF is the dynamic relation used by the application layer for storing super messages. The RLAP_HEADC is the dynamic relation used by the application layer for queuing of super message buffer tuples. The APBUF audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The APBUF audit verifies the set member linkage from the RLAPBUF relation to the RLAP_HEADC relation. The audit treats each of the lists as a "critical" list. This means that if an error is detected, the list is patched instead of being torn down. The indirect access sequential key driver (DRINASEK) is utilized to step through the RLAPBUF relation tuple by tuple. The audit performs the point-to point-back check between RLAPBUF and RLAP_HEADC. The driver also checks the key of the tuple for validity and the quarstate to see if the tuple has quarantined through the driver functions.

4. ERROR HANDLING

If a function detects a linkage error, the driver invokes recovery. Since all of the RLAP_HEADC lists are being considered critical, patches are made to the list if possible. Then the mutilated tuple is zeroed out and placed on the idle list. DRINASEK determines the idle list. The recovery of the RLAP_HEADC is propagated by scheduling the APHEADC audit as related.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

APDU

Software Release: 5E10 and later

1. APPLICATION PROTOCOL DATA UNIT (APDU) AUDIT

For a TR303 remote terminal (RT) on an integrated digital carrier unit (IDCU) the embedded operations channel (EOC) is used by the Level 7 administration to process tasks using messages. The rIAPDU tuple is linked to the rIRTEILA or rIRTLB tuples to identify the process receiving or transmitting messages. The rIAPDU tuple owns the rIRMSGBUF tuples which contain the messages. The APDU audit maintains the integrity of these linkages.

The dynamic data structures for EOC/TMC message processing on TR303 RT are shown in Figure 1 .

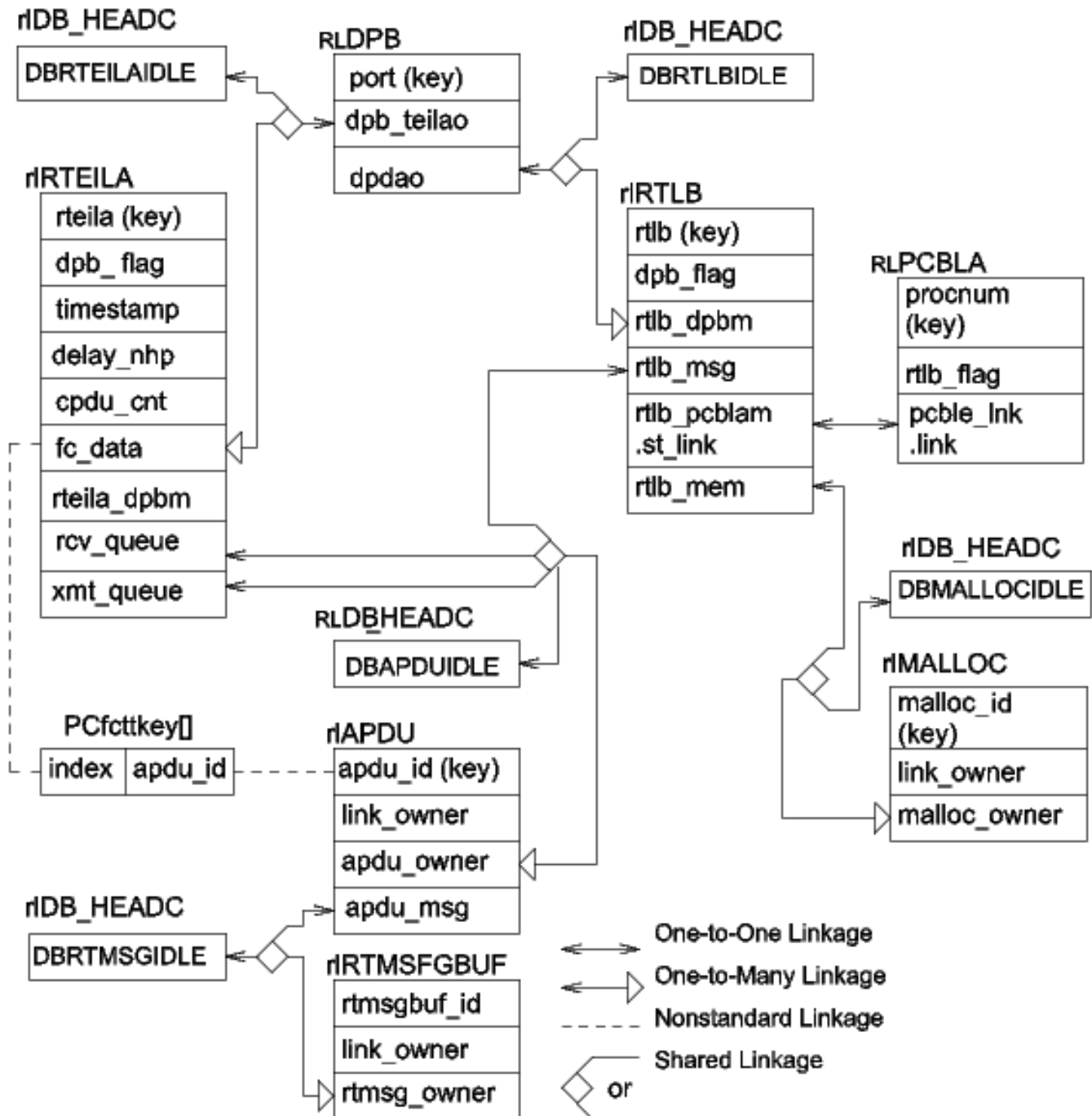


Figure 1 (APDU-1) Dynamic Data Structures for EOC/TMC Message Processing on TR30 RT

2. DATA STRUCTURES

The rAPDU tuple is linked to the idle list (rDB_HEADC) or used to identify the application which is receiving or transmitting messages. When used for message control the rAPDU can be linked to the inbound queue (rRTEILA), outbound queue (rRTEILA), or the application process linkage (rRTLB). The link_owner field is used to determine the correct linkage.

3. AUDIT CHECKS

The APDU audit invokes the DRINASEK driver to make the linkages and semantic checks for each rIAPDU tuple. For each tuple, the APDU audit makes the following checks:

INVOWNER = Verify link_owner field.

INV_TIMER = Verify index to timer array.

MEMBAD = Check member linkage to rIRTEILA, rIRTLB, rIDALB, or rIDB_HEADC.

MISSTUP = Verify the existence of a rIRTMSGBUF tuple.

MSGPTR = Verify curr_msg and delay_nhp field for non-head of outgoing queue.

NOTZERO = Verify curr_msg and fc_data field for head of outgoing queue.

OWNLINK = Verify rIRTMSGBUF linkage to rIAPDU tuple.

PORT_TM = Verify port field.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	This rIAPDU tuple has a key to a timer (rIAPDU.fc_data) that is the index to an array that contains the rIAPDU key for each timer. The timer index could be zero when the rIRTEILA.delay_nhp is DBNO (0) or the timer array for this index does not have this rIAPDU key.
Possible Error Effect:	Improper processing of messages on outbound queue after timer has expired.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The EOC/TMC channel will not be recovered unless a linkage error exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Value of index (rIAPDU.fc_data).
<i>Good Data</i> =	Key of rIAPDU in timer array or 0 if no timer exists.
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump</i> =	Contents of rIAPDU tuple.

Table 2 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The link_owner field has been found in an invalid state.
Possible Error Effect:	The message queue is lost.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The port will not be recovered unless a linkage error exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	The link_owner field in rIAPDU tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)

<i>Dump</i> =	Contents of rIAPDU tuple.

Table 3 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	This rIAPDU tuple is not linked to the owner indicated by the flag (rIAPDU.link_owner).
Possible Error Effect:	The messages for this channel could be dropped.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The EOC/TMC channel will not be recovered unless a linkage error exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Key of owner tuple (rIRTEILA, rIRTLB, rIDALB, or rIDB_HEADC).
<i>Good Data</i> =	The link_owner field as follows: 6 = MIB head cell in rIDB_HEADC. 5 = Inbound rIDALB. 4 = Outbound rIDALB. 3 = rIRTLB. 2 = Inbound rIRTEILA. 1 = Outbound rIRTEILA. 0 = rIDB_HEADC.
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump 1</i> =	Contents of rIAPDU tuple.
<i>Dump 2</i> =	Contents of either rIDPB or rIDB_HEADC.

Table 4 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The rIAPDU tuple is linked to one of the rIRTEILA queues and is not linked to the rIRMSGBUF tuple.
Possible Error Effect:	The messages for this channel could be dropped.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The EOC/TMC channel will not be recovered unless a linkage error exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Key of missing rIRMSGBUF tuple.
<i>Good Data</i> =	The key of the rIRMSGBUF tuple (rIAPDU.curr_msg).
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump</i> =	Contents of rIAPDU tuple.

Table 5 Error Code: MSGPTR

Error Code:	MSGPTR
Error Description:	This rIAPDU tuple is at the head of the outbound queue. If the curr_msg field is zero the rIRTEILA.delay_nhp should be DBYES (1).
Possible Error Effect:	Improper processing of messages on outbound queue.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The EOC/TMC channel will not be recovered unless a linkage error exists.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Value of rIAPDU.curr_msg.
<i>Good Data</i> =	Value of rIRTEILA.delay_nhp.
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump 1</i> =	Contents of rIAPDU tuple.
<i>Dump 2</i> =	Contents of rIRTEILA tuple.

Table 6 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	This rIAPDU tuple is not at the head of the outbound queue. Both the curr_msg and fc_data fields should be zero.
Possible Error Effect:	Improper processing of messages on outbound queue.
Corrective Action Taken:	The curr_msg and fc_data fields in rIAPDU are set to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Value of rIAPDU.curr_msg.
<i>Good Data</i> =	Value of rIAPDU.fc_data.
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump</i> =	Contents of rIAPDU tuple.

Table 7 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	This rIAPDU tuple is processing a rRTMSGBUF tuple that does not link to this rIAPDU.
Possible Error Effect:	Improper processing of messages on outbound queue.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The EOC/TMC channel will not be recovered unless a linkage error exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Key of linked rRTMSGBUF tuple.
<i>Good Data</i> =	Key of processing rRTMSGBUF tuple (rIAPDU.curr_msg).
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump 1</i> =	Contents of rIAPDU tuple.
<i>Dump 2</i> =	Contents of rRTMSGBUF tuple being processed.

Table 8 Error Code: PORT_TM

Error Code:	PORT_TM
Error Description:	The port field in rIAPDU does not match the port name in the rDPB tuple linked to the owner rIRTEILA tuple.
Possible Error Effect:	Messages are queued to the wrong channel and are lost.
Corrective Action Taken:	The rIAPDU tuple is recovered which will also recover any messages linked to this tuple and purge any process associated with this tuple. The EOC/TMC channel will not be recovered unless a linkage error exists between the rIRTEILA and rIAPDU tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Port in rIAPDU.port.
<i>Good Data</i> =	Port in rIRTEILA.rteila_dpbm.st_ownk.
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id)
<i>Dump 1</i> =	Contents of rIAPDU tuple.
<i>Dump 2</i> =	Contents of rIRTEILA tuple.

APHEADC

Software Release: 5E10 and later

1. APPLICATIONS PROCESSOR MEMORY BUFFER HEAD CELLS (APHEADC) AUDIT

The relation RLAP_HEADC contains owner head cell tuples for the super message tuples in RLAPBUF. There is an idle list for rIAPBUF tuples and two priority lists per applications processor (AP) link on a switching module (SM). An rIAP_HEADC tuple is allocated for each of these lists. The linkage driver (DRLINK) is used to step through rIAP_HEADC tuple by tuple. The APHEADC audit verifies the head and tail linkages between the owner rIAP_HEADC tuple and the member rIAPBUF tuples. Also, the audit tears down the queue of rIAPBUF tuples when there is no corresponding AP entry in the AP link status table.

2. DATA STRUCTURES

The APHEADC audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The APHEADC audit insures the integrity of the RLAP_HEADC relation. The linkage driver (DRLINK) is used to step through the RLAP_HEADC relation tuple by tuple. The audit performs the point-to point-back check between RLAP_HEADC and RLAPBUF. The driver also checks the key (apheadkey) of the tuple for validity and the quarstate to see if the tuple has been quarantined through the driver functions.

The semantic function insures rIAPBUF tuple validity.

4. ERROR HANDLING

When a tuple is found with a broken linkage, the tuple is zeroed. The result is to lose the whole list of rIAPBUF tuples. Since the APBUF audit is scheduled as a related audit, the rIAPBUF tuples left without an owner are recovered.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	There is an rIAP_HEADC queue of rIAPBUF messages for a link to an AP. However the AP link status table does not have a corresponding entry for this queue.
Possible Error Effect:	The rIAPBUF tuples are a lost resource.
Corrective Action Taken:	The rIAPBUF tuples linked off of this queue are returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of corresponding PSaplst[] entry.
<i>Bad Data</i> =	Site ID of corresponding PSaplst[] entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Physical key of rIAP_HEADC tuple.
<i>Dump 1</i> =	PSaplst entry.
<i>Dump 2</i> =	rIAP_HEADC tuple.

APLST

Software Release: 5E10 and later

1. APPLICATIONS PROCESSOR LINK STATUS TABLE (APLST) AUDIT

The applications processor (AP) link status table has one and only one entry for every AP on a given SM. A entry contains fields for an apsite and apindx. The other fields in give information on each of the DSL data links to an AP. Users of this other information do self audit on this data. The APLST audit compares the apsite and apindx fields against the corresponding fields in the static relation rLAPSITE. There must exist an rLAPSITE tuple for each entry to exist. If a rLAPSITE tuple does not exist, then the link status table entry is invalid and this audit recovers it to zero.

2. DATA STRUCTURES

The APLST audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment. The relationships between data structures audited by the APCP audits is shown in Figure 1 .

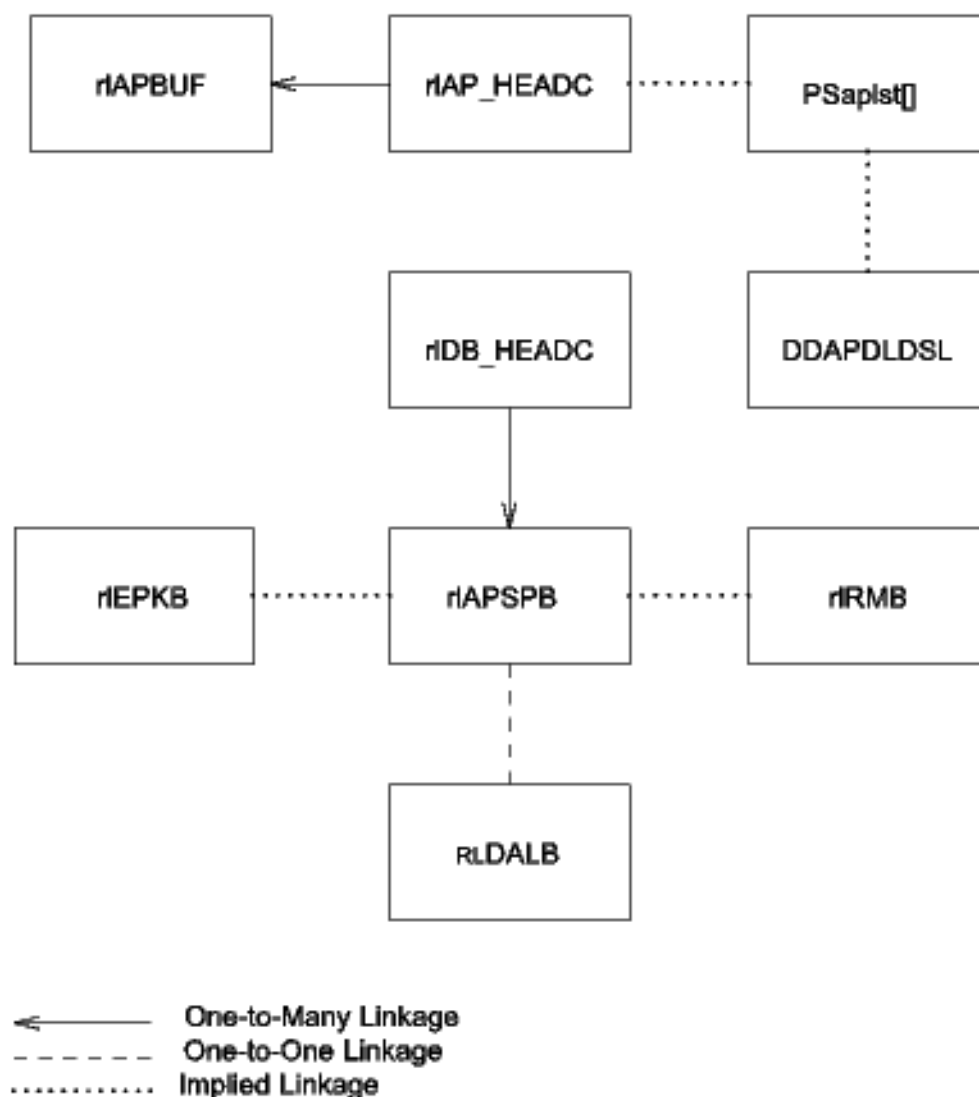


Figure 1 (APLST-1) Relationships Between Data Structures Audited by the APCP Audits

3. AUDIT CHECKS

The APLST audit insures that every link status table entry has a static equivalent in the relation RLAPSITE. The audit is entered from the audit control by a call to a function. The dynamic to static function is also called. These are the steps performed:

- A loop is performed through the link status table.
- For every non-zero apsite attribute in the table, a database read of the relation RLAPSITE is performed. A tuple should be found for each of the non-zero entries.
- With every successful read of RLAPSITE, the apindex attributes are compared.
- The link status table entries with zero in the apsite field are checked to verify that the remaining portion of the entry is zero.

4. ERROR HANDLING

Error correction is performed in-line within the dynamic to static function. The possible error and recovery processes are:

- An audit error is reported when the apsite field in the link status table is zero and the remaining portion of the entry is non-zero. The remaining portion of the entry is recovered to zero.
- Extra entries in the link status table generate audit errors and are recovered by zeroing out the entries in the table.
- If the apindex indices of the link status table and rIAPSITE do not match, the table is recovered to the value found in rIAPSITE.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_LNK

Error Code:	ACT_LNK
Error Description:	One of the AP links has been marked as active in the AP link status table (PSaplst) but there is no terminal process alive for this link.
Possible Error Effect:	Users of the AP link status table could be queuing/dequeuing messages to an AP link which is not active. Also the MCC might be reporting that the AP link is active when it is really disconnected.
Corrective Action Taken:	The bit in the representing the link to the AP is set to zero in PSaplst.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PSaplst[] entry.
<i>Bad Data =</i>	AP act_lnk field of PSaplst[] entry.
<i>Good Data =</i>	act_lnk field with link bit idled.
<i>Logical Index =</i>	xxxxyyyy where: xxxx = PSaplst index yyyy = PSaplst relative link.
<i>Dump =</i>	PSaplst entry, rIXNDSL tuple, rIDPB tuple, rIDALB tuple, rIAPSPB tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The AP link status table has an entry for which there is no corresponding static tuple in rIAPSITE.
Possible Error Effect:	There is an invalid entry in the AP link status table.
Corrective Action Taken:	The AP link status table entry is removed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PSaplst[] entry.
<i>Bad Data</i> =	Site ID of PSaplst[] entry.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Index of PSaplst[] entry.
<i>Dump</i> =	PSaplst[] entry.

Table 3 Error Code: INDEX

Error Code:	INDEX
Error Description:	The index field of the AP link status table entry does not agree with the index field in the static rIAPSITE tuple.
Possible Error Effect:	Users of the AP link status table could be queuing/dequeuing messages to the wrong AP.
Corrective Action Taken:	The AP index field in the AP link status table is changed to the value found in the static rIAPSITE tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PSaplst[] entry.
<i>Bad Data</i> =	AP index field of PSaplst[] entry.
<i>Good Data</i> =	AP index field of rIAPSITE tuple.
<i>Logical Index</i> =	Index of PSaplst[] entry.
<i>Dump 1</i> =	PSaplst[] entry.
<i>Dump 2</i> =	rIAPSITE tuple.

Table 4 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	An unused entry in the AP link status table has data in it other than zero.
Possible Error Effect:	Someone might be trying to use this entry in the AP link status table when they should not be doing so.
Corrective Action Taken:	The AP link status table entry is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PSaplst[] entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Index of PSaplst[] entry.
<i>Dump</i> =	PSaplst[] entry.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or ASSERTS.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

APSITE

Software Release: 5E10 and later

1. APPLICATIONS PROCESSOR SITE INDEX CONVERSION (APSITE) AUDIT

The relation RLAPSITE is defined in static data. It contains two attributes, site_id and apindex. This relation provides a translation between the site identification number of the applications processor (AP) and the AP index for switch use. There is a dynamic equivalent to RLAPSITE. The link status table structure has one and only one entry for every tuple in RLAPSITE. Each link status table entry contains fields in addition to an apsite and an apindx. The other fields give information on each of the DSL data links to an AP. The audit does not look at these other fields. Users of this other information do self auditing on the data. The APSITE audit uses the static driver (DRSTATIC) to traverse through the RLAPSITE relation tuple by tuple. For each tuple, a check is made to see that there exists a corresponding entry in the link status table. At the same time, a check is made to make sure that duplicate entries do not exist in the link status table. An entry is inserted into the link status table when a link is missing and duplicates are deleted.

2. DATA STRUCTURES

The RLAPSITE is a static relation for translation between the site ID and the AP index. The APSITE audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The APSITE audit insures that there exists one and only one entry for every rIAPSITE tuple. The static driver (DRSTATIC) is used to step through the RLAPSITE relation tuple by tuple. The audit is entered from audit control. Here are the actions performed for each tuple:

- A loop is performed through the link status table array.
- If two or more link status table entries contain the same apsite as the rIAPSITE tuple, then a duplicate tuple(s) has been found.
- If the end of the loop is reached, and a link status table entry with the same apsite as found in the rIAPSITE tuple has not been encountered, there is a missing link status table entry.

4. ERROR HANDLING

Error correction is performed in-line within the static to dynamic function. The possible error and recovery processes are:

- The duplicate tuple error is reported, and then recovery zeroes out all link status table entries after the first matching one is found.
- Missing link status table entries are inserted into the table.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DUPTUP

Error Code:	DUPTUP
Error Description:	The AP link status table has a duplicate entry.

Possible Error Effect:	There is an invalid entry in the AP link status table.
Corrective Action Taken:	The duplicate AP link status table entry is removed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PSaplst[] entry.
<i>Bad Data</i> =	Site ID of PSaplst[] entry.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Index of PSaplst[] entry.
<i>Dump 1</i> =	PSaplst[] entry.
<i>Dump 2</i> =	rlAPSITE tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The AP link status table has a missing entry.
Possible Error Effect:	The given AP link is not being used.
Corrective Action Taken:	An AP link status table entry is inserted according to the static rlAPSITE tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Site ID of rlAPSITE tuple.
<i>Logical Index</i> =	Site ID of rlAPSITE tuple.
<i>Dump</i> =	rlAPSITE tuple.

Table 3 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	An attempt to insert an AP link status table entry failed because there was insufficient room.
Possible Error Effect:	The given AP link cannot be used.
Corrective Action Taken:	After the APLST audit runs, there will be room to insert the AP link status table entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Site ID of rlAPSITE tuple.
<i>Logical Index</i> =	Site ID of rlAPSITE tuple.
<i>Dump</i> =	rlAPSITE tuple.

APSPB

Software Release: 5E10 and later

1. APPLICATIONS PROCESSOR SUPPLEMENTARY PROTOCOL BLOCK (APSPB) AUDIT

The applications processor supplementary protocol block (APSPB) audit performs some checks on the dynamic relation RLAPSPB. The tuples of RLAPSPB contain data to provide X.25 protocol for level 3 processing in the switching module (SM). The APSPB audit uses the indirect access sequential key driver (DRINASEK) to traverse through the RLAPSPB relation tuple by tuple. The key and quarantine bit are checked for each tuple. Also the linkage to rIDALB is verified. The linkages to the busy and idle lists (rIDB_HEADC tuples) are audited. If a given rIAPSPB tuple is found on the idle list, then the audit makes sure it is not linked to an rIDALB tuple. The audit does not look at the other attributes in the tuple. Users of this other information do self auditing on the data.

2. DATA STRUCTURES

The RLAPSPB is the dynamic relation containing data to provide X.25 protocol for level 3 processing in the SMP. The RLDALB is the D-channel application linkage block relation (information for routine messages from D-Channel to terminal process). The APSPB audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The APSPB audit insures that the key in each rIAPSPB tuple is correct and a check of the quarantine bit is made. Also the linkages to rIDALB and rIDB_HEADC are validated. The indirect key sequential driver (DRINASEK) is used to step through the rIAPSPB relation tuple by tuple. Here are the steps performed for each tuple:

- Checks are performed for a quarantine error.
- A test is performed on the key of the rIAPSPB tuple.
- If the tests pass, the linkages to rIDALB and rIDB_HEADC are validated.

A semantic check is performed to see if the idle rIAPSPB is not linked to rIDALB.

4. ERROR HANDLING

If an error is found, the DRINASEK driver zeroes out the tuple and put the key back in. The tuple is zeroed out and returned to the DBAPSPBIDLE list by the driver.

DRINASEK also purges the terminal process associated with this rIAPSPB tuple. DRINASEK traces back the linkages from rIAPSPB to rIDALB and so on until the terminal process is found. The driver knows how to do this by looking at the per relation information for the RLAPSPB relation. Note that if the APSPB audit has found a linkage error, it is unable to traverse the linkages back to the terminal process. However, one of the related audits would find the problem and would purge the terminal process.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The rIAPSPB tuple is not either the busy or the idle list.
Possible Error Effect:	The rIAPSPB tuple is a lost resource.

Corrective Action Taken:	The rAPSPB tuple is linked back up to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rAPSPB tuple.
<i>Bad Data =</i>	Physical rIDB HEADC key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rAPSPB logical key.
<i>Dump 1 =</i>	rAPSPB tuple.
<i>Dump 2 =</i>	rIDB HEADC tuple.

Table 2 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	The rAPSPB tuple is on the idle list, but the rIDALB tuple is still linked to it.
Possible Error Effect:	The rIDALB tuple is a lost resource.
Corrective Action Taken:	The rAPSPB tuple is unlinked from the rIDALB tuple and the DALB audit is scheduled as related to clean up the rest.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rAPSPB link to rIDALB.
<i>Bad Data =</i>	Physical rIDALB key linked to rAPSPB.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	rAPSPB logical key.
<i>Dump 1 =</i>	rAPSPB tuple.
<i>Dump 2 =</i>	rIDALB tuple.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rAPSPB tuple is on the busy list, which indicates that there should be a live terminal process for the AP port, but there is no rIDALB tuple linked to it.
Possible Error Effect:	The rAPSPB tuple is a lost resource.
Corrective Action Taken:	The rAPSPB tuple is linked back up to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rAPSPB link to rIDALB.
<i>Bad Data =</i>	Physical rIDALB key linked to rAPSPB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rAPSPB logical key.
<i>Dump =</i>	rAPSPB tuple.

APXCTBL

Software Release: 5E10 and later

1. AUTOPLEX[®] SYSTEM PROCESS CALL TABLE (APXCTBL) AUDIT

The AUTOPLEX[®] system process (APX_SP) is a system process which is designed for handling messages from the executive control processor complex (ECPC) when there is no APX terminal process (APX_TP) currently active on the call. Such a situation can happen, for example, in a trunk-to-trunk call. In those cases in which the APX_TP gets removed from a call, the call is "folded" into an APX_SP. The APX_SP then has the task of handling any ECPC messages which come in for this call, and it has the task of informing the ECPC of any change in status of the call. The APX_SP maintains a table associated with each call that it is monitoring, and when a stimulus message comes in for one of the calls it is monitoring, the APX_SP will communicate with RTA to create an AUTOPLEX[®] terminal process to handle the message.

The AUTOPLEX[®] system process call table (APXCTBL) contains data associated with each call it is either monitoring or involved with in a transformation. An APX_SP will become the owner of the call when the call is in an active/stable state, there is no cell site trunk (CST) or peripheral manipulation required, and only two non-CSTs are involved in the call.

When the APX_SP is in control of the call there is no port associated with the process and there is no direct control of the path between the two trunks involved in the call being monitored by the APX_SP.

The purpose of the APXCTBL audit is to ensure the integrity of the semantic data and linkages of the rAPXCTBL tuples in the RLAPXCTBL relation. The APXCTBL audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in segmented mode. It is also called during selective init.

2. DATA STRUCTURES

The RLAPXCTBL dynamic relation contains indices to two rGPAXDB tuples, tags for timers, a subordinate process identifier, and the custat. At this point in the APX call there is no rIPCBLA tuple, so all pertinent information is kept in either rAPXCTBL tuple or rGPAXDB tuple.

The APXCTBL audit references the following dynamic relations:

RLAPXHTBL	= This audit performs semantic checks on the dynamic relation RLAPXHTBL. Tuples exist in this relation only if they contain a valid index to an rAPXCTBL tuple.
RLDB_HEADC	= Two tuples of rIDB_HEADC act as the idle and busy list headcells for relation RLAPXCTBL, linked through <i>idle_act.st_ownk</i> .
RLGPAXDB	= A busy rAPXCTBL tuple is linked through the semantic linkages <i>gpaxidx1</i> and <i>gpaxidx2</i> to two rGPAXDB tuples.

3. AUDIT CHECKS

The APXCTBL audit makes calls to two audit drivers. It calls the DRINASEK audit driver to check the rAPXCTBL tuples and the DRSTATIC audit driver to check the rAPXHTBL tuples.

For each rAPXCTBL tuple found in the relation, the audit makes the following checks:

NOTIDLE	= If the rAPXCTBL tuple is idle, the customer state must be null, and both semantic links to the rGPAXDB tuples must be zero.
TIMESTAMP	= The difference between the last time the APXCTBL audit was run and the current time must

not be zero or less.

WRONGLIST = The rAPXCTBL tuple must be attached to either the busy or idle list.

If the rAPXCTBL tuple is busy:

BACKLINK1/2 = The rGPAXDB tuple (rGPAXDB.spidx) must point to this rAPXCTBL tuple.

DEADPID = If a subordinate process exists (rAPXCTBL.sub_pid), it must still be an active process.

INVOWNER = An rGPAXDB tuple whose semantic link in the rAPXCTBL tuple is gpaxidx1 (rAPXCTBL.gpaxidx1 = rGPAXDB.gpaxdb) must have a *rGPAXDB.client_tag* of CL_APX1; if the semantic link is in gpaxidx2 (rAPXCTBL.gpaxidx2 = rGPAXDB.gpaxdb), the client_tag must be CL_APX2.

KEYFUNC = The semantic link to rGPAXDB in the rAPXCTBL tuple (rAPXCTBL.gpaxidx2) must be equal to the key of the rGPAXDB tuple.

NOTBUSY = If the rAPXCTBL tuple is busy, the customer state must not be null.

STUCK = The rAPXCTBL tuple must not be in a transient state (*custat*) for more than 5 minutes.

UNEQUAL = The semantic link to rGPAXDB in the rAPXCTBL tuple (rAPXCTBL.gpaxidx1) must be equal to the key of the rGPAXDB tuple.

ZERO_LINK = If the rAPXCTBL tuple is busy, both semantic links to rGPAXDB must be non-zero.

For each rAPXHTBL tuple found in the relation, the audit makes the following checks:

CUSTAT = The customer state in the rAPXCTBL tuple must not be null.

INV = The key of the rAPXCTBL tuple must be equal to its semantic link (rAPXHTBL.idx) in the rAPXHTBL tuple.

IS_NOTUSED = The semantic link to the rAPXCTBL in the rAPXHTBL tuple must not be zero.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The system process index in the rGPAXDB tuple does not match the key of the rAPXCTBL tuple.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rAPXCTBL tuples.
Corrective Action Taken:	The rAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rAPXCTBL tuple where the error was found.

<i>Bad Data</i> =	The semantic link to the rIGPAXDB tuple (rIAPXCTBL.gpaxidx1).
<i>Good Data</i> =	The semantic link to the rIAPXCTBL tuple (rIGPAXDB.spidx).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump 1</i> =	The rIAPXCTBL tuple where the error was found.
<i>Dump 2</i> =	The rIGPAXDB tuple where the error was found.

Table 2 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The system process index in the rIGPAXDB tuple does not match the key of the rIAPXCTBL tuple.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIAPXCTBL tuples.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The semantic link to the rIGPAXDB tuple (rIAPXCTBL.gpaxidx2).
<i>Good Data</i> =	The semantic link to the rIAPXCTBL tuple (rIGPAXDB.spidx).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump 1</i> =	The rIAPXCTBL tuple where the error was found.
<i>Dump 2</i> =	The rIGPAXDB tuple where the error was found.

Table 3 Error Code: CUSTAT

Error Code:	CUSTAT
Error Description:	The hash table tuple rIAPXHTBL points to an rIAPXCTBL tuple that is no longer active (rIAPXCTBL.custat = FCC NULL).
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIAPXHTBL tuples.
Corrective Action Taken:	The rIAPXHTBL tuple is deleted, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXHTBL tuple where the error was found.
<i>Bad Data</i> =	The customer state of the rIAPXCTBL tuple (rIAPXCTBL.custat).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIAPXHTBL tuple (rIAPXCTBL.idx).
<i>Dump 1</i> =	The rIAPXCTBL tuple where the error was found.
<i>Dump 2</i> =	The rIAPXHTBL tuple where the error was found.

Table 4 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The subordinate process ID (rIAPXCTBL.sub_pid) is not null, but is not active.
Possible Error Effect:	This resource may not be needed any longer for the call.
Corrective Action Taken:	The subordinate process ID is set to null, and the audit continues.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The contents of the <i>sub_pid</i> structure in the rIAPXCTBL tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 5 Error Code:

Error Code:	INV
Error Description:	The key of the rIAPXCTBL tuple (rIAPXCTBL.idx) must be equal to its semantic link in the rIAPXHTBL tuple (rIAPXHTBL.idx).
Possible Error Effect:	Corruption has occurred. This may cause possible call failures if the system exhausts the number of available rIAPXHTBL tuples.

Corrective Action Taken:	The rIAPXHTBL tuple is deleted, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXHTBL tuple where the error was found.
<i>Bad Data</i> =	The semantic link to the rIAPXCTBL tuple (rIAPXHTBL.idx).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIAPXHTBL tuple (rIAPXHTBL.pid.procno).
<i>Dump</i> =	The rIAPXHTBL tuple where the error was found.

Table 6 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	An rIGPAXDB tuple whose semantic link in the rIAPXCTBL tuple is gpaxidx1 (rIAPXCTBL.gpaxidx1 = rIGPAXDB.gpaxdb) does not have a <i>rIGPAXDB.client_tag</i> of CL_APX1 if the semantic link is in gpaxidx2 (rIAPXCTBL.gpaxidx2 = rIGPAXDB.gpaxdb), the client_tag is not CL_APX2.
Possible Error Effect:	Corruption has occurred, causing possible call failures.
Corrective Action Taken:	The rIAPXCTBL tuple is deleted, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The client of the rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Good Data</i> =	Either CL_APX1 (for <i>rIAPXCTBL.gpaxidx1</i>) or CL_APX2 (for <i>rIAPXCTBL.gpaxidx1</i>).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (<i>rIAPXCTBL.idx</i>).
<i>Dump 1</i> =	The rIAPXCTBL tuple where the error was found.
<i>Dump 2</i> =	The rIGPAXDB tuple where the error was found.

Table 7 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	There is no connection to the APX system process call table (rIAPXHTBL.idx = 0).
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIAPXHTBL tuples.
Corrective Action Taken:	The rIAPXHTBL tuple is deleted, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXHTBL tuple where the error was found.
<i>Bad Data</i> =	The semantic link to the rIAPXCTBL (rIAPXHTBL.idx).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIAPXHTBL tuple (rIAPXHTBL.pid.procno).
<i>Dump</i> =	The rIAPXHTBL tuple where the error was found.

Table 8 Error Code: KEYFUNC

Error Code:	KEYFUNC
Error Description:	The key of the rIGPAXDB tuple (rIGPAXDB.gpaxdb) must be equal to its semantic link in the rIAPXCTBL tuple (rIAPXCTBL.gpaxidx2).
Possible Error Effect:	Corruption has occurred. This may cause possible call failures if the system exhausts the number of available rIAPXCTBL tuples.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The key to the rIGPAXDB tuple (rIGPAXDB.gpaxdb).
<i>Good Data</i> =	The semantic link to the rIGPAXDB tuple (rIAPXCTBL.gpaxidx2).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 9 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIAPXCTBL tuple is linked to its busy headcell but its customer state is null.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIAPXCTBL tuples.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The customer state (rIAPXCTBL.custat).
<i>Good Data</i> =	The linkage to the busy head cell (rIAPXCTBL.idle_actv.st_ownk).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 10 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIAPXCTBL tuple is linked to its idle headcell but its customer state is not null, or one of the semantic linkages to the rGPAXDB tuples is not zero.
Possible Error Effect:	If the rIAPXCTBL tuple is taken from the idle list and used in a call without initializing one of these fields, the bad data may bring that call down.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The customer state (rIAPXCTBL.custat).
<i>Good Data</i> =	The linkage to the idle head cell (rIAPXCTBL.idle_actv.st_ownk).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 11 Error Code: STUCK

Error Code:	STUCK
Error Description:	The customer state has been stuck in a transient state for over five minutes.
Possible Error Effect:	This may cause possible call failures if the system exhausts the number of available rIAPXCTBL tuples.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The customer state (rIAPXCTBL.custat).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 12 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The Operating Systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUapx_time, used for transient timing may have occurred.
Possible Error Effect:	The audit's transient timing has been interrupted. Lost resources may take longer to recover because transient timing must be restarted.
Corrective Action Taken:	AUapx_time will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the transient timing global variable (AUapx_time).
<i>Bad Data</i> =	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Content of the transient timing variable (AUapx_time).
<i>Dump 2</i> =	Content of the current OSDS ITIME (OSnow).

Table 13 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The key of the rIGPAXDB tuple (rIGPAXDB.gpaxdb) must be equal to its semantic link in the rIAPXCTBL tuple (rIAPXCTBL.gpaxidx1).
Possible Error Effect:	Corruption has occurred. This may cause possible call failures if the system exhausts the number of available rIAPXCTBL tuples.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The key to the rIGPAXDB tuple (rIGPAXDB.gpaxdb).
<i>Good Data</i> =	The semantic link to the rIGPAXDB tuple (rIAPXCTBL.gpaxidx1).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 14 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rIAPXCTBL tuple is not linked to its idle headcell nor to its busy headcell. It must be linked to one or the other.
Possible Error Effect:	The rIAPXCTBL tuple is a lost resource, unusable by call processing.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The link to the idle / busy headcell (rIAPXCTBL.idle_actv.st_ownk).
<i>Good Data</i> =	The value of the key to the idle head cell (DBAPXCTIDLE).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

Table 15 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	One or both of the semantic linkages to rIGPAXDB tuples is zero (rIAPXCTBL.gpaxidx1 and gpaxidx2).
Possible Error Effect:	This call is corrupt. Without the rIGPAXDB tuples, there is no knowledge of the far parties.
Corrective Action Taken:	The rIAPXCTBL tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIAPXCTBL tuple where the error was found.
<i>Bad Data</i> =	The semantic link to one rIGPAXDB tuple (rIAPXCTBL.gpaxidx1).
<i>Good Data</i> =	The semantic link to the other rIGPAXDB tuple (rIAPXCTBL.gpaxidx2).
<i>Logical Index</i> =	The key to the rIAPXCTBL tuple (rIAPXCTBL.idx).
<i>Dump</i> =	The rIAPXCTBL tuple where the error was found.

7. AQ - B

ATCALL

Software Release: 5E10 and later

1. AUTOCALL (ATCALL) AUDIT

The relation RLATCALL contains the auto call data for message service system (MSS) and electronic directory service (EDS). The ATCALL audit runs in a segmented mode and is in the switching modules.

2. DATA STRUCTURES

The ATCALL audit checks rLATCALL tuples to verify that the tuples are linked to the proper busy or idle list in RLDB_HEADC, and that busy tuples are linked to an active rITEDB tuple. The ATCALL audit uses the following fields in the RLATCALL relation to perform these checks:

quarstate	= The tuple's quarantine state bit.
in_use	= The tuple's busy bit.
busyidle.st_ownk	= The tuple's owner linkage to the RLDB_HEADC head cell owner tuple (DBAC_IDLE, idle list, or one of the two busy lists, DBDCALL or DBLGCALL).
vport	= The tuple's semantic linkage to a rITEDB tuple, whose rITEDB.owndalb linkage must not be null.

3. AUDIT CHECKS

The following checks are performed by the ATCALL audit:

NOTIDLE	= Verify that rLATCALL.in_use is DBNO when the tuple is on the idle list.
NOTBUSY	= Verify that rLATCALL.in_use is DBYES when the tuple is on the busy list.
MISSTUP	= Verify that a busy rLATCALL tuple is linked to a valid, non-zero rITEDB tuple.
DALBLINK	= Verify that the rITEDB tuple is associated with one or more rIDALB tuples.
IDLE_LINK	= Verify that the rLATCALL tuple is linked to either the busy or idle list in rIDB_HEADC.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	The rLATCALL tuple is invalidly marked busy. It is linked to a rITEDB tuple which is not associated with a call (rITEDB.owndalb.st_ownk is null), and is not validly busy.
Possible Error Effect:	Possible exhaustion of rLATCALL tuples.
Corrective Action Taken:	The rLATCALL tuple is recovered to an idle condition (zeroed out and relinked to the RLDB_HEADC idle list head cell).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the rIATCALL tuple.
<i>Bad Data</i> =	The linkage to the rIDALB tuple in the rITEDB tuple (rITEDB.owndalb.st_headk).
<i>Good Data</i> =	The semantic link from the rIATCALL tuple to the rITEDB tuple (rIATCALL.vport).
<i>Logical Index</i> =	The key of the rIATCALL tuple (atcall_ent).
<i>Dump</i> =	The contents of the rIATCALL tuple.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rIATCALL tuple is linked to neither the idle nor busy lists.
Possible Error Effect:	Possible exhaustion of rIATCALL tuples.
Corrective Action Taken:	The rIATCALL tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIATCALL tuple.
<i>Bad Data</i> =	The rIATCALL tuple's invalid linkage to RLDB_HEADC (rIATCALL.busyidle.st_ownk).
<i>Good Data</i> =	The value of the idle list in RLDB_HEADC that the rIATCALL tuple is recovered to (DBAC_IDLE).
<i>Logical Index</i> =	The key of the rIATCALL tuple (atcall_ent).
<i>Dump</i> =	The contents of the rIATCALL tuple.

Table 3 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	An rIATCALL tuple marked as busy is not linked correctly to a rITEDB tuple, therefore the tuple is not validly busy.
Possible Error Effect:	Possible exhaustion of rIATCALL tuples.
Corrective Action Taken:	The rIATCALL tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIATCALL tuple.
<i>Bad Data</i> =	The semantic link from the rIATCALL tuple to the rITEDB tuple (rIATCALL.vport).
<i>Good Data</i> =	The busy bit in the rIATCALL tuple (rIATCALL.in_use).
<i>Logical Index</i> =	The key of the rIATCALL tuple (atcall_ent).
<i>Dump</i> =	The contents of the rIATCALL tuple.

Table 4 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIATCALL tuple is linked to one of the busy lists in RLDB_HEADC but the busy attribute in rIATCALL (rIATCALL.in_use) is not set, therefore it should not be a busy tuple.
Possible Error Effect:	Possible exhaustion of rIATCALL tuples.
Corrective Action Taken:	The rIATCALL tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIATCALL tuple.
<i>Bad Data</i> =	The busy bit in the rIATCALL tuple (rIATCALL.in_use).
<i>Good Data</i> =	The rIATCALL tuple's linkage to RLDB_HEADC (rIATCALL.busyidle.st_ownk).
<i>Logical Index</i> =	The key of the rIATCALL tuple (atcall_ent).
<i>Dump</i> =	The contents of the rIATCALL tuple.

Table 5 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIATCALL tuple is linked to the idle list in RLDB_HEADC but the busy

	attribute in rIATCALL (rIATCALL.in_use) is set to DBYES.
Possible Error Effect:	Call Processing may think the tuple has useful information, but at any point it could be taken from the idle list and used by a different call.
Corrective Action Taken:	The rIATCALL tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIATCALL tuple.
<i>Bad Data</i> =	The busy bit in the rIATCALL tuple (rIATCALL.in_use).
<i>Good Data</i> =	The rIATCALL tuple's linkage to RLDB HEADC (rIATCALL.busyidle.st_ownk).
<i>Logical Index</i> =	The key of the rIATCALL tuple (atcall_ent).
<i>Dump</i> =	The contents of the rIATCALL tuple.

ATPDB

Software Release: 5E10 and later

1. ANNOUNCEMENT TERMINAL PROCESS DATA BLOCK (ATPDB) AUDIT

Design alternatives for operator actions (OA) data included using the stack or creating a new area or areas for them and linking them from the position data block. In an effort to reduce the number of stacks required for Operator Services Position System (OSPS) purposes, the call and announcement data for the automated announcement portion of an OSPS call is stored in the announcement terminal process data block relation (RLATPDB). The ATPDB audit is part of the OSPS audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The audit reads the tuples of the RLATPDB relation, a dynamic relation in the SM. The RLATPDB relation is owned by the dynamic data base head cells relation (RLDB_HEADC). Each tuple of RLATPDB is linked either to the busy or idle list. These data structures are shown in Figure 1.

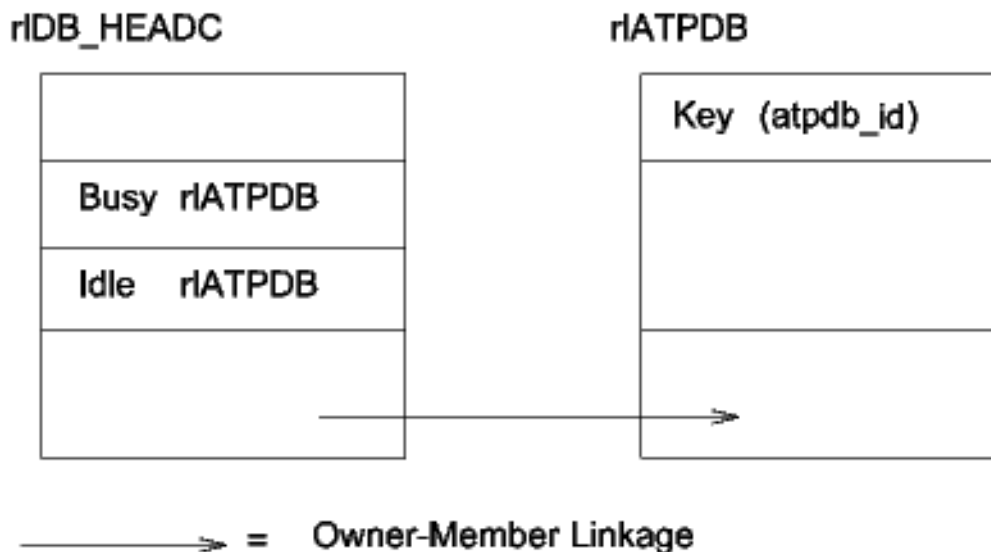


Figure 1 (ATPDB-1) Data Structures of the ATPDB Audit

3. AUDIT CHECKS

The audit calls the sequential index driver (DRINASEK) to check the rIATPDBs member linkages with DBHEADC and perform key checks. The driver invokes the semantic check function to perform the following checks:

- (a) The audit verifies that when the tuple is idle that it is owned by the DBHEADC idle list.
- (b) The audit verifies that when the tuple is busy that it is owned by the DBHEADC busy list.
- (c) The audit reports an error if the tuple is marked neither as busy nor idle.

4. ERROR HANDLING

The audit zeroes out erroneous rIATPDB tuples, reinitializes the keys, and puts them back on the correct idle list.

The ATPDB audit schedules no related audits.

The audit produces these error codes and those generated by the DRINASEK driver.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The bsy_idlls field is set to an invalid option.
Possible Error Effect:	The rIATPDB will be lost to the system.
Corrective Action Taken:	The corresponding rIATPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIATPDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIATPDB number.
<i>Dump =</i>	rIATPDB.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIATPDB is not busy when it should be.
Possible Error Effect:	The rIATPDB will be lost to the system.
Corrective Action Taken:	The corresponding rIATPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIATPDB.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	Expected owner's physical key.
<i>Logical Index =</i>	rIATPDB number.
<i>Dump =</i>	rIATPDB.

Table 3 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIATPDB is not idle when it should be.
Possible Error Effect:	The rIATPDB will be lost to the system.
Corrective Action Taken:	The corresponding rIATPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIATPDB.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	Expected owner's physical key.
<i>Logical Index =</i>	rIATPDB number.
<i>Dump =</i>	rIATPDB.

ATPMBUF-A

Software Release: 5E12 only

1. ETHERNET ESATP MESSAGE BUFFERS (ATPMBUF) AUDIT

Message buffers are used to store data that is being sent to an external system (such as, real time call detail manager) using an acknowledged protocol. The acknowledged transport protocol message buffers (ATPMBUFs) are used to represent per-message information about the data being sent to the external system. The ATPMBUFs are passed between data structures by linking them in different lists. The audit checks the linkages among CMatpmbuf[], CMatpxctrl[] and CMatpidlembuf and data within every entry in CMatpmbuf[].

2. DATA STRUCTURES

The following data are used by the ATPMBUF audit:

CMatpmbuf[] = ATP Message Buffers

CMatpxctrl[] = ATP Busy List

CMatpidlembuf = ATP Idle List

3. AUDIT CHECKS

The ATPMBUF audit makes the following checks:

BACKLINK1	= Check linkage of next field.
BACKLINK2	= Check linkage of previous field.
BADLINK	= Range check next field.
BADRANGE_INDEX	= Range check atpclient field.
BADSTATE	= Range check usage field.
BAD_LINK_DATA	= Range check previous field.
BUSYIDLE	= Check an entry that is allocated but not linked.
IDLE_LINK	= Check an in-use list entry.
IDLE_OWNER	= Check the idle list.
INV_STATE	= Verify values of state.
IS_NOTUSED	= Check an unused list entry.
MARKED_INDEX	= Check if entry on idle list.
NON_NULL	= Check an idle list entry.
ZAP	= Check for system interrupt handling.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The CMatpmbuff[] entry is the first member in a linked list of busy entries. One of the following conditions has occurred: • CMatpmbuff[index].usage.prev_mb is not within range, • CMatpmbuff[index].usage.next_mb is not within range, • CMatpmbuff[index].usage.next_mb equals CMatpmbuff[index].usage.prev_mb, • CMatpmbuff[index].usage.next_mb equals the index into CMatpmbuff[], • CMatpmbuff[index].usage.prev_mb equals the index into CMatpmbuff[], • CMatpmbuff[CMatpmbuff[index].usage.next_mb].usage.prev_mb does not equal the index into CMatpmbuff[], • CMatpmbuff[CMatpmbuff[index].usage.prev_mb].usage.next_mb does not equal the index into CMatpmbuff[], • CMatpmbuff[CMatpxctrl[CMatpmbuff[index].usage.(DMSHORT)atpclient].first_mb].usage.atpclient does not equal CMatpmbuff[index].usage.atpclient or • CMatpmbuff[CMatpxctrl[CMatpmbuff[index].usage.(DMSHORT)atpclient].last_mb].usage.atpclient does not equal CMatpmbuff[index].usage.atpclient
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuff).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpmbuff[index].usage.atpclient.
<i>Bad Data =</i>	CMatpmbuff[CMatpmbuff[index].usage.next_mb].usage.atpclient.
<i>Good Data =</i>	CMatpmbuff[CMatpmbuff[index].usage.next_mb].usage.prev_mb.
<i>Logical Index =</i>	Index into CMatpmbuff[].
<i>Dump =</i>	Contents of CMatpmbuff[index].usage.

Table 2 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The CMatpmbuff[] entry is the last member in a linked list of busy entries. One of the following conditions has occurred: • CMatpmbuff[index].usage.next_mb is not within range, • CMatpmbuff[index].usage.next_mb equals the index into CMatpmbuff[], • CMatpmbuff[CMatpmbuff[index].usage.next_mb].usage.prev_mb does not equal the index into CMatpmbuff[], • CMatpmbuff[CMatpmbuff[index].usage.next_mb].usage.atpclient does not equal CMatpmbuff[index].usage.atpclient or • CMatpxctrl[CMatpmbuff[index].usage.(DMSHORT)atpclient].first_mb does not equal the index into CMatpmbuff[].

Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpmbuf[index].usage.atpclient.
<i>Bad Data =</i>	CMatpmbuf[CMatpmbuf[index].usage.prev_mb].usage.atpclient.
<i>Good Data =</i>	CMatpmbuf[CMatpmbuf[index].usage.prev_mb].usage.next_mb.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 3 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The next field in the current CMatpmbuf[] entry is out of range.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpxctrl[(DMSHORT)CMatpmbuf[index].usage.atpclient].first_mb.
<i>Bad Data =</i>	CMatpxctrl[(DMSHORT)CMatpmbuf[index].usage.atpclient].last_mb.
<i>Good Data =</i>	CMatpmbuf[index].usage.next_mb.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 4 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	The value of the side of the busy list to which the current CMatpmbuf[] entry is linked is not within range (CMatpmbuf[index].usage.atpclient $\geq$ max).
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMatpmbuf[index].usage.atpclient.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 5 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The usetype of the current CMatpmbuf[] entry is not within range.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMatpmbuf[index].usage.usetype.
<i>Good Data =</i>	CMMAXATPMB.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 6 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The previous field in the current CMatpmbuf[] entry is out of range.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	CMatpxctrl[(DMSHORT)CMatpmbuf[index].usage.atpclient].first_mb.
<i>Bad Data</i> =	CMatpxctrl[(DMSHORT)CMatpmbuf[index].usage.atpclient].last_mb.
<i>Good Data</i> =	CMatpmbuf[index].usage.prev_mb.
<i>Logical Index</i> =	Index into CMatpmbuf[].
<i>Dump</i> =	Contents of CMatpmbuf[index].usage.

Table 7 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	This CMatpmbuf[] entry is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.toppt.
<i>Bad Data</i> =	CMatpmbuf[index].usage.usetype.
<i>Good Data</i> =	CMatpmbuf[index].usage.qindx.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	Contents of CMatpmbuf[index].usage.

Table 8 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An ATPMBUF in-use idle stack entry (index $\leq$ idle stack <i>toppt</i>) is corrupt. One or more of the following is true ("idle_atpmbuf" refers to the contents of the idle stack entry <i>mbindx[index]</i>): - idle_atpmbuf $\geq$ max - The ATPMBUF indexed the contents of idle stack entry <i>mbindx[idle_atpmbuf]</i> and does not point back to the idle list (ATPMBUF.mbindx $\neq$ idle_atpmbuf).
Possible Error Effect:	ATPMBUF may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry (index = idle stack <i>toppt</i>), the audit clears the idle stack entry (<i>mbindx[index]</i> = CMNIL_NDX) and decrements the count of idle ATPMBUFs (<i>toppt</i> is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets <i>mbindx[index]</i> = <i>sopcb[toppt]</i> , relinks (ATPMBUF.qindx = index), clears the old entry (<i>mbindx[CMatpedlembuf.toppt]</i> = CMNIL_NDX), and decrements the number of idle ATPMBUFs (decrements <i>toppt</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Indicates ATPMBUF idle list (CMatpidlembuf).
<i>Bad Data</i> =	Value from idle stack entry (ATPMBUF index).
<i>Good Data</i> =	value of "max".
<i>Logical Index</i> =	Index into idle stack.
<i>Dump</i> =	NA.

Table 9 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMatpmbuf[] idle stack is corrupt. One or more of the following is true: - The idle stack <i>toppt</i> value is out-of-range ($>$ max or $<$ min). - The idle stack entry <i>inpcb[toppt]</i> = CMNIL_NDX.

	The idle stack <i>toppt</i> is < max and the idle stack entry <i>inpcb[toppt+1] != CMNIL_NDX</i>.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit clears the CMatpmbuff[] idle stack (<i>toppt</i> = 0) and rebuilds the idle list (links all entries whose <i>inpcbstate</i> is CMINCBFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.mbindx[CMatpidlembuf.toppt]
<i>Bad Data</i> =	CMatpidlembuf.mbindx[CMatpidlembuf.toppt+1].
<i>Good Data</i> =	CM_NUMATPMBUF.
<i>Logical Index</i> =	CMatpidlembuf.toppt
<i>Dump</i> =	NA.

Table 10 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The usetype of a CMatpmbuff[] is one not recognized by the audit.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.toppt.
<i>Bad Data</i> =	CMatpmbuff[index].usage.usetype.
<i>Good Data</i> =	CM_MAXATPMB.
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 11 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused ATPMBUF idle stack entry is not null (index < <i>toppt</i> and <i>mbindx[index] != CMNIL_NDX</i>).
Possible Error Effect:	ATPMBUFs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the ATPMBUF idle stack entry (<i>mbindx[index] = CMNIL_NDX</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Indicate ATPMBUF idle list (CMatpidlembuf).
<i>Bad Data</i> =	Contents of the idle stack entry (<i>mbindx[index]</i>).
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into the idle stack.
<i>Dump</i> =	NA.

Table 12 Error Code: MARKED_INDEX

Error Code:	MARKED_INDEX
Error Description:	This CMatpmbuff[] entry should be linked to another CMatpmbuff[] entry on the CMatpmbuff[] busy list but the index into idle list is not its null value.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpmbuff[index].usage.qindx.
<i>Bad Data</i> =	CMatpmbuff[index].usage.next_mb.
<i>Good Data</i> =	CMatpmbuff[index].usage.prev_mb.
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 13 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	One of the fields in an idle CMatpmbuff[] entry is not its default value.

Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpidlembuf.toppt,
<i>Bad Data =</i>	CMatpmbuff[index].usage.qindx.
<i>Good Data =</i>	CMatpidlembuf.mbindx[CMatpmbuff[index].usage.qindx].
<i>Logical Index =</i>	Index into CMatpmbuff[].
<i>Dump =</i>	Contents of CMatpmbuff[index].usage.

Table 14 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle stack entry, if any, that was being audited when the stimulus occurred. • Recovers the CMatpmbuff[] entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	AUcursor.
<i>Dump =</i>	NA.

ATPMBUF-B

Software Release: 5E13 and later

1. ETHERNET ESATP MESSAGE BUFFERS (ATPMBUF) AUDIT

Message buffers are used to store data that is being sent to an external system (such as, real time call detail manager) using an acknowledged protocol. The acknowledged transport protocol message buffers (ATPMBUFs) are used to represent per-message information about the data being sent to the external system. The ATPMBUFs are passed between data structures by linking them in different lists. The audit checks the linkages among CMatpmbuf[], CMatpxctrl[] and CMatpidlembuf and data within every entry in CMatpmbuf[].

2. DATA STRUCTURES

The following data are used by the ATPMBUF audit:

CMatpmbuf[] = ATP Message Buffers
 CMatpxctrl[] = ATP Busy List
 CMatpidlembuf = ATP Idle List

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The CMatpmbuf[] entry is the first member in a linked list of busy entries. One of the following conditions has occurred: • CMatpmbuf[index].usage.prev_mb is not within range, • CMatpmbuf[index].usage.next_mb is not within range, • CMatpmbuf[index].usage.next_mb equals CMatpmbuf[index].usage.prev_mb, • CMatpmbuf[index].usage.next_mb equals the index into CMatpmbuf[], • CMatpmbuf[index].usage.prev_mb equals the index into CMatpmbuf[], • CMatpmbuf[CMatpmbuf[index].usage.next_mb].usage.prev_mb does not equal the index into CMatpmbuf[], • CMatpmbuf[CMatpmbuf[index].usage.prev_mb].usage.next_mb does not equal the index into CMatpmbuf[], • CMatpmbuf[CMatpxctrl[CMatpmbuf[index].usage.(DMSHORT)atpclient].first_mb].usage.atpclient does not equal CMatpmbuf[index].usage.atpclient or

	CMatpmbuf[CMatpxctrl[CMatpmbuf[index].usage.(DMSHORT)atpclient].last_mb].usage.atpclient does not equal CMatpmbuf[index].usage.atpclient.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpmbuf[index].usage.atpclient.
<i>Bad Data =</i>	CMatpmbuf[CMatpmbuf[index].usage.next_mb].usage.atpclient.
<i>Good Data =</i>	CMatpmbuf[CMatpmbuf[index].usage.next_mb].usage.prev_mb.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 2 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The CMatpmbuf[] entry is the last member in a linked list of busy entries. One of the following conditions has occurred: - CMatpmbuf[index].usage.next_mb is not within range, - CMatpmbuf[index].usage.next_mb equals the index into CMatpmbuf[], - CMatpmbuf[CMatpmbuf[index].usage.next_mb].usage.prev_mb does not equal the index into CMatpmbuf[], - CMatpmbuf[CMatpmbuf[index].next_mb].usage.atpclient does not equal CMatpmbuf[index].usage.atpclient or - CMatpxctrl[CMatpmbuf[index].usage.(DMSHORT)atpclient].first_mb does not equal the index into CMatpmbuf[].
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpmbuf[index].usage.atpclient.
<i>Bad Data =</i>	CMatpmbuf[CMatpmbuf[index].usage.prev_mb].usage.atpclient.
<i>Good Data =</i>	CMatpmbuf[CMatpmbuf[index].usage.prev_mb].usage.next_mb.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 3 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The next field in the current CMatpmbuf[] entry is out of range.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMatpxctrl[(DMSHORT)CMatpmbuf[index].usage.atpclient].first_mb.
<i>Bad Data =</i>	CMatpxctrl[(DMSHORT)CMatpmbuf[index].usage.atpclient].last_mb.
<i>Good Data =</i>	CMatpmbuf[index].usage.next_mb.
<i>Logical Index =</i>	Index into CMatpmbuf[].
<i>Dump =</i>	Contents of CMatpmbuf[index].usage.

Table 4 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	The value of the side of the busy list to which the current CMatpmbuf[] entry is

	linked is not within range (CMatpmbuff[index].usage.atpclient $\geq$ max).
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMatpmbuff[index].usage.atpclient.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 5 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The usetype of the current CMatpmbuff[] entry is not within range.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMatpmbuff[index].usage.usetype.
<i>Good Data</i> =	CMMAXATPMB.
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 6 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The previous field in the current CMatpmbuff[] entry is out of range.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpxctrl[(DMSHORT)CMatpmbuff[index].usage.atpclient].first_mb.
<i>Bad Data</i> =	CMatpxctrl[(DMSHORT)CMatpmbuff[index].usage.atpclient].last_mb.
<i>Good Data</i> =	CMatpmbuff[index].usage.prev_mb.
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 7 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	This CMatpmbuff[O] entry is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.toppt.
<i>Bad Data</i> =	CMatpmbuff[index].usage.usetype.
<i>Good Data</i> =	CMatpmbuff[index].usage.qindx.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 8 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A ATPMBUF in-use idle stack entry (index $\leq$ CMatpidlembuf.toppt) is corrupt. One or more of the following is true: CMatpidlembuf.mbindx[index] $\geq$ CM_NUMATPMBUF

	• CMatpidlembuf.mbindx[index] ≥ 0 • CMatpmbuf[CMatpidlembuf.mbindx[index]].usage.qindx != index into CMatpidlembuf.mbindx[].
Possible Error Effect:	ATPMBUF may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry (index = CMatpidlembuf.toppt), the audit clears the idle stack entry (CMatpidlembuf[index] = CMNIL_NDX) and decrements the count of idle ATPMBUFs (CMatpidlembuf.toppt is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets CMatpidlembuf.mbindx[index] = CMatpidlembuf.mbindx[CMatpidlembuf.toppt], relinks the ATPMBUF (CMatpmbuf[CMatpidlembuf.mbindx[CMatpidlembuf.toppt]].usage.qindx = index), clears the old entry (CMatpidlembuf.mbindx[CMatpidlembuf.toppt] = CMNIL_NDX) and decrements the number of idle ATPMBUFs (decrements CMatpidlembuf.toppt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUATPTYPE.
<i>Bad Data</i> =	Index of next available idle CMatpmbuf[].
<i>Good Data</i> =	CM_NUMATPMBUF.
<i>Logical Index</i> =	Index into CMatpidlembuf.mbindx[].
<i>Dump</i> =	NA.

Table 9 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMatpmbuf[] idle stack is corrupt. One or more of the following is true: • The idle stack CMatpidlembuf.toppt value is out-of-range (> max or < min). • The idle stack entry CMatpidlembuf.mbindx[CMatpidlembuf.toppt] = CMNIL_NDX. • The idle stack CMatpidlembuf.toppt is < max and the idle stack entry CMatpidlembuf.mbindx[CMatpidlembuf.toppt+1] != CMNIL_NDX.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit sets CMatpidlembuf.toppt = 0 and rebuilds the idle list (links all entries whose CMatpmbuf[].usage.usetype is CMFREEATPMB to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.mbindx[CMatpidlembuf.toppt]
<i>Bad Data</i> =	CMatpidlembuf.mbindx[CMatpidlembuf.toppt+1].
<i>Good Data</i> =	CM_NUMATPMBUF.
<i>Logical Index</i> =	CMatpidlembuf.toppt.
<i>Dump</i> =	NA.

Table 10 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The usetype of a CMatpmbuf[] is one not recognized by the audit.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.toppt.
<i>Bad Data</i> =	CMatpmbuf[index].usage.usetype.
<i>Good Data</i> =	CM_MAXATPMB.

<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 11 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused ATPMBUF idle stack entry (index > CMatpidlembuf.toppt) is not null (CMatpidlembuf.mbindx[index] != CMNIL_NDX and index < CMatpidlembuf.toppt).
Possible Error Effect:	ATPMBUFs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the ATPMBUF idle stack entry (CMatpidlembuf.mbindx[index] = CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUATPTYPE.
<i>Bad Data</i> =	Contents of the idle stack entry (CMatpidlembuf.mbindx[index]).
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into the idle stack.
<i>Dump</i> =	NA.

Table 12 Error Code: MARKED_INDEX

Error Code:	MARKED_INDEX
Error Description:	This CMatpmbuff[] entry should be linked to another CMatpmbuff[] entry on the CMatpxctrl[] busy list but the index into idle list is not its null value.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpmbuff[index].usage.qindx.
<i>Bad Data</i> =	CMatpmbuff[index].usage.next_mb.
<i>Good Data</i> =	CMatpmbuff[index].usage.prev_mb.
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 13 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	One of the fields in an idle CMatpmbuff[] entry is not its default value.
Possible Error Effect:	An RTCD billing record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMatpidlembuf).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMatpidlembuf.toppt.
<i>Bad Data</i> =	CMatpmbuff[index].usage.qindx.
<i>Good Data</i> =	CMatpidlembuf.mbindx[CMatpmbuff[index].usage.qindx].
<i>Logical Index</i> =	Index into CMatpmbuff[].
<i>Dump</i> =	Contents of CMatpmbuff[index].usage.

Table 14 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle stack entry, if any, that was being audited when the stimulus occurred. • Recovers the CMatpmbuff[] entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any

	other data structures, sets all fields to default values and puts the entry on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	NA.

ATTHBLF

Software Release: 5E10 and later

1. ATTENDANT CONSOLE LINK HEADER BUSY LAMP FIELD (ATTHBLF) AUDIT

The RLATTH_BLF relation contains data used to monitor a 100's group in a given switching module. The 100's group may be monitored by one or more attendant consoles. The ATTHBLF audit verifies the owner-member linkages between the RLATTHBLF (busy list) and RLATTMBLF relations or between the RLATTHBLF and the RLDBHEADC (idle list) relation. The ATTHBLF audit cycles through the RLATTH_BLF dynamic relation one tuple at a time using the DRINASEK audit driver. The ATTHBLF audit performs linkage and key checks on the tuple. If either of these checks fail, an error is printed and the RLATTH_BLF tuple is placed on the idle list.

The ATTHBLF audit is part of the ISDNBRCS package and resides in the AUatfeat module. This audit runs in the OSDS-M environment. Related audits are ATTMBLF and DBHEADC.

2. DATA STRUCTURES

The data structure for the ATTHBLF audit is RLATTH_BLF. There exists one tuple for each 100's group that is monitorable by the attendant console.

3. AUDIT CHECKS

The ATTHBLF audit verifies owner-member linkages between the RLATTHBLF and RLATTMBLF relations or between the RLATTHBLF and RLDBHEADC, the tuple must be linked to one of these relations.

4. ERROR HANDLING

The semantic function checks the owner of the rLATTHBLF tuple. If the owner is the idle rLDBHEADC, a check is made to determine if the set member field in the rLATTHBLF is zero; if not, an error is printed and the idle function called. This action triggers a return of the correct idle list to the driver for idling and the ATTMBLF and DB_HEADC audits will be scheduled.

If the setowner linkage is checked and the tuple is on the busy list, the head and tail keys are checked for non-zero. If either is equal to zero, an error is printed and the idle function is called; this returns the correct idle list to the driver for idling. The ATTMBLF and DB_HEADC audits will be scheduled.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	This tuple is not on the idle list, the owner linkage is zero but the member linkage is also zero.
Possible Error Effect:	The rLATTH_BLF tuple has been garbaged, you cannot trust the data in this tuple.
Corrective Action Taken:	Report the error on the ROP and return to the driver to put the tuple on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLATTH_BLF.
<i>Bad Data =</i>	Member linkage to rLATTH_BLF.
<i>Good Data =</i>	Owner linkage.
<i>Logical Index =</i>	Key of rLATTH_BLF.
<i>Dump =</i>	rLATTH_BLF tuple.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rIATTH_BLF is supposed to be idle but is not linked to the rIDBHEADC idle list.
Possible Error Effect:	The rIATTH_BLF is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rIATTH_BLF tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIATTH_BLF.
<i>Bad Data</i> =	Owner link of rIATTH_BLF.
<i>Good Data</i> =	Physical key of idle head cell.
<i>Logical Index</i> =	Key of rIATTH_BLF.
<i>Dump</i> =	rIATTH_BLF tuple.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rIATTH_BLF is in the idle state but some of its linkages involved in the idle list mapping are nonzero.
Possible Error Effect:	A source is linked to a rIATTH_BLF on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rIATTH_BLF tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIATTH_BLF.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rIATTH_BLF.
<i>Dump</i> =	rIATTH_BLF tuple.

ATTMBLF

Software Release: 5E10 and later

1. ATTENDANT CONSOLE LINK MEMBERS BUSY LAMP FIELD (ATTMBLF) AUDIT

The ATTMBLF audit verifies that a tuple within the RLATTM_BLF dynamic relation is properly allocated.

2. DATA STRUCTURES

The RLATTM_BLF relation is a dynamic, indirect access nonsequential key relation that is used to store integrated services digital network (ISDN) attendant console (AC) data for each attendant. The ISTERM and ATTH_BLF audits will be scheduled to run as related audits, upon detection of an error by the ATTMBLF audit. The ATTMBLF audit will run in the segmented and initialization modes and operate in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The ATTMBLF audit verifies the dynamic to static status for RLATTM_BLF. The ISTERM audit verifies the static to dynamic status for the same relation. For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The ATTMBLF audit invokes the drindex driver. The driver verifies that the RLATTM_BLF being checked has a corresponding entry in RLIS_TERM. If no entry is found, the driver removes the rLATTM_BLF being checked. If the existence check passes, the audit checks appropriate linkages as well as the key field. If any errors are found, the tuple being checked is removed, linked to the appropriate idle list, and semantic data is initialized. The related audits are scheduled. An rLATTH_BLF tuple serves as the idle list head cell for the RLATTM_BLF relation. When an rLATTM_BLF is busy, it is linked to a busy rLATTH_BLF tuple.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The audit is unable to get the owning rLATTH_BLF tuple for this rLATTM_BLF tuple.
Possible Error Effect:	The owner field in the rLATTM_BLF may be incorrectly populated. The incorrect data may result in the inadvertent loss of AC information.
Corrective Action Taken:	The audit will zero the tuple in question, get it initialized, and have it linked to an appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to rLATTM_BLF tuple.
<i>Bad Data =</i>	The index field of the thought-to-be owner rLATTH_BLF tuple.
<i>Good Data =</i>	The index field of rLATTH_BLF passed to data base read primitives.
<i>Logical Index =</i>	The key of the rLATTM_BLF tuple in question.
<i>Dump =</i>	rLATTM_BLF tuple in question; rLATTH_BLF tuple accessed.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The audit has detected an inconsistency between the groupno, and/or grptype, and/or member fields of the rLATTM_BLF tuple. These fields are used to access the related rIGROUP_PORT tuple.

Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; the member field of the rIATTM_BLF tuple being audited.
<i>Bad Data =</i>	Groupno field of rIATTM_BLF tuple.
<i>Good Data =</i>	Return value from database read for rIGROUP_PORT.
<i>Logical Index =</i>	latidx field of the rIATTM_BLF tuple in question.
<i>Dump =</i>	rIATTM_BLF in question; buffer used to access rIGROUP_PORT tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an extra rIATTM_BLF tuple. A relationship exists such that, for every rIATTM_BLF dynamic tuple, there should exist a corresponding rIGROUP_PORT static tuple where: rIGROUP_PORT.prtgrp.grpnum = rIATTM_BKF.groupno, rIGROUP_PORT.member = rIATTM_BLF.member, and rIGROUP_PORT.grptyp = TMLINE.
Possible Error Effect:	Attendant Console actions and operations may be disrupted if call processing attempts to access the unassigned attendant monitoring resources represented by the tuple.
Corrective Action Taken:	The rIATTM_BLF dynamic tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rIATTM_BLF tuple address.
<i>Bad Data =</i>	The logical key rIATTM_BLF tuple in question (rIATTM_BLF.latidx).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIATTM_BLF tuple in question (rIATTM_BLF.latidx).
<i>Dump =</i>	The rIATTM_BLF tuple in question.

Table 4 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The audit has detected an inconsistency between the hgsbn field and the hgid field in the rIATTM_BLF tuple. The hgid field indicates that a hundreds group id is displayed while the hgsbn field indicates that no hundreds group has been selected.
Possible Error Effect:	An incorrect hundreds group and/or hundreds group select could be indicated on an attendant console.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the value of the hgid field of the rIATTM_BLF tuple in question.
<i>Bad Data =</i>	The hgscpec field of the rIATTM_BLF tuple in question this field shows the currently displayed hundreds group select CPE component.
<i>Good Data =</i>	The hgsbn field of the rIATTM_BLF tuple in question.
<i>Logical Index =</i>	The key of the rIATTM_BLF tuple in question.
<i>Dump =</i>	The rIATTM_BLF tuple in question.

Table 5 Error Code: GRBG

Error Code:	GRBG
Error Description:	The audit has detected an inconsistency between the hgid field of the rIATTM_BLF tuple and the hgid field of the owning rIATTM_BLF tuple. They should be the same.
Possible Error Effect:	An incorrect hundreds group could be indicated on an attendant console.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked up to

	the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to the rATTM_BLF tuple.
<i>Bad Data =</i>	hgid field of the owning rATTM_BLF tuple.
<i>Good Data =</i>	hgid field of the rATTM_BLF tuple.
<i>Logical Index =</i>	The key field of the rATTM_BLF tuple.
<i>Dump =</i>	rATTM_BLF tuple in question; owning rATTM_BLF tuple.

Table 6 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The audit has detected an inconsistency between the hgsbn field and the hgid field in the rATTM_BLF tuple. The hgid field indicates that a hundreds group id is idle (not displayed) while the hgsbn field indicates that a hundreds group has been selected.
Possible Error Effect:	An incorrect hundreds group and/or hundreds group select could be indicated on an attendant console.
Corrective Action Taken:	The rATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the value of the hgid field of the rATTM_BLF tuple in question.
<i>Bad Data =</i>	The hgspec field of the rATTM_BLF tuple in question; this field shows the currently displayed hundreds group select CPE component.
<i>Good Data =</i>	The hgsbn field of the rATTM_BLF tuple in question.
<i>Logical Index =</i>	The key of the rATTM_BLF tuple in question.
<i>Dump =</i>	The rATTM_BLF tuple in question.

Table 7 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	The audit cannot access a related static rLAT_LPT tuple based on the latidx field of the dynamic rATTM_BLF tuple being audited.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rATTM_BLF tuple will be removed. If necessary, it will be reinserted, initialized, and linked to the appropriate idle list by the related ISTERM audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Local pointer to rATTM_BLF tuple being audited.
<i>Bad Data =</i>	latidx field of rATTM_BLF tuple being audited.
<i>Good Data =</i>	Return value from data base read of rLAT_LPT.
<i>Logical Index =</i>	latidx field of local buffer used to read rLAT_LPT.
<i>Dump =</i>	rATTM_BLF tuple being audited; local buffer used to access the related rLAT_LPT tuple.

Table 8 Error Code: LNK2STK

Error Code:	LNK2STK
Error Description:	The audit has found an inconsistency with the link2ip field. This field should always be DBNO when the audit gets to check it. It is supposed to be marked to DBYES and reset to DBNO by the operational code without taking time breaks. It is used by the operational code to determine that the second DALB link in the PCBLA is in use.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to the rATTM_BLF tuple.
<i>Bad Data =</i>	Link2ip field of the rATTM_BLF tuple.
<i>Good Data =</i>	DBNO.
<i>Logical Index =</i>	Key field of the rATTMBLF tuple in question.

<i>Dump</i> =	rIATTM_BLF tuple in question.

Table 9 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit has detected an inconsistency between the member field of the rIATTM_BLF tuple and the member field of the associated static data rIGROUP_PORT tuple. These fields indicate which member the attendant is, within an attendant group.
Possible Error Effect:	None.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rIATTM_BLF tuple.
<i>Bad Data</i> =	Member field of the rIATTM_BLF tuple.
<i>Good Data</i> =	Member field of the associated static rIGROUP_PORT tuple.
<i>Logical Index</i> =	Key field of the rIATTM_BLF tuple in question.
<i>Dump</i> =	rIATTM_BLF tuple; rIGROUP_PORT tuple.

Table 10 Error Code: MINDX

Error Code:	MINDX
Error Description:	The audit has detected an inconsistency between the isat_idx field of the rIATTM_BLF tuple and the associated static data rIRT_MHG tuple isat_idx field.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rIATTM_BLF tuple.
<i>Bad Data</i> =	isat_idx field of rIATTM_BLF tuple.
<i>Good Data</i> =	Related static data rIRT_MHG isat_idx field.
<i>Logical Index</i> =	Key field of the rIATTM_BLF tuple in question.
<i>Dump</i> =	rIATTM_BLF in question; related rIRT_MHG tuple.

Table 11 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The audit has detected an inconsistency between the latidx and/or the termno fields of the rIATTM_BLF tuple being audited.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIATTM_BLF tuple.
<i>Bad Data</i> =	Termno field of rIATTM_BLF tuple being audited.
<i>Good Data</i> =	Return value from data base read for rILAT_LPT.
<i>Logical Index</i> =	latidx field of the rIATTM_BLF tuple in question.
<i>Dump</i> =	rIATTM_BLF in question; buffer used to access rILAT_LPT tuple.

Table 12 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has detected an inconsistency between the static relations rIGROUP_PORT and rIRT_MHG.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	Check the rIRT_MHG tuple in question whose key is in the BAD DATA indicated below with the rIGROUP_PORT tuple whose key fields groupno and member come from the rIATTM_BLF tuple dumped. The grptype should be TM_LINE.
Dump Description:	

<i>Error Address =</i>	NA.
<i>Bad Data =</i>	grpnum field from rIGROUP_PORT used to access rIRT_MHG.
<i>Good Data =</i>	Return from data base read.
<i>Logical Index =</i>	Key field of the rIATTM_BLF tuple in question.
<i>Dump =</i>	The rIATTM_BLF tuple in question.

Table 13 Error Code: PORT

Error Code:	PORT
Error Description:	The audit has detected an inconsistency between the static relations rIGROUP_PORT and rLPT_PT.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	Check the rLPT_PT relation tuple whose key is from the rIGROUP_PORT relation tuple indicated.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	lport field used to access rLPT_PT.
<i>Good Data =</i>	Return from data base read.
<i>Logical Index =</i>	lport field from rIGROUP_PORT, used to access rLPT_PT.
<i>Dump =</i>	The rIATTM_BLF tuple in question; related rIGROUP_PORT tuple.

Table 14 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The audit has detected an inconsistency between the dport field in the rIATTM_BLF tuple and the port field of the related static data rLPT_PT tuple.
Possible Error Effect:	Port related attendant console activity may not operate properly.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to rIATTM_BLF tuple.
<i>Bad Data =</i>	dport field of rIATTM_BLF tuple.
<i>Good Data =</i>	Port field of related rLPT_PT tuple.
<i>Logical Index =</i>	Key field of the rIATTM_BLF tuple in question.
<i>Dump =</i>	The rIATTM_BLF tuple in question; related rLPT_PT tuple.

Table 15 Error Code: TEDBATR

Error Code:	TEDBATR
Error Description:	The rIATTM_BLF tuple's tedbkey field does not match data obtained from the corresponding rIGROUP_PORT static tuple. The rIATTM_BLF tuple's tedbkey field should be the same as the virtual port (VPORT) calculated from the corresponding rIGROUP_PORT static tuple for this rIATTM_BLF dynamic tuple by the following algorithm: $rIATTM\_BLF.tedbkey = ((rIGROUP\_PORT.lp\_type \ll BFLPORT) \mid rIGROUP\_PORT.lport)$ where $rLPORT\_GROUP.grpnum = rIATTM\_BLF.groupno,$ $rIGROUP\_PORT.member = rIATTM\_BLF.member, \text{ and}$ $rIGROUP\_PORT.grptype = TM\_LINE.$
Possible Error Effect:	No logical mapping between the Attendant and the virtual port (represented by an rTEDB tuple where rTEDB.tedb = rIATTM_BLF.tedbkey) exists. Attendant initiated actions may not be set up correctly, and attendant console button indicators may reflect inaccurate status.
Corrective Action Taken:	The rIATTM_BLF tuple will be initialized and linked to the appropriate RLATTH_BLF idle list head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIATTM_BLF tuple.

<i>Bad Data</i> =	The RLATTM_BLF tedbkey field as found.
<i>Good Data</i> =	The calculated RLATTM_BLF tedbkey value from rIGROUP_PORT.lport and rIGROUP_PORT.lpt_type.
<i>Logical Index</i> =	Key of rIATTM_BLF tuple (rIATTM_BLF.latidx).
<i>Dump 1</i> =	rIATTM_BLF tuple.
<i>Dump 2</i> =	related rIGROUP_PORT static tuple.

Table 16 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The audit has previously detected an error and must call an initialization function as part of recovery. Prior to calling the initialization function, the pointer to the related tedb must be found. This error indicates that the pointer to the tedb was not obtained. A related tedb audit will be scheduled for recovery.
Possible Error Effect:	Attendant initiated actions may not be set up correctly.
Corrective Action Taken:	The TEDB audit will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from DA primitive to get TEDB pointer.
<i>Good Data</i> =	Port that the related TEDB is for.
<i>Logical Index</i> =	Key field of the rIATTM_BLF tuple in question.
<i>Dump</i> =	NA.

Table 17 Error Code: UNASSIGNED

Error Code:	UNASSIGNED
Error Description:	The audit has detected an inconsistency between the sddi field and the str_param field in the rIATTM_BLF tuple. The sddi field indicates that there is no active transfer request while the str_param field indicates there is an active transfer request.
Possible Error Effect:	The attendant may try to make transfers that don't exist.
Corrective Action Taken:	The rIATTM_BLF tuple in question will be zeroed, initialized, and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rIATTM_BLF tuple.
<i>Bad Data</i> =	sddi field of rIATTM_BLF tuple.
<i>Good Data</i> =	str_param field of rIATTM_BLF tuple.
<i>Logical Index</i> =	Key field of the rIATTM_BLF tuple in question.
<i>Dump</i> =	The rIATTM_BLF tuple in question.

AUTOQ

Software Release: 5E10 and later

1. AUTOMATIC QUEUE (AUTOQ) AUDIT

The maintenance request administrator (MRA) program runs four diagnostic processes simultaneously and queues the rest. Five test processes may be run simultaneously also. The automatic queue relation (RLAUTOQ) contains an entry for all queued diagnostic and test processes. For every entry in the RLAUTOQ relation, there is a corresponding entry in the switch maintenance cross-reference table relation (RLSMXRF), which lists all the currently active processes controlled by MRA. The automatic queue audit (AUTOQ) checks the linkage between the automatic queue relation (RLAUTOQ) and its owner, the database head cells relation (RLDB_HEADC).

This audit is part of the switch maintenance audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structure checked by the AUTOQ audit is the automatic queue relation (RLAUTOQ), a dynamic relation in the switching modules (SMs). The RLAUTOQ relation may have up to eight entries, one for each active process. The RLAUTOQ relation is a member relation of the database head cells relation (RLDB_HEADC), a dynamic relation in the SMs. The RLDB_HEADC relation has one tuple for idle rIAUTOQ tuples, one tuple for automatic diagnostic request rIAUTOQ tuples, one tuple for manual diagnostic request rIAUTOQ tuples, and one tuple for test request rIAUTOQ tuples. The audit also checks the process ID field in RLSMEST, a dynamic relation in the SMs.

Figure 1 shows these data structures.

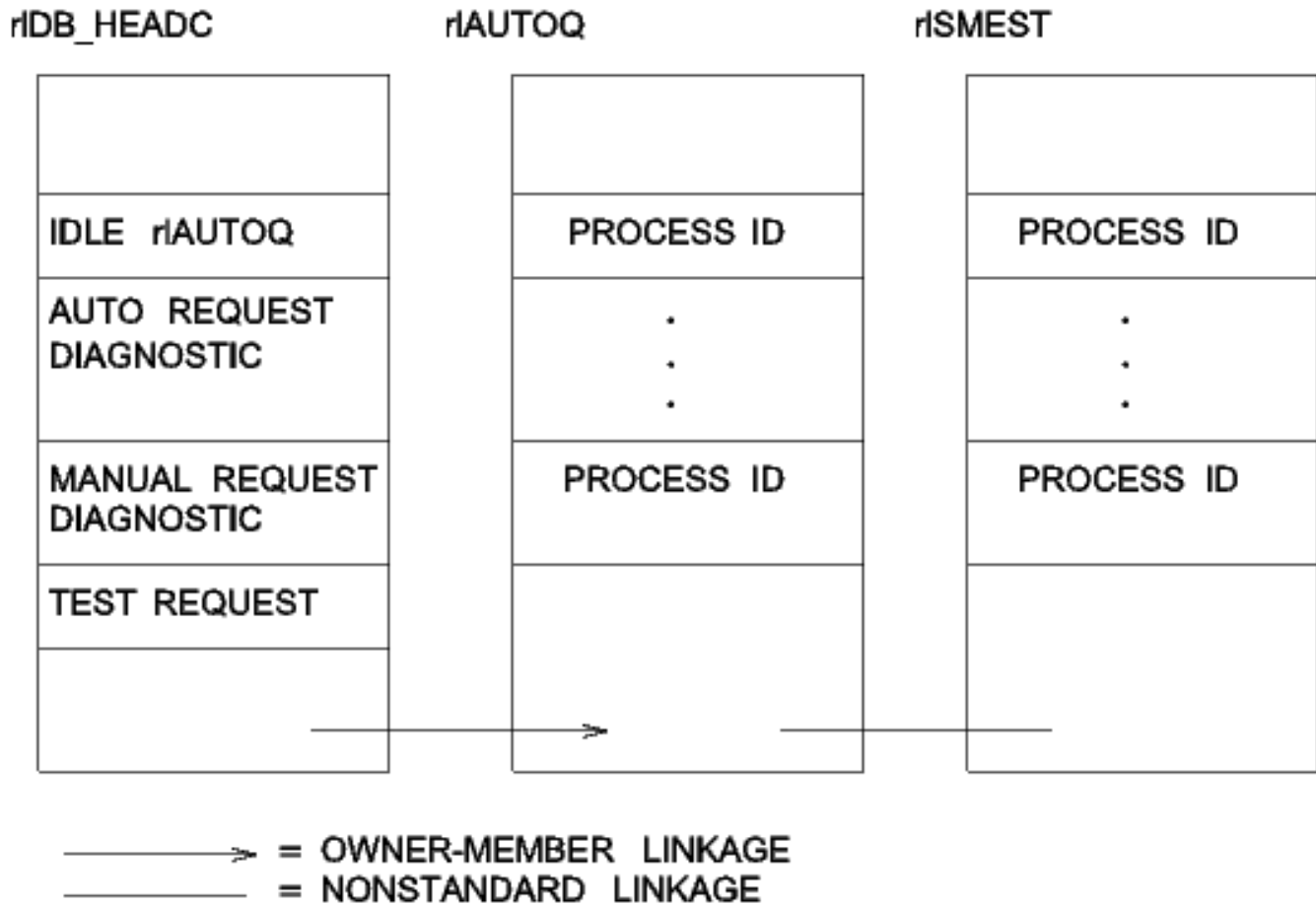


Figure 1 (AUTOQ-1) Data Structures of the AUTOQ Audit

3. AUDIT CHECKS

The audit invokes the linkage driver, DRLINK, to verify the linkages between the tuples of rIAUTOQ and rIDB_HEADC.

If the linkage check passes, DRLINK invokes the semantic check function to determine whether the rIAUTOQ tuple is on the idle list. If it is on the idle list, the audit proceeds to the next tuple. If it is on the busy list, the audit makes the following check:

- Verifies that the terminal process that owns the rIAUTOQ tuple is alive.
- Obtains the corresponding rISMEST tuple and verifies that the terminal process ID in the rISMEST tuple matches the terminal process ID in the rIAUTOQ tuple.

If both checks pass, the semantic check function returns to DRLINK to fetch the next tuple. If a check fails, recovery is done on the rIAUTOQ tuple.

4. ERROR HANDLING

The audit handles errors by zeroing the appropriate tuples in RLAUTOQ and linking them to the idle list.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error

code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	The pid fields in the rSMEST and rIAUTOQ tuples do not match.
Possible Error Effect:	MRA may not process maintenance requests for the circuit correctly.
Corrective Action Taken:	The rIAUTOQ tuple in error is recovered and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIAUTOQ tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIAUTOQ tuple.
<i>Dump 1 =</i>	rIAUTOQ tuple.
<i>Dump 2 =</i>	rSMEST tuple.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	An rIAUTOQ tuple with a dead process ID (PID) but is on a list of active rIAUTOQ tuples.
Possible Error Effect:	MRA may not process maintenance requests for the circuit correctly.
Corrective Action Taken:	The rIAUTOQ tuple in error is recovered and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIAUTOQ tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIAUTOQ tuple.
<i>Dump =</i>	rIAUTOQ tuple.

BASE

Software Release: 5E10 and later

1. *SLC*[®] 96 BASE (BASE) AUDIT

The *SLC*[®] 96 digital carrier line unit (DCLU) connects to the module controller/time-slot interchanger (MCTSI) of the switching modules by means of peripheral interface control buses (PICBs). Each PICB maintains control information for a group of *SLC*[®] 96 digital facilities interfaces (SDFIs). Control information includes the detection of customer originations, hardware faults, and service interrupts. There are two types of PICBs, DCLU common control PICBs and DCLU DFI control PICBs. The DCLU DFI control PICBs communicate between the control interface (CI) in the MCTSI and the SDFIs; they are not audited. The common control PICBs communicate between the CI and the registers in the common control area for the DCLU service group. See Figure 1 .

The determination of which PICB controls each group of SDFIs is stored in the *SLC*[®] 96 base relation (RLS96BASE). The BASE audit verifies the accuracy of the information in this relation and ensures that at least one service group of SDFIs is in service at all times.

The BASE audit is part of the switch maintenance audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented, nonsegmented, and initialization modes.

2. DATA STRUCTURES

The BASE audit checks the dynamic RLS96BASE relation against the static peripheral interface control bus relation (RLPICB). The audit also acquires information from the switch maintenance equipment status table relation (RLSMEST). Figure 2 shows the fields in these relations checked by the audit. The RLPICB relation contains tuples for every peripheral interface control bus in the system. If the system has *SLC*[®] 96, RLPICB will also contain, among other control bus information, the digital carrier line unit (DCLU) number, service group number, and common control PICB to which the SDFI connects.

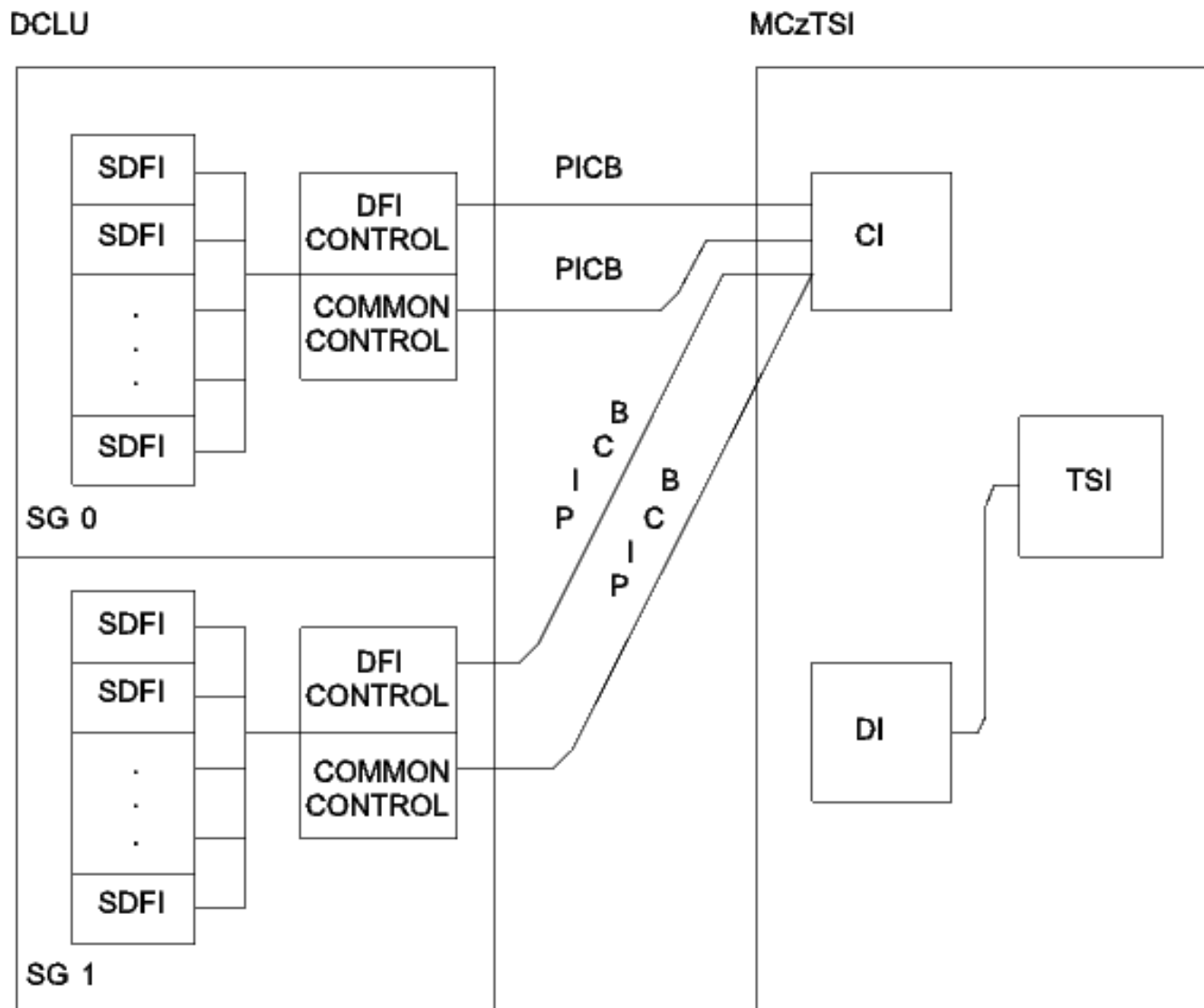


Figure 1 (BASE-1) BASE Audit Hardware Configuration

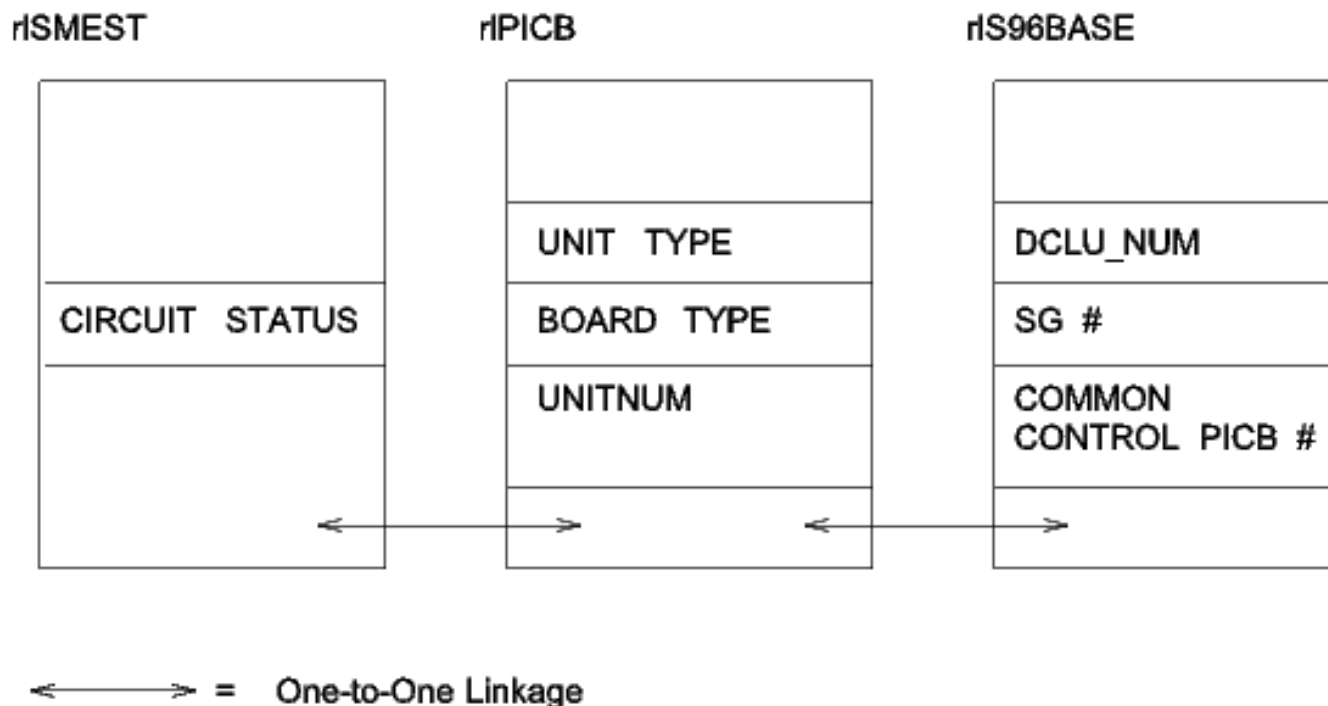


Figure 2 (BASE-2) Data Structures of the BASE Audit

3. AUDIT CHECKS

The BASE audit calls the static driver (DRSTATIC) to read rLPICB and find common control PICB tuples. If the unit type of the tuple in RLPICB is DCLU and the board type is SLC[®] 96 common control PICB, the audit reads RLS96BASE using the unitnum field from RLPICB and the service group as the key. The audit assumes that PICBs with odd numbers belong to service group 1 and PICBs with even numbers belong to service group 0. The audit ensures that all the necessary tuples in RLPICB have entries in RLS96BASE. If a tuple does not exist in RLS96BASE, there is an error. For tuples that exist in RLS96BASE, the audit ensures that the dclu_sgroup numbers are consistent. The audit then checks RLSMEST for the PICB number. If the PICB number is in-service, the DCLU should be using it as a normal PICB and RLS96BASE should contain its number. If the PICB is out of service, RLS96BASE should contain the PICB number from the mate service group. The SDFI control bits in the DCLU are checked to ensure that they are consistent with the in-service/out-of-service status of the DCLU service group.

4. ERROR HANDLING

The BASE audit recovers from errors by inserting new tuples in RLS96BASE if they exist in rLPICB but are not found in RLS96BASE. The key of the new tuple is obtained from the dclu_sgroup number in rLPICB and the appropriate common control PICB is obtained from the RLSMEST relation tuple for that DCLU service group. If the PICB exists in RLS96BASE in the out-of-service state and RLSMEST shows the PICB to be in-service, the audit overwrites the tuple in RLS96BASE to make it consistent with the RLSMEST information. The corrective action for wrong SDFI control bits in the DCLU is to overwrite them according to the status of the DCLU service groups in the RLSMEST relation.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The DCLU mux select control bit for service group 0 is not 0 when DCLU service group 0 is ACTIVE and service group 1 is out-of-service.
Possible Error Effect:	The switch maintenance module processor will not be able to communicate with the SDFIs belonging to the service group in error. The consequence of this is that no calls can originate from nor terminate at the ports associated with these SDFIs. Also, it is not possible to perform maintenance (such as, run diagnostics) on the SDFI belonging to this service group.
Corrective Action Taken:	The SDFI control bits and the mate SDFI control bits are corrected according to the statuses of both service group 0 and 1 in SMEST (the dynamic relation for the switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SDFI control bits for service group 0.
<i>Bad Data =</i>	Value of service group 0 control bit in the DCLU control mux select register.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The key should be interpreted as: bit 0 - the service group number bits 1-3 - the digital concentration line unit number.
<i>Dump 1 =</i>	rSMEST (equipment status table) tuple of service group 0.
<i>Dump 2 =</i>	rSMEST (equipment status table) tuple of service group 1.

Table 2 Error Code: CFACT_F

Error Code:	CFACT_F
Error Description:	According to SMEST, both DCLU service groups 0 & 1 are ACTIVE. The exclusive OR of the SDFI control bits and mate SDFI control bits is not equal to zero for DCLU service group 0.
Possible Error Effect:	The switch maintenance module processor will not be able to communicate with the SDFIs belonging to the service group in error. The consequence of this is that no calls can originate from nor terminate at the ports associated with these SDFIs. Also, it is not possible to perform maintenance (such as, run diagnostics) on the SDFI belonging to this service group.
Corrective Action Taken:	The SDFI control bits and the mate SDFI control bits are corrected according to the statuses of both service group 0 and 1 in SMEST (the dynamic relation for the switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SDFI control bits for service group 0.
<i>Bad Data =</i>	Value of service group 0 control bit in the DCLU control mux select register.
<i>Good Data =</i>	Service group 0 control bit in the mate DCLU control mux select register. The problem here is both bad and good data are not 0. The good data field here should not be interpreted as the correct data.
<i>Logical Index =</i>	The key should be interpreted as: bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump 1 =</i>	rSMEST (equipment status table) tuple of service group 0.
<i>Dump 2 =</i>	rSMEST (equipment status table) tuple of service group 1.

Table 3 Error Code: CFACT_T

Error Code:	CFACT_T
Error Description:	According to SMEST, both DCLU service groups 0 & 1 are ACTIVE. The exclusive OR of the SDFI control bits and mate SDFI control bits is not equal to zero for DCLU service group 1.
Possible Error Effect:	The switch maintenance module processor will not be able to communicate with the SDFIs belonging to the service group in error. The consequence of this is that no calls can originate from nor terminate at the ports associated with these SDFIs. Also, it is not possible to perform maintenance (such as, run diagnostics) on the SDFI belonging to this service group.
Corrective Action Taken:	The SDFI control bits and the mate SDFI control bits are corrected according to the statuses of both service group 0 and 1 in SMEST (the dynamic relation for the

	switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the SDFI control bits for service group 1.
<i>Bad Data</i> =	Value of service group 1 control bit in the DCLU control mux select register.
<i>Good Data</i> =	Service group 1 control bit in the mate DCLU control mux select register. The problem here is both bad and good data are not 0. The good data field here should not be interpreted as the correct data.
<i>Logical Index</i> =	The key should be interpreted as: bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump 1</i> =	rSMEST (equipment status table) tuple of service group 1.
<i>Dump 2</i> =	rSMEST (equipment status table) tuple of service group 0.

Table 4 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	Own PICB service group is in service, but its primary PICB is not being used.
Possible Error Effect:	May not be able to set up a call through the erroneous PICB control path.
Corrective Action Taken:	Insert the correct PICB information from RLSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the common control PICB field of a copy of the tuple in RLS96BASE being audited.
<i>Bad Data</i> =	Value of the PICB (peripheral interface control bus) being audited.
<i>Good Data</i> =	The expected value of the PICB.
<i>Logical Index</i> =	The key of the tuple in the dynamic relation RLS96BASE being audited; the key should be interpreted as: bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump</i> =	rSMEST (equipment status table) tuple of the corresponding unit.

Table 5 Error Code: KEY

Error Code:	KEY
Error Description:	The service group number part of the access key was garbaged. This was provided as a check for wild writes.
Possible Error Effect:	There should be no effect on the system.
Corrective Action Taken:	The correct key is written.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of a copy of the tuple in RLS96BASE being audited.
<i>Bad Data</i> =	Value of the service group number that is found to be in error.
<i>Good Data</i> =	Static copy of the service group number.
<i>Logical Index</i> =	The key of the tuple in the dynamic relation RLS96BASE being audited; the key should be interpreted as: bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump</i> =	NA.

Table 6 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	The DCLU (digital concentration line unit) part of the access key for the rLS96BASE tuple was garbaged.
Possible Error Effect:	There should be no effect on the system.
Corrective Action Taken:	The correct key is written.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of a copy of the tuple in RLS96BASE to be audited.
<i>Bad Data</i> =	Value of DCLU number that is found to be in error.
<i>Good Data</i> =	Static copy of the DCLU number.
<i>Logical Index</i> =	The key of the tuple in the dynamic relation RLS96BASE being audited; the key should be interpreted as: bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump</i> =	NA.

Table 7 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Could not find the tuple we were looking for in RLS96BASE.
Possible Error Effect:	Cannot set up a call through the PICB control path that is missing.
Corrective Action Taken:	The correct rLS96BASE tuple is inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of a copy of the rLS96BASE tuple where the information would be placed by DBfrdtup.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Relation identity of rLS96BASE, this field should not be interpreted in this case as good data.
<i>Logical Index =</i>	The key of the tuple in the dynamic relation RLS96BASE being audited; the key should be interpreted as bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump =</i>	NA.

Table 8 Error Code: MC_ALWOOS

Error Code:	MC_ALWOOS
Error Description:	The mate DCLU mux select control bit for service group 0 is not 1 when DCLU service group 1 is ACTIVE and service group 0 is out-of-service.
Possible Error Effect:	The switch maintenance module processor will not be able to communicate with the SDFIs belonging to the service group in error. The consequence of this is that no calls can originate from nor terminate at the ports associated with these SDFIs. Also, it is not possible to perform maintenance (such as, run diagnostics) on the SDFI belonging to this service group.
Corrective Action Taken:	The SDFI control bits and the mate SDFI control bits are corrected according to the statues of both service group 0 and 1 in SMEST (the dynamic relation for the switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SDFI control bits for service group 0.
<i>Bad Data =</i>	Value of service group 0 control bit in the DCLU control mux select register.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	The key should be interpreted as: bit 0 - the service group number bits 1-3 - the digital concentration line unit number.
<i>Dump 1 =</i>	rSMEST (equipment status table) tuple of service group 0.
<i>Dump 2 =</i>	rSMEST (equipment status table) tuple of service group 1.

Table 9 Error Code: MM_ALWOOS

Error Code:	MM_ALWOOS
Error Description:	The mate DCLU mux select control bit for service group 1 is not 1 when DCLU service group 0 is ACTIVE and service group 1 is out-of-service.
Possible Error Effect:	The switch maintenance module processor will not be able to communicate with the SDFIs belonging to the service group in error. The consequence of this is that no calls can originate from nor terminate at the ports associated with these SDFIs. Also, it is not possible to perform maintenance (such as, run diagnostics) on the SDFI belonging to this service group.
Corrective Action Taken:	The SDFI control bits and the mate SDFI control bits are corrected according to the statues of both service group 0 and 1 in SMEST (the dynamic relation for the switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SDFI control bits for service group 0.
<i>Bad Data =</i>	Value of service group 1 control bit in the mate DCLU control mux select register.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	The key should be interpreted as: bit 0 - the service group number bits 1-3 - the digital concentration line unit number.

<i>Dump 1 =</i>	rlSMEST (equipment status table) tuple of service group 1.
<i>Dump 2 =</i>	rlSMEST (equipment status table) tuple of service group 0.

Table 10 Error Code: OOS_MATE

Error Code:	OOS_MATE
Error Description:	The dynamic relation RLS96BASE does not contain its mate PICB when its primary service group is out-of-service.
Possible Error Effect:	Cannot set up a call through the erroneous PICB control path.
Corrective Action Taken:	Put the correct PICB information in RLS96BASE from RLSMEST (the dynamic relation for the switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the control picb field of a copy of the tuple in RLS96BASE being audited.
<i>Bad Data =</i>	Value of the control PICB being audited.
<i>Good Data =</i>	The correct PICB value.
<i>Logical Index =</i>	The key of the tuple in the dynamic relation RLS96BASE being audited; the key should be interpreted as: bit 0 - the service group number bits 1-3 - the digit concentration line unit number.
<i>Dump =</i>	rlSMEST (equipment status table) tuple of the corresponding unit.

Table 11 Error Code: P_ALWOOS

Error Code:	P_ALWOOS
Error Description:	The DCLU mux select control bit for service group 1 is not 0 when DCLU service group 1 is ACTIVE and service group 0 is out-of-service.
Possible Error Effect:	The switch maintenance module processor will not be able to communicate with the SDFIs belonging to the service group in error. The consequence of this is that no calls can originate from nor terminate at the ports associated with these SDFIs. Also, it is not possible to perform maintenance (such as, run diagnostics) on the SDFI belonging to this service group.
Corrective Action Taken:	The SDFI control bits and the mate SDFI control bits are corrected according to the statues of both service group 0 and 1 in SMEST (the dynamic relation for the switch maintenance equipment status table).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SDFI control bits for service group 0.
<i>Bad Data =</i>	Value of service group 1 control bit in the DCLU control mux select register.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The key should be interpreted as: bit 0 - the service group number bits 1-3 - the digital concentration line unit number.
<i>Dump 1 =</i>	rlSMEST (equipment status table) tuple of service group 1.
<i>Dump 2 =</i>	rlSMEST (equipment status table) tuple of service group 0.

Table 12 Error Code: PICB

Error Code:	PICB
Error Description:	Common board circuit name of GLNULCKT was returned which indicates a problem with CKTDATA read. We are unable to continue the checks.
Possible Error Effect:	Invalid circuit name could be used in checks.
Corrective Action Taken:	None.
Manual Action Required:	ODD needs to be investigated to determine where the inconsistency is in CKTDATA.
Dump Description:	
<i>Error Address =</i>	1 if we are reading SMEST for the mate service group. Otherwise, this is 0.
<i>Bad Data =</i>	Mate service group PICB if error address is 1. Otherwise, this PICB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Common board circuit name.
<i>Dump =</i>	NA.

Table 13 Error Code: SMEST

Error Code:	SMEST
Error Description:	Could not read rLSMEST (switch maintenance equipment status table).
Possible Error Effect:	The system would not be able to maintain the PICBs properly.
Corrective Action Taken:	The SMEST audit is requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of a copy of the tuple where the information is placed by DBfrdtup().
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Relation identity of rLSMEST, this field should not be interpreted as good data in this case.
<i>Logical Index =</i>	The key (in this case the unit identity) used to read the RLSMEST relation.
<i>Dump =</i>	NA.

BCLGCNT

Software Release: 5E10 and later

1. BCLID MLHG/TRUNK GROUP DYNAMIC COUNTS (BCLGCNT) AUDIT

The RLBCLCNT relation is used by the local area signaling services (LASS) feature bulk calling line identification (BCLID) to store the number of BCLID messages to a multi-line hunt group (MLHG) or trunk group.

2. DATA STRUCTURES

The RLBCLCNT relation is a dynamic relation used by the BCLGCNT audit to perform dynamic to static checks against the static RLBCL_MHTG data structure. The BCLGCNT audit references the **grpidx** attribute of tuples of the RLBCLCNT relation. This attribute is the trunk group or the multi-line hunt group Identifier.

3. AUDIT CHECKS

The BCLGCNT audit uses the DRINDEX audit driver to step through the RLBCLCNT relation.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

BCLLCNT

Software Release: 5E10 and later

1. BCLID LOGICAL TERMINATION PORTS DYNAMIC COUNTS (BCLLCNT) AUDIT

The RLBCLCNT relation is used by the local area signaling services (LASS) feature bulk calling line identification (BCLID) to store the number of BCLID messages to a multi-line hunt group (MLHG) or trunk group (TG).

2. DATA STRUCTURES

The RLBCLCNT relation is a dynamic relation used by the BCLLCNT audit to perform dynamic to static checks against the static RLBC_LTM data structure. The BCLLCNT audit references the following attributes of tuples of the relation RLBCLCNT:

lead_module = Lead module.

ltm_idx = The logical termination index.

ltm_type = The type of ltm_idx.

3. AUDIT CHECKS

The BCLLCNT audit uses the DRINDEX audit driver to step through the RLBCLCNT relation.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

BD-A**Software Release:** 5E10 - 5E11**1. BUFFER DESCRIPTOR (BD) AUDIT**

Buffers are used in providing packet services. Each buffer has a buffer descriptor (BD) associated with it. The BD holds information about the ownership and state of its associated buffer. Buffers are passed between data structures by changing the owner of the associated BD. There are two types of BDs:

- General purpose buffer descriptors (GPBDs) GPBDs are found in all protocol handlers (PHs) and all packet interface (PIs).
- Synchronous protocol data emulator buffer descriptors (SPBDs). These are also known as SPYDER BDs. SPYDER BDs are found only in PH2s and the PI1.

The BD audit is provided in the PH2 and PI's port processor and in the PH3's associated processor (AP) side.

2. DATA STRUCTURES

The BD audit verifies the following data:

OCgpcnt[] = Array of overload control counts of GPBD usage (PHs only).

OCspcnt[] = Array of overload control counts of SPBD usage (PH2 only).

PS_iscb.fbdb_ptr = First and last free list BD indicators and restock counter (PH3 and PI2).

PSbapfront, PS_iscb.ipszcnt and PS_iscb.ipretcnt = Overload control counts used to track the number of queued GPBDs (PH3 only).

PSbdgp[] = Array of GPBDs.

PSbdgphd = Headcell of GPBD free list.

PSbdsp[] = Array of SPBDs (PH2 only).

PSbdsplst = Tail of the subscriber BD pool (PH2 only).

PSbufgp[] = Array of GP buffers.

PSbufsp[] = Array of SP buffers (PH2 only).

PSlscb[] = Array of LSCBs (Local area network System Control Blocks) (PH2 and PI1 only).

PSrbdend[] = Array of offsets that mark the end of each LAN's RBD (Local Area Network's Restart Buffer Descriptor) pool (PH2 and PI1 only).

PSrbdfrz[] = Array of offsets containing the head of each LAN's list of BDs to be used when restarting the LAN from an exhaustion condition (PH2 and PI1 only).

PSs_fbdb = Head of the subscriber BD pool (PH2 only).

Next is a list of fields of data structures which own lists of BDs. It is organized by structure and field name. The order these lists are checked varies somewhat among the processors.

PSlccb[] = Array of logical channel control blocks (LCCBs).

- hold_buf[] (PHs only).
- retran_buf[] (PHs only).
- qdsetup_index (PHs only).
- rctl_bfr (PHs only).
- ictl_bfr (PHs only).
- intrpt_bfr (PHs only).
- mptmbd (PHs only).

PSalcb[] = Array of access line control blocks (ALCBs).

- al_pvcmap (PHs only).

PSifd[] = Array of internal protocol blocks (IFDs).

- bd.

PSs_tfd[] = Array of SPYDER Transmit Frame Descriptors (STFDs).

- tbd (PHs only).

PSs_rfd[] = Array of SPYDER receive frame descriptors (SRFDs).

- rbd (PHs only).

PSI0tfd[] and PSI1tfd[] = Arrays of LAN transmit frame descriptors (LTFDs).

- tbdoff (PH2 only).
- tbdptr (PH3 and PI2).

PSI0rfd[] and PSI1rfd[] = Arrays of LAN receive frame descriptors (LRFDs).

- rbdoff (PH2 only).
- rbdptr (PH3 and PI2).

PSI0txblk[], PSI1txblk[] = Arrays of LAN transmit command blocks (LTCBs).

- tbdoff (PH2 and PI1 only).
- GPBD free pool.
- SPBD free pool (PH2 only).

- LAN RBD pools (PH2 and PI1 only).

All of the data structure mentioned in this section are global and dynamic. Those not verified by the BD audit have audits of their own or are maintained by operational code as they are used.

3. AUDIT CHECKS

All error codes used by the BD audit are listed. Not all of these error codes are used in every processor. Detailed information is given in the comment blocks that follow. This list gives a (very) high level view of the order in which the checks are made. If the audit has just been entered:

ZAP = System interrupt handling.

The remainder of the checks are done on a per list or per BD basis:

ACBIT = Last BD in LAN RBD pool.
 AVAIL_CNT = Restock count.
 BDCOUNT = Number of queued BDs.
 DEAD_LNKD = Linkage to LAN RBD pools.
 EVEN_COUNT = Length of LAN RBD pool.
 INV = Last BD in a list.
 LGLOSTTUP = "Lost" GPBDs.
 LIST = Ability to access free list.
 LOSTTUP = "Lost" SPBDs.
 OUTOFRANGE = Buffer addresses and next BD offsets.
 R_COUNTS = OC GPBD usage counts.
 RMCOUNTS = OC SPBD usage counts.
 TRAVERSE = Integrity of a BD list.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	A LAN's RBD pool was found whose last BD does not have its end of list (el) bit set.
Possible Error Effect:	May cause the LAN processor to behave unpredictably if it gets to the last BD on

	its RBD pool.
Corrective Action Taken:	The Audit sets the el bit on the last BD in the LAN's RBD pool if at least one BD is in the pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index</i>	The LAN side with the RBD pool whose last BD in the pool does not have its el bit set.
<i>Dump =</i>	The last BD in the RBD pool whose el bit should have been set.

Table 2 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT
Error Description:	The BD restock counter (PS_iscb.fdbd_ptr→restock_cnt) in a PH3 is normally set to a default value of PSBMINFREEBDS. If the number of available BDs goes to zero, it is also set to zero. As BDs are returned (and seized) from the free list, this counter is updated until it reaches or exceeds the default value. While it is less than the default value, the PH is in restock mode. Allocation of BDs in the IOP will be restricted. This error indicates that the restock counter was less than the default value but there were at least twice PSBMINFREEBDS actually available on the free list.
Possible Error Effect:	The PH is erroneously in restock mode. BD allocation in the IOP will be restricted.
Corrective Action Taken:	Reset the threshold (PS_iscb.fdbd_ptr→restock_cnt) to the default value PSBMINFREEBDS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the first free BD.
<i>Bad Data =</i>	The restock threshold (PS_iscb.fdbd_ptr→restock_cnt).
<i>Good Data =</i>	The actual number of BDs on the free list.
<i>Logical Index =</i>	The default restock count value (PSBMINFREEBDS).
<i>Dump =</i>	NA.

Table 3 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	In a PH3 both the AP and IOP processors use the same free list (GPBDs). Furthermore, GPBDs which have been removed from this list but are still queued up waiting to be processed are marked as being "owned" by the free list (BD.bd_owner = OCCH_FREE). Thus, overload control's counts of the number of free BDs may be inflated. To counteract this three counters are kept (PSbapfrcnt, PS_iscb.ipszcnt and PS_iscb.ipretcnt) whose sum equals the number of BDs currently queued. This error results when the actual number of BDs on the free list (as calculated by the audit) + the number of BDs queued differs from OCgpcnt[OCCH_FREE] (OC's count of GPBDs marked as being owned by the free list) by more than PSBDCHMX. The tolerance is there to allow for the fact that the IOP may be manipulating BDs concurrently with the audit.
Possible Error Effect:	May result in erroneous overload control states.
Corrective Action Taken:	The audit will add or subtract the appropriate amount from PSbapfrcnt to make all the values consistent.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	OC's count of free GPBDs (OCgpcnt[OCCH_FREE]).
<i>Bad Data =</i>	The number of queued BDs.

<i>Good Data</i> =	The actual number of BDs on the free list.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The audit encountered one of two conditions of a LAN that prevents the BD Audit from collecting lost BDs and verifying the overload control counts. The first condition is that a LAN was in the suspended state, and the head of the LAN's RBD pool could not be located. This situation can occur if all of a LAN's RFDs are in use and the LAN had an incoming frame for which it needed an RFD. In this case, only the LAN has access to the head of its RBD pool. This is a self correcting situation in that the operational code will resume the LAN as it frees up RFDs by unloading frames. This condition implies that the BD Audit can only traverse the other BD linked lists, but can not reclaim lost BDs nor verify the overload control buffer usage counts. The other condition is that while traversing a LAN's RFD (circular) list, the BD Audit encountered an RFD with an out of range or misaligned next RFD offset. It is unlikely that the BD Audit will encounter this situation before the operational code does. NOTE: If the operational code encounters this situation, an assert is called that will result in an initialization of packet switching data structures that will clear up this problem.
Possible Error Effect:	This error report is only meant to document self-correcting situations.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Has a value of 1 if the condition found is a bad RFD linkage. Has a value of 2 if the condition found is that of a LAN in the suspended state.
<i>Bad Data</i> =	If the condition found was a bad RFD linkage, this is the value found as the next RFD offset, in the RFD. Otherwise, this is significant.
<i>Good Data</i> =	If the condition found was a bad RFD linkage, this is the number of RFDs that had been traversed, excluding the one with the bad linkage.
<i>Logical Index</i> =	The LAN side that is in the suspended state.
<i>Dump</i> =	NA.

Table 5 Error Code: EVEN_COUNT

Error Code:	EVEN_COUNT
Error Description:	A LAN was found whose RBD pool does not have the expected number of BDs.
Possible Error Effect:	If the RBD pool was in fact out of BDs, then the LAN could go into the "no resource" state. Otherwise, this error condition is a warning that the designed allocation of BDs has been disturbed, and that subsequent GP BD shortages may occur.
Corrective Action Taken:	If the RBD pool had less than the designed allocation of BDs and there are enough BDs in the GP BD pool, the audit will attempt to replenish RBD pool. If the RBD pool had more than the designed allocation, the audit will update the value of [IMlbdelta] to indicate the number of surplus BDs in the RBD pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The length of the RBD pool as was computed by the audit.
<i>Good Data</i> =	The number of BDs by which the pool was over (positive value) or under (negative value).
<i>Logical Index</i> =	The LAN side whose RBD pool is not of the right length.

<i>Dump</i> =	NA.

Table 6 Error Code: INV

Error Code:	INV
Error Description:	The last BD offset of a LAN's RBD pool, or the GP BD pool, of the SP BD pool that did not match the BD, that was found by the audit at the end of the pool.
Possible Error Effect:	This error can cause BDs to be lost, since the LAN RBD pool replenishment function or the BD returning function most likely had been using an erroneous "last BD" offset variable.
Corrective Action Taken:	The "last BD" offset variable is set to contain the offset of the BD found by the audit at the end of the pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value that identifies which pool had an inconsistent last BD offset. 1 = A LAN RBD pool. 2 = The GP BD pool. 6 = The SP BD pool.
<i>Bad Data</i> =	The erroneous "last BD" offset.
<i>Good Data</i> =	The offset of the BD found by the audit at the end of the LAN RBD pool.
<i>Logical Index</i> =	If the error address is 1, this is the LAN side whose "last RBD" offset is in error. Otherwise, the logical address is not meaningful.
<i>Dump</i> =	The number of BDs in the RBD pool as counted by the audit. Read only the high-order (16-bit) halfword.

Table 7 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	The audit found a general purpose (GP) BD that was untouched, and therefore lost.
Possible Error Effect:	If this error condition occurs often enough, it may ultimately result in BD shortage or even exhaustion.
Corrective Action Taken:	The lost BD is put back into the GP BD pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index into the array of GP BDs that identifies the lost BD found by the audit.
<i>Dump</i> =	The contents of the lost GP BD.

Table 8 Error Code: LIST

Error Code:	LIST
Error Description:	The BD audit was unable to audit the free list of GPBDs in a PH3 due to contention with the IOP while trying to access the head of the list.
Possible Error Effect:	The audit will be unable to complete its checks. These include audits of overload control levels and checks for lost BDs.
Corrective Action Taken:	The audit will complete without taking any further action.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the first BD on the free list (PS_iscb.fbdb_ptr→bd_head).
<i>Bad Data</i> =	Offset of the first BD on the free list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 9 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a SPYDER BD that was untouched, and therefore lost.
Possible Error Effect:	If this error condition occurs often enough, it may ultimately result in BD shortage or even exhaustion.
Corrective Action Taken:	The lost BD is put back into the SP BD pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index into the array of SP BDs that identifies the lost BD found by the audit.
<i>Dump</i> =	The contents of the lost SP BD.

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A BD was found whose buffer address is out of range, or whose next BD offset is out of range or alignment.
Possible Error Effect:	This error can cause illegal memory access (read or write) by the code that is resident in the port processor, the LAN, or the SPYDER/SPORT processors.
Corrective Action Taken:	If the BD's buffer address is out of range, it will be changed to contain the address of the beginning of its assigned buffer. Note, that the user bits are preserved. If the BD's next BD offset is out of range or alignment, it will be corrected to the null offset value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value that identifies the type of BD. 13 = A GP BD. 14 = A SP BD.
<i>Bad Data</i> =	The bad buffer address or next BD offset as indicated by "GOOD DATA."
<i>Good Data</i> =	An indication of the type of error found (that is, 4 if the next BD offset was anomalous or 5 if the buffer address was out of range).
<i>Logical Index</i> =	The index into the appropriate BD array that identifies the BD.
<i>Dump</i> =	The contents of the BD and its buffer.

Table 11 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The audit found a physical PH channel whose overload control usage count of general purpose (GP) buffers was in error.
Possible Error Effect:	May result in erroneous overload control states.
Corrective Action Taken:	The audit corrects the count found to be in error with the difference between the bad value and the value as recomputed by the BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of the overload control (GP) buffer usage count that was found to be in error.
<i>Good Data</i> =	The value of the overload control buffer usage count as recomputed by the audit.
<i>Logical Index</i> =	The ID of the physical PH channel whose count was found to be in error.
<i>Dump</i> =	NA.

Table 12 Error Code: RMCOUNTS

Error Code:	RMCOUNTS
Error Description:	The audit found a physical PH channel whose overload control usage count of SPYDER (SP) buffers was in error.
Possible Error Effect:	May result in erroneous overload control states.
Corrective Action Taken:	The audit corrects the count found to be in error with the difference between the

	bad value and the value as recomputed by the BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value of the overload control (SP) buffer usage count that was found to be in error.
<i>Good Data =</i>	The value of the overload control buffer usage count as recomputed by the audit.
<i>Logical Index =</i>	The ID of the physical PH channel whose count was found to be in error.
<i>Dump =</i>	NA.

Table 13 Error Code: TRAVERSE

Error Code:	TRAVERSE
Error Description:	The audit found an anomaly in a linked list of BDs that it was traversing. The list and error type can be determined using the data dumped by the audit.
Possible Error Effect:	Varies according to conditions and the type of error found. However, some possibilities are: BD shortages, resource exhaustion or overload control states.
Corrective Action Taken:	The audit will truncate the list of BDs in which the error was found. If the BD list is owned by an ALCB, LCCB or IFD, the owning structure will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	A value indicating what type of data structure the list was owned by. 1 = Verification of the LAN RBD Pool. The LOGICAL INDEX represents the LAN side. 2 = Verification of the GP BD Pool. The LOGICAL INDEX is not relevant. 3 = PH2 or PI: Verification of BDs attached to LAN TCBs (PS10txblk[] or PS11txblk[]). PH3: Verification of BDs attached to LAN TFDs (PSI0tfd[] or PSI1tfd[]). 4 = Verification of BDs attached to LAN RFDs (PSI0rfd[] or PSI1rfd[]). 5 = Verification of BDs attached to IFDs (PSifd[]). 6 = Verification of BDs attached to SPYDER BD Pool. The LOGICAL INDEX is not relevant. 7 = Verification of BDs attached to SPYDER RFDs (PSs_rfd[]). 8 = Verification of BDs attached to SPYDER TFDs (PSs_tfd). 9 = Verification of BDs attached to ALCBs (PSalcb[]). 10 = Verification of BDs attached to LCCBs (PSlccb[]).
<i>Bad Data =</i>	The length of the free list if that was the list being traversed. Otherwise, none.
<i>Good Data =</i>	A value that indicates the type of error found by the audit. 4 = A BD in the BD list being traversed was found whose next BD offset was out of range. 7 = A BD in the BD list being traversed was found to have an owner that was inappropriate for the BD list, (such as, a BD whose owner is "free" but is

	attached to an LCCB).
	8 = A BD list (other than any of the BD pools) whose length exceeded the maximum allowed for a valid packet/frame.
<i>Logical Index</i> =	The index of the owning data structure unless otherwise specified in "Error Address."
<i>Dump 1</i> =	The contents of the last error-free BD in the list. This dump can be a dummy BD that is used by the audit to facilitate its traversal of the list.
<i>Dump 2</i> =	The contents of the BD in error.
<i>Dump 3</i> =	The contents of the buffer associated with the BD in error.

Table 14 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Resets and/or ASSERTS.
Corrective Action Taken:	If the audit was traversing lists belonging to LCCBs, ALCBs or IFDs at the time of the error, the appropriate structure will be idled. In any case, the audit will not do any more checks during this invocation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value indicating the type of work the BD audit was doing at the time the error occurred. Note that the audit may have been finished with the work item indicated and is in the process of moving on to the next task. If the operational code encounters this situation, an assert is called that will result in an initialization of packet switching data structures that will clear up this problem. If the operational code encounters this situation, an assert is called that will result in an initialization of packet switching data structures that will clear up this problem.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	If the audit was working on an array of data structures, this is the index of the element in the array which was being accessed at the time the error occurred. Exceptions are noted in "Error Address."
<i>Dump</i> =	NA.

BD-B

Software Release: 5E12 and later

1. BUFFER DESCRIPTOR (BD) AUDIT

Buffers are used in providing packet services. Each buffer has a buffer descriptor (BD) associated with it. The BD holds information about the ownership and state of its associated buffer. Buffers are passed between data structures by changing the owner of the associated BD.

There are two types of BDs:

- General purpose buffer descriptors (GPBDs) GPBDs are found in all protocol handlers (PHs) and all packet interface (PIs).
- Synchronous protocol data emulator buffer descriptors (SPBDs). These are also known as SPYDER BDs. SPYDER BDs are found only in PH2s and the PI1.

The BD audit is provided in the PH2 and PI's port processor and in the PH3's associated processor (AP) side.

2. DATA STRUCTURES

For details on the fields contained in the PSGPBD and PSSPBD structures, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	A LAN's RBD pool was found whose last BD does not have its end of list (el) bit set.
Possible Error Effect:	May cause the LAN processor to behave unpredictably if it gets to the last BD on its RBD pool.
Corrective Action Taken:	The Audit sets the el bit on the last BD in the LAN's RBD pool if at least one BD is in the pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index</i>	The LAN side with the RBD pool whose last BD in the pool does not have its el bit set.
<i>Dump =</i>	The last BD in the RBD pool whose el bit should have been set.

Table 2 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT

Error Description:	The BD restock counter (PS_iscb.fdbd_ptr->restock_cnt) in a PH3 is normally set to a default value of PSBMINFREEBDS. If the number of available BDs goes to zero, it is also set to zero. As BDs are returned (and seized) from the free list, this counter is updated until it reaches or exceeds the default value. While it is less than the default value, the PH is in restock mode. Allocation of BDs in the IOP will be restricted. This error indicates that the restock counter was less than the default value but there were at least twice PSBMINFREEBDS actually available on the free list.
Possible Error Effect:	The PH is erroneously in restock mode. BD allocation in the IOP will be restricted.
Corrective Action Taken:	Reset the threshold (PS_iscb.fdbd_ptr->restock_cnt) to the default value PSBMINFREEBDS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the first free BD.
<i>Bad Data =</i>	The restock threshold (PS_iscb.fdbd_ptr->restock_cnt).
<i>Good Data =</i>	The actual number of BDs on the free list.
<i>Logical Index =</i>	The default restock count value (PSBMINFREEBDS).
<i>Dump =</i>	NA.

Table 3 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	Interworking data arriving at the PHV is loaded into a set of BDs linked to an RFD. The received frame is processed at Interject and put into a queue of BDs that are associated with a specific <i>AUTOPLEX</i>TM data trunk (ADT). The accumulated BDs make up a BD headcell structure in the application/<i>AUTOPLEX</i>TM data control block (ADCB). This error indicates that the BD head or tail pointers are invalid.
Possible Error Effect:	This error can cause BDs to be lost.
Corrective Action Taken:	The head and tail pointers are re-initialized to PSNULLADDR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Contents of the BD tail pointer.
<i>Bad Data =</i>	Contents of the BD head pointer.
<i>Good Data =</i>	Address of the first BD GP (&PSbdgp[0]).
<i>Logical Key =</i>	The index into the ADCB.
<i>Dump =</i>	NA.

Table 4 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	In a PH3 both the AP and IOP processors use the same free list (GPBDs). Furthermore, GPBDs which have been removed from this list but are still queued up waiting to be processed are marked as being "owned" by the free list (BD.bd_owner = OCCH_FREE). Thus, overload control's counts of the number of free BDs may be inflated. To counteract this three counters are kept (PSbapfrcnt, PS_iscb.ipszcnt and PS_iscb.ipretcnt) whose sum equals the number of BDs currently queued. This error results when the actual number of BDs on the free list (as calculated by the audit) + the number of BDs queued differs from OCgpcnt[OCCH_FREE] (OC's count of GPBDs marked as being owned by the free list) by more than PSBDCHMX. The tolerance is there to allow for the fact that the IOP may be manipulating BDs concurrently with the audit.
Possible Error Effect:	May result in erroneous overload control states.

Corrective Action Taken:	The audit will add or subtract the appropriate amount from PSbapfrnt to make all the values consistent.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	OC's count of free GPBDs (OCgpcnt[OCCH_FREE]).
<i>Bad Data =</i>	The number of queued BDs.
<i>Good Data =</i>	The actual number of BDs on the free list.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 5 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The audit encountered one of two conditions of a LAN that prevents the BD audit from collecting lost BDs and verifying the overload control counts. The first condition is that a LAN was in the suspended state, and the head of the LAN's RBD pool could not be located. This situation can occur if all of a LAN's RFDs are in use and the LAN had an incoming frame for which it needed an RFD. In this case, only the LAN has access to the head of its RBD pool. This is a self correcting situation in that the operational code will resume the LAN as it frees up RFDs by unloading frames. This condition implies that the BD audit can only traverse the other BD linked lists, but can not reclaim lost BDs nor verify the overload control buffer usage counts. The other condition is that while traversing a LAN's RFD (circular) list, the BD audit encountered an RFD with an out of range or misaligned next RFD offset. It is unlikely that the BD audit will encounter this situation before the operational code does. NOTE: If the operational code encounters this situation, an assert is called that will result in an initialization of packet switching data structures that will clear up this problem.
Possible Error Effect:	This error report is only meant to document self-correcting situations.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Has a value of 1 if the condition found is a bad RFD linkage. Has a value of 2 if the condition found is that of a LAN in the suspended state.
<i>Bad Data =</i>	If the condition found was a bad RFD linkage, this is the value found as the next RFD offset, in the RFD. Otherwise, this is significant.
<i>Good Data =</i>	If the condition found was a bad RFD linkage, this is the number of RFDs that had been traversed, excluding the one with the bad linkage.
<i>Logical Index =</i>	The LAN side that is in the suspended state.
<i>Dump =</i>	NA.

Table 6 Error Code: EVEN_COUNT

Error Code:	EVEN_COUNT
Error Description:	A LAN was found whose RBD pool does not have the expected number of BDs.
Possible Error Effect:	If the RBD pool was in fact out of BDs, then the LAN could go into the "no resource" state. Otherwise, this error condition is a warning that the designed allocation of BDs has been disturbed, and that subsequent GP BD shortages may occur.
Corrective Action Taken:	If the RBD pool had less than the designed allocation of BDs and there are enough BDs in the GP BD pool, the audit will attempt to replenish RBD pool. If the RBD pool had more than the designed allocation, the audit will update the value of IMlbdelta[] to indicate the number of surplus BDs in the RBD pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The length of the RBD pool as was computed by the audit.

<i>Good Data</i> =	The number of BDs by which the pool was over (positive value) or under (negative value).
<i>Logical Index</i> =	The LAN side whose RBD pool is not of the right length.
<i>Dump</i> =	NA.

Table 7 Error Code: INV

Error Code:	INV
Error Description:	The last BD offset of a LAN's RBD pool, or the GP BD pool, of the SP BD pool that did not match the BD, that was found by the audit at the end of the pool.
Possible Error Effect:	This error can cause BDs to be lost, since the LAN RBD pool replenishment function or the BD returning function most likely had been using an erroneous "last BD" offset variable.
Corrective Action Taken:	The "last BD" offset variable is set to contain the offset of the BD found by the audit at the end of the pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value that identifies which pool had an inconsistent last BD offset. 1 = A LAN RBD pool. 2 = The GP BD pool. 6 = The SP BD pool.
<i>Bad Data</i> =	The erroneous "last BD" offset.
<i>Good Data</i> =	The offset of the BD found by the audit at the end of the LAN RBD pool.
<i>Logical Index</i> =	If the error address is 1, this is the LAN side whose "last RBD" offset is in error. Otherwise, the logical address is not meaningful.
<i>Dump</i> =	The number of BDs in the RBD pool as counted by the audit. Read only the high-order (16-bit) halfword.

Table 8 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	The audit found a general purpose (GP) BD that was untouched, and therefore lost.
Possible Error Effect:	If this error condition occurs often enough, it may ultimately result in BD shortage or even exhaustion.
Corrective Action Taken:	The lost BD is put back into the GP BD pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index into the array of GP BDs that identifies the lost BD found by the audit.
<i>Dump</i> =	The contents of the lost GP BD.

Table 9 Error Code: LIST

Error Code:	LIST
Error Description:	The BD audit was unable to audit the free list of GPBDs in a PH due to contention with the IOP while trying to access the head of the list.
Possible Error Effect:	The audit will be unable to complete its checks. These include audits of overload control levels and checks for lost BDs.
Corrective Action Taken:	The audit will complete without taking any further action.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the first BD on the free list (PH3: PS_iscb.fbdb_ptr->bd_head PHV & PHA: PSadcb[.iaqueue.bdchain.bd_head)
<i>Bad Data</i> =	Offset of the first BD on the free list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.

<i>Dump</i> =	NA.

Table 10 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a SPYDER BD that was untouched, and therefore lost.
Possible Error Effect:	If this error condition occurs often enough, it may ultimately result in BD shortage or even exhaustion.
Corrective Action Taken:	The lost BD is put back into the SP BD pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index into the array of SP BDs that identifies the lost BD found by the audit.
<i>Dump</i> =	The contents of the lost SP BD.

Table 11 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A BD was found whose buffer address is out of range, or whose next BD offset is out of range or alignment.
Possible Error Effect:	This error can cause illegal memory access (read or write) by the code that is resident in the port processor, the LAN, or the SPYDER/SPORT processors.
Corrective Action Taken:	If the BD's buffer address is out of range, it will be changed to contain the address of the beginning of its assigned buffer. Note, that the user bits are preserved. If the BD's next BD offset is out of range or alignment, it will be corrected to the null offset value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value that identifies the type of BD. 13 = A GP BD. 14 = A SP BD.
<i>Bad Data</i> =	The bad buffer address or next BD offset as indicated by <i>Good Data</i> .
<i>Good Data</i> =	An indication of the type of error found (that is, 4 if the next BD offset was anomalous or 5 if the buffer address was out of range).
<i>Logical Index</i> =	The index into the appropriate BD array that identifies the BD.
<i>Dump</i> =	The contents of the BD and its buffer.

Table 12 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The audit found a physical PH channel whose overload control usage count of general purpose (GP) buffers was in error.
Possible Error Effect:	May result in erroneous overload control states.
Corrective Action Taken:	The audit corrects the count found to be in error with the difference between the bad value and the value as recomputed by the BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of the overload control (GP) buffer usage count that was found to be in error.
<i>Good Data</i> =	The value of the overload control buffer usage count as recomputed by the audit.
<i>Logical Index</i> =	The ID of the physical PH channel whose count was found to be in error.
<i>Dump</i> =	NA.

Table 13 Error Code: RMCOUNTS

Error Code:	RMCOUNTS
Error Description:	The audit found a physical PH channel whose overload control usage count of

	SPYDER (SP) buffers was in error.
Possible Error Effect:	May result in erroneous overload control states.
Corrective Action Taken:	The audit corrects the count found to be in error with the difference between the bad value and the value as recomputed by the BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value of the overload control (SP) buffer usage count that was found to be in error.
<i>Good Data =</i>	The value of the overload control buffer usage count as recomputed by the audit.
<i>Logical Index =</i>	The id of the physical PH channel whose count was found to be in error.
<i>Dump =</i>	NA.

Table 14 Error Code: TRAVERSE

Error Code:	TRAVERSE
Error Description:	The audit found an anomaly in a linked list of BDs that it was traversing. The list and error type can be determined using the data dumped by the audit.
Possible Error Effect:	Varies according to conditions and the type of error found. However, some possibilities are: BD shortages, resource exhaustion or overload control states.
Corrective Action Taken:	The audit will truncate the list of BDs in which the error was found. If the BD list is owned by an ALCB, LCCB or IFD, the owning structure will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	A value indicating what type of data structure the list was owned by. 1 = Verification of the LAN RBD Pool. The <i>Logical Index</i> represents the LAN side. 2 = Verification of the GP BD Pool. The <i>Logical Index</i> is not relevant. 3 = PH2 or PI: Verification of BDs attached to LAN TCBs (PS10txblk[] or PS11txblk[]). PH3: Verification of BDs attached to LAN TFDs (PS10tfd[] or PS11tfd[]). 4 = Verification of BDs attached to LAN RFDs (PS10rfd[] or PS11rfd[]). 5 = Verification of BDs attached to IFDs (PS1fd[]). 6 = Verification of BDs attached to SPYDER BD Pool. The <i>Logical Index</i> is not relevant. 7 = Verification of BDs attached to SPYDER RFDs (PSs_rfd[]). 8 = Verification of BDs attached to SPYDER TFDs (PSs_tfd). 9 = Verification of BDs attached to ALCBs (PSalcb[]). 10 = Verification of BDs attached to LCCBs (PSlccb[]).
<i>Bad Data =</i>	The length of the free list if that was the list being traversed. Otherwise, none.
<i>Good Data =</i>	A value that indicates the type of error found by the audit. 4 = A BD in the BD list being traversed was found whose next BD offset was out of range.

	7 = A BD in the BD list being traversed was found to have an owner that was inappropriate for the BD list, (such as, a BD whose owner is "free" but is attached to an LCCB). 8 = A BD list (other than any of the BD pools) whose length exceeded the maximum allowed for a valid packet/frame.
<i>Logical Index</i> =	The index of the owning data structure unless otherwise specified in "Error Address."
<i>Dump 1</i> =	The contents of the last error-free BD in the list. This dump can be a dummy BD that is used by the audit to facilitate its traversal of the list.
<i>Dump 2</i> =	The contents of the BD in error.
<i>Dump 3</i> =	The contents of the buffer associated with the BD in error.

Table 15 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Resets and/or ASSERTS.
Corrective Action Taken:	If the audit was traversing lists belonging to LCCBs, ALCBs or IFDs at the time of the error, the appropriate structure will be idled. In any case, the audit will not do any more checks during this invocation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value indicating the type of work the BD audit was doing at the time the error occurred. Note that the audit may have been finished with the work item indicated and is in the process of moving on to the next task. 1 = Verification of the LAN RBD Pool. The <i>Logical Index</i> represents the LAN side. 2 = Verification of the GP BD Pool. The <i>Logical Index</i> is not relevant. 3 = PH2 or PI: Verification of BDs attached to LAN TCBs (PS10txblk[] or PS11txblk[]). PH3: Verification of BDs attached to LAN TFDs (PS10tfd[] or PS11tfd[]). 4 = Verification of BDs attached to LAN RFDs (PS10rfd[] or PS11rfd[]). 5 = Verification of BDs attached to IFDs (PS1fd[]). 6 = Verification of BDs attached to SPYDER BD Pool. The <i>Logical Index</i> is not relevant. 7 = Verification of BDs attached to SPYDER RFDs (PSs_rfd[]). 8 = Verification of BDs attached to SPYDER TFDs (PSs_tfd[]). 9 = Verification of BDs attached to ALCBs (PSalcb[]). 10 = Verification of BDs attached to LCCBs (PSlccb[]). 11 = Verification of the overload control buffer usage counts.

	12 = Search for and reclamation of any lost BDs. 13 = Range checking GPBDs. 14 = Range checking SPBDs. 0xffffffff = The BD audit has not started any task yet.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	If the audit was working on an array of data structures, this is the index of the element in the array which was being accessed at the time the error occurred. Exceptions are noted in <i>Error Address</i>.
<i>Dump =</i>	NA.

BFDATA

Software Release: 5E10 and later

1. BUFFER DATA (BFDATA) AUDIT

Some recent change relations in the office-dependent data (ODD) in the administrative module (AM) have been moved from the AM to disk storage to increase the amount of available AM memory space. The disk access mechanism uses a direct interface with the disk driver (DKDRV) kernel process to transfer data between buffers in AM memory and data base partitions on disk. This mechanism reduces overhead by bypassing the file manager (FMGR). Recent change transactions work by copying blocks from disk into the buffer data area in AM memory. Ten buffers are allowed for each transaction. Disk updates are done in the AM. The buffer data audit verifies the transfer of data between disk and AM main memory by checking the contents of the buffer data control structure stored in the AM. The audit runs in the segmented modes in the OKP environment in the AM.

2. DATA STRUCTURES

The buffer data audit verifies the buffer data control structure relation (RLDB_BFDAT), a dynamic relation in the AM. The fields checked within this relation include the buffer ID (bufid), the disk ID (dskid), the write flag (buf_wtflag), and the page state (pgstate). The write flag tells whether a tuple is in the write-protected used state. The page state field may assume one of four values: idle, used (read-only), released, or touched. Used rldb_BFDAT tuples are linked to the data base coupler relation (RLDB_CUPLR). The linkage between rldb_BFDAT and rldb_CUPLR is null for all other states.

There is a page table global array containing buffer IDs associated with the rldb_BFDAT tuple on which the audit makes consistency checks. The index into the page table is the disk ID.

The audit also makes linkage checks between RLDB_BFDAT and its owner relation, the transaction dictionary (RLDB_TRNDC). The RLDB_TRNDC relation contains head cells for three page states: idle, used, and released. Figure 1 shows these data structures.

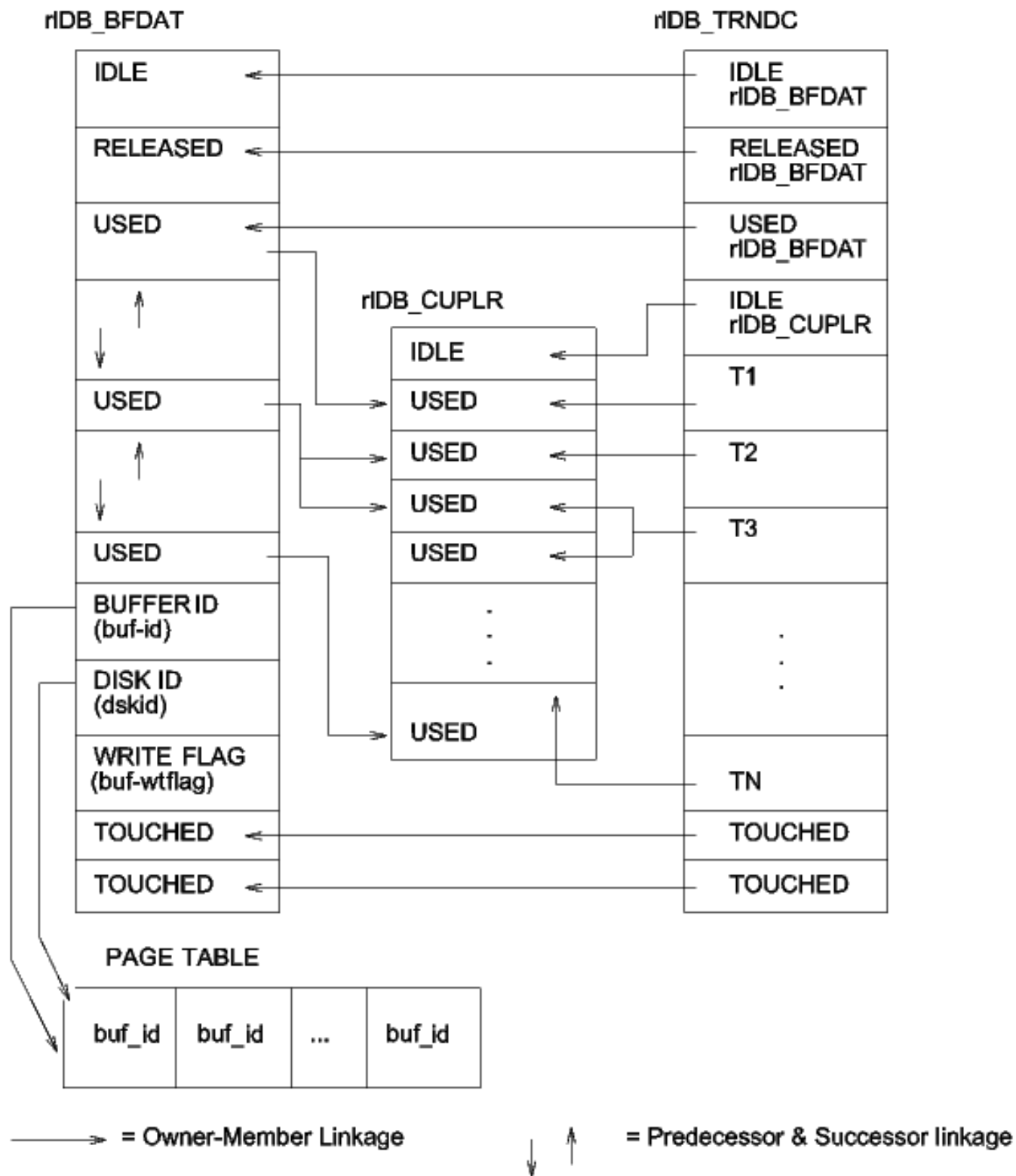


Figure 1 (BFDATA-1) Data Structures of the BFDATA Audit

3. AUDIT CHECKS

The audit calls the linkage driver (DRLINK) to check the linkages between RLDB_BFDAT, RLDB_TRNDC, and RLDB_CUPLR. The linkage driver then calls the semantic check function to make the following checks.

- (a) For non-idle pages, the audit ensures that the disk ID and buffer ID combination listed in the page table array agrees with the disk ID and buffer ID combination in RLDB_BFDAT.
- (b) The audit uses the page state field to ensure that the rIDB_BFDAT tuple is linked to the correct rIDB_TRNDC head cell (for example, tuples with an idle page state should be linked to the idle head cell in rIDB_TRNDC).
- (c) If the page state is not used, the other fields in the tuple should be in their initialized state and the linkage to rIDB_CUPLR should be nul.
- (d) If the page state is used, the audit checks to ensure that the tuple is also linked to the rIDB_CUPLR relation (see the DBCUPLR audit).
- (e) If the page state is touched, the audit makes a member-owner check to ensure that the key of the touched tuple is in the proper range in rIDB_TRNDC.

4. ERROR HANDLING

If DRLINK detects a linkage error or the semantic check function detects an error, the error is reported and the tuple in RLDB_BFDAT is initialized and linked to the appropriate idle list. The page table array entry is cleared.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_DSKID

Error Code:	BAD_DSKID
Error Description:	The diskid in this buffer control block does not agree with the diskid in the page table.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 2 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	An rIDB_BFDAT tuple with a BUFRELEASE state is not linked to the RELEASE list.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.

<i>Dump</i> =	rIDB_BFDAT tuple.

Table 3 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An rIDB_BFDAT tuple with a BUFUSED state is not linked to the USED list.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 4 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A rIDB_BFDAT tuple, that is not in the BUFUSED page state, is linked to rIDB_CUPLR. This is only valid for the BUFUSED case which keeps track of all the transaction IDs that are associated with this tuple.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 5 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	Depending on the given page state, some of the other semantic data on a rIDB_BFDAT tuple contains invalid information.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 6 Error Code: STATE

Error Code:	STATE
Error Description:	An rIDB_BFDAT tuple with a BUFTOUCH state is not linked on the TOUCH list.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 7 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	An rIDB_BFDAT tuple with a BUFIDLE state is not linked on the IDLE list.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 8 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The page state of an rIDB_BFDAT tuple is invalid.
Possible Error Effect:	The rIDB_BFDATA tuple contains wrong information. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

Table 9 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rIDB_BFDAT tuple in the BUFUSED (read-only) state has been in a transient state for more than five minutes.
Possible Error Effect:	A transaction in progress is stuck in a transient state and cannot terminate. Disk image cannot be updated correctly.
Corrective Action Taken:	The rIDB_BFDATA tuple is recovered, reinitialized, and put back on the idle list. The page table entry is also reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_BFDAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_BFDATA tuple.
<i>Dump</i> =	rIDB_BFDAT tuple.

BLOCK

Software Release: 5E10 and later

1. SM PERIPHERAL FAULT RECOVERY ERROR BLOCK (BLOCK) AUDIT

SM peripheral fault recovery consists of a group of programs that react to errors related to switching module (SM) peripheral circuits such as line units and trunk units. The errors are recorded in peripheral error blocks (such as, tuples of RLSMPERBLK), which contain the name of the circuit that failed, the type of failure, and the time when the error last occurred. The error blocks also contain a current error count for this type of error on the given circuit. This count is used as a "leaky bucket". That is, the count is incremented each time the error being tracked occurs on the circuit involved. If a specified number of errors have accumulated in a specific amount of time, the circuit is usually taken out of service. If the error has not occurred in a given amount of time, the count will be decremented. If the count reaches zero, the error block is released.

The BLOCK audit runs in the OSDS-M environment in the SM in either segmented or directed mode.

2. DATA STRUCTURES

The BLOCK audit verifies the following attributes of tuples of relation RLSMPERBLK:

audtouch	= Touch bit for "INV" checks.
bckerrblk	= Back list linkage.
ctkid	= ID of the circuit (if not idle) the tuple is in use for.
count	= Count of the errors of this type on this circuit.
deltatime	= The amount of time sufficient to cause a decrement of the error count if the error has not recurred (such as, the size of the "leaky bucket").
erblk	= The tuple's key.
errtype	= The type of error being tracked by this tuple.
fwderrblk	= Forward list linkage.
timest	= Time the last error of this type occurred on this circuit.

All tuples of RLSMPERBLK should either be found on the idle list (tuple 0 of the relation acts as the list owner) or linked in a list attached to a tuple of RLSMEST associated with the circuit found in the "ctkid" field. Refer to Figure 1

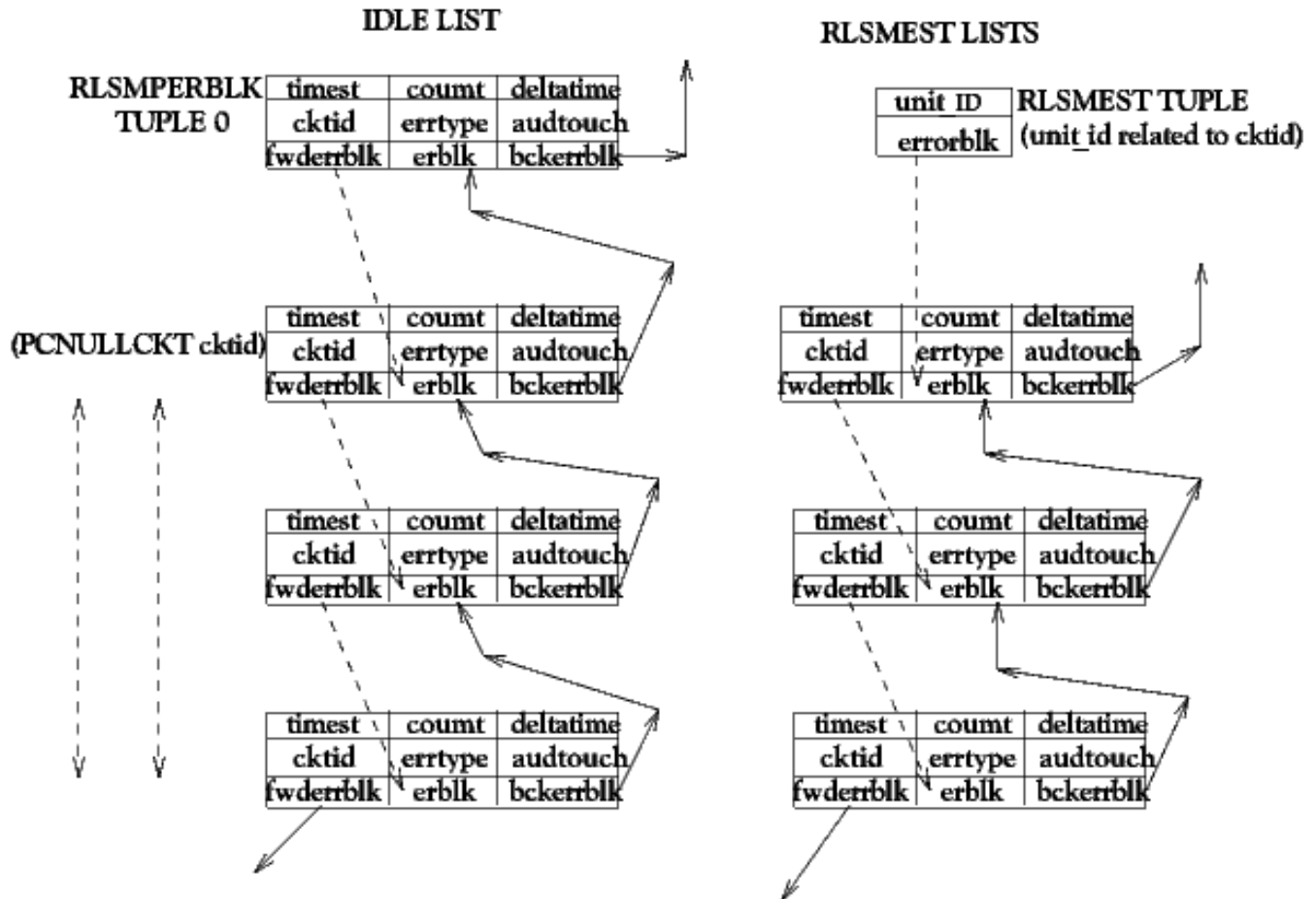


Figure 1 (BLOCK-1) RLSMPERBLK Lists

3. AUDIT CHECKS

The BLOCK audit invokes the DRDYN driver to step through the RLSMPERBLK relation. The driver also provides ZAP error checking. For each tuple, the BLOCK audit makes the following checks:

BLNKINQUE = Tuple's back linkage.
 KEY = Tuple's key.
 PT2PT = Tuple's forward linkage.
 TYPE = Tuple's associated error type.

If the tuple has a non-null circuit ID in it:

TIMEOUT = Size of the tuple's leaky bucket.
 TIMESTAMP = Timestamp of the associated error.

If the tuple has a null circuit ID in it:

INV = Lost tuples.

It should also be noted that if the audit finds a tuple whose "count" field is ≤ 0 (or would be if it were updated relative to the timestamp of the last occurrence of the error it tracks), the audit will silently return it to the idle list.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	The back link of the rISMPERBLK tuple being checked (bckerrblk), contains the key of an rISMPERBLK tuple whose forward link (fwderrblk) is not equal to the key of the tuple being checked (erblk).
Possible Error Effect:	If the tuple being checked is in a list, the list is broken. For the idle list this could mean resource failures due to lost tuples. For a list attached to a tuple of RLSMEST this could mean a failure on the part of fault recovery to report an error and take action.
Corrective Action Taken:	The tuple in error is returned to the RLSMPERBLK idle list. Any tuples found linked to it (using its "fwderrblk" field) are also idled since, by implication, they are also lost. The counts for any error being tracked by the tuple being checked (or tuples attached to it) will be lost.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The tuple's forward linkage (rISMPERBLK.fwderrblk).
<i>Bad Data =</i>	The amount of time (in milliseconds) since the fault recovery error (if any) associated with this tuple occurred (from rISMPERBLK.timest).
<i>Good Data =</i>	The tuple's circuit ID (rISMPERBLK.cktid).
<i>Logical Index =</i>	The tuple in error's key (rISMPERBLK.erblk).
<i>Dump 1 =</i>	The tuple keyed by "Logical Key".
<i>Dump 2 =</i>	The tuple keyed by the "fwderrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 3 =</i>	The tuple keyed by the "bckerrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 4 =</i>	A count of the number of tuples idled.

Table 2 Error Code: INV

Error Code:	INV
Error Description:	An rISMPERBLK tuple with a null circuit ID (cktid = PCNULLCKT) was found not linked to any list.
Possible Error Effect:	If this happened to enough rISMPERBLK tuples, eventually fault recovery would not be able to track errors (and thus be unable to report them or act on them).
Corrective Action Taken:	The tuple is returned to the RLSMPERBLK idle list. Any tuples found linked to the tuple in error (through its "fwderrblk" field) are also idled since, by implication, they are also lost.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The tuple's forward link (rISMPERBLK.fwderrblk).
<i>Bad Data =</i>	The amount of time (in milliseconds) since the fault recovery error (if any) associated with this tuple occurred (from rISMPERBLK.timest).
<i>Good Data =</i>	The tuple's circuit ID (rISMPERBLK.cktid).
<i>Logical Index =</i>	The tuple in error's key (rISMPERBLK.erblk).
<i>Dump 1 =</i>	The tuple keyed by "Logical Key".
<i>Dump 2 =</i>	The tuple keyed by the "fwderrblk" field of the tuple in error. If the linkage was

	null, this will be tuple 0, the head of the idle list.
<i>Dump 3 =</i>	The tuple keyed by the "bckerrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 4 =</i>	The number of tuples idled.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	An rISMPERBLK tuple's key (erblk) is not correct.
Possible Error Effect:	Fault recovery cannot reliably access this tuple and so may not peg (or report or react to) error counts correctly.
Corrective Action Taken:	The tuple is unlinked from any list it is in and is returned to the RLSPERBLK idle list. The counts for any error being tracked by the tuple being checked will be lost.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISMPERBLK tuple in error.
<i>Bad Data =</i>	The key found in the tuple (rISMPERBLK.erblk).
<i>Good Data =</i>	The correct value of the key.
<i>Logical Index =</i>	The correct value of the key.
<i>Dump =</i>	The rISMPERBLK tuple in error.

Table 4 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	The forward link (fwderrblk) of the rISMPERBLK tuple being checked contains the key of an rISMPERBLK tuple whose back link (bckerrblk) is not equal to the key (erblk) of the tuple being checked.
Possible Error Effect:	If this tuple is in a list, the list is broken. For the idle list this could mean resource failures due to lost tuples. For a list attached to a tuple of RLSPERBLK this could mean a failure on the part of fault recovery to report an error and take action.
Corrective Action Taken:	The tuple is unlinked from any list it is in and is returned to the RLSPERBLK idle list. Note that the process of checking the linkage will cause the tuple's "fwderrblk" field to get set to null (0) if it is in error. The counts for any error being tracked by the tuple being checked will be lost.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The forward linkage (rISMPERBLK.fwderrblk) prior to the audit check.
<i>Bad Data =</i>	The amount of time (in milliseconds) since the fault recovery error (if any) associated with this tuple occurred (from rISMPERBLK.timest).
<i>Good Data =</i>	The tuple's circuit ID (rISMPERBLK.cktid).
<i>Logical Index =</i>	The tuple in error's key (rISMPERBLK.erblk).
<i>Dump 1 =</i>	The tuple keyed by <i>Logical Key</i> .
<i>Dump 2 =</i>	The tuple keyed by the "fwderrblk" field as given in the <i>Error Address</i> . If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 3 =</i>	The tuple keyed by the "bckerrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.

Table 5 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	The size of the "leaky bucket" associated with the tuple being checked (rISMPERBLK.deltatime) is ≤ 0 .
Possible Error Effect:	The error count held in this tuple ("count") will not be updated correctly. This could cause fault recovery to either take action when it should not or fail to take action when it should.
Corrective Action Taken:	The tuple is unlinked from any list it is in and is returned to the RLSPERBLK idle list. The counts for any error being tracked by the tuple being checked will be lost.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The forward link (rlSMPERBLK.fwderrblk) of the tuple in error.
<i>Bad Data =</i>	The amount of time (in milliseconds) since the fault recovery error (if any) associated with this tuple occurred (from rlSMPERBLK.timest).
<i>Good Data =</i>	The tuple's circuit ID (rlSMPERBLK.cktid).
<i>Logical Index =</i>	The tuple in error's key (rlSMPERBLK.erblk).
<i>Dump 1 =</i>	The tuple keyed by <i>Logical Key</i> .
<i>Dump 2 =</i>	The tuple keyed by the "fwderrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 3 =</i>	The tuple keyed by the "bckerrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.

Table 6 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The time stamp of the last occurrence of the error associated with the tuple being checked (rlSMPERBLK.timest) is out of range (less than 0 or greater than one day).
Possible Error Effect:	Fault recovery may incorrectly evaluate the status of the error this tuple is being used to track. This could cause fault recovery to either take action when it should not or fail to take action when it should.
Corrective Action Taken:	The tuple is unlinked from any list it is in and is returned to the RLSMPERBLK idle list. The counts for any error being tracked by the tuple being checked will be lost.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The forward linkage (rlSMPERBLK.fwderrblk) of the tuple in error.
<i>Bad Data =</i>	The amount of time (in milliseconds) since the fault recovery error (if any) associated with this tuple occurred (from rlSMPERBLK.timest).
<i>Good Data =</i>	The tuple's circuit ID (rlSMPERBLK.cktid).
<i>Logical Index =</i>	The tuple in error's key (rlSMPERBLK.erblk).
<i>Dump 1 =</i>	The tuple keyed by "Logical Key".
<i>Dump 2 =</i>	The tuple keyed by the "fwderrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 3 =</i>	The tuple keyed by the "bckerrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.

Table 7 Error Code: TYPE

Error Code:	TYPE
Error Description:	The type of error held in the tuple being checked is out of range (rlSMPERBLK.errtype $\geq$ SMLASTERR).
Possible Error Effect:	Fault recovery would be unable to determine the action to be taken if the "count" value in this tuple reached its threshold (such as, if the "leaky bucket" overflowed).
Corrective Action Taken:	The tuple is unlinked from any list it is in and is returned to the RLSMPERBLK idle list. The counts for any error being tracked by the tuple being checked will be lost.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The forward linkage (rlSMPERBLK.fwderrblk) of the tuple in error.
<i>Bad Data =</i>	The amount of time (in milliseconds) since the fault recovery error (if any) associated with this tuple occurred (from rlSMPERBLK.timest).
<i>Good Data =</i>	The tuple's circuit ID (rlSMPERBLK.cktid).
<i>Logical Index =</i>	The tuple in error's key (rlSMPERBLK.erblk).
<i>Dump 1 =</i>	The tuple keyed by "Logical Key".
<i>Dump 2 =</i>	The tuple keyed by the "fwderrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.
<i>Dump 3 =</i>	The tuple keyed by the "bckerrblk" field of the tuple in error. If the linkage was null, this will be tuple 0, the head of the idle list.

BRCSDb

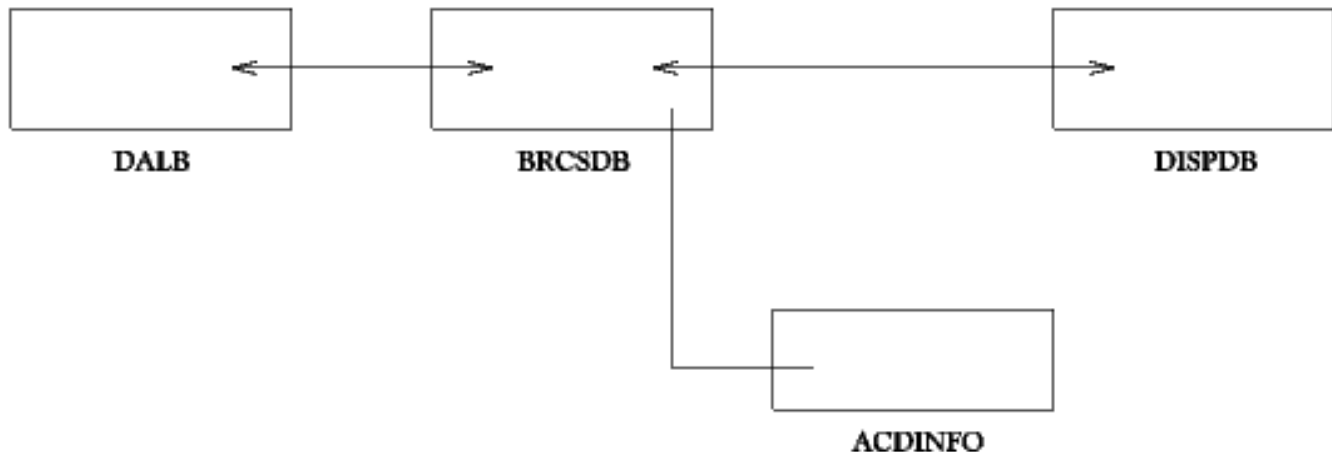
Software Release: 5E10 and later

1. BRCS DATA BLOCK (BRCSDb) AUDITS

The RLBRCSDb relation is used by integrated services digital network (ISDN) business and residential custom services (BRCS) applications to contain some BRCS related attributes such as call appearance, shared DN, or associated terminal. Idle tuples in the RLBRCSDb relation are linked to a specific tuple in the RLDB_HEADC relation. When originating from or terminating to ISDN BRCS line, an rLBRCSDb tuple will be retrieved from idle list and linked to rLDALB and rLDISPDB (if the terminal has display capability).

2. DATA STRUCTURES

The data structure checked by the BRCSDb audit is the BRCS data block relation (RLBRCSDb). Figure 1 illustrates the layout of the relation. RLBRCSDb has an owner-member linkage with RLDB_HEADC and one-to-one linkage with RLDALB and RLDISPDB.



BRCSDb tuple used for calling has one-to-one linkage between DALB, DISPDB and ACDINFO.

—————> = Owner-Member Linkage.
 <-----> = One to One Linkage.
 ————— = Non-standard Linkage.

Figure 1 (BRCSDb-1) BRCSDb Data Structure

3. AUDIT CHECKS

This audit runs in the initialization and segmented modes. The BRCSDb audit calls the DRINASEK driver to step through the tuples, validating keys and linkages. The following semantic checks are performed by this audit:

- Verify that an idle rLBRCSDb tuple only links to the correct rLDB_HEADC tuple for the idle list and does not link to other data structures.

- If there is no process owner for the rDALB tuples associated with this rIBRCSDB tuple, verify that the rIBRCSDB is shared.
- Verify that the logical terminal index contained in the rIBRCSDB is consistent with that in the rIPCBLA (process control linkage area).
- If the *acdinfokey* field in the rIBRCSDB tuple does not have a value of RINULLKEY, then verify that this field contains the logical key of a valid rIACDINFO tuple.
- If the *acdinfokey* field does point to a valid rIACDINFO tuple, verify that this rIACDINFO tuple also has its *brcsdbkey* field equal to the logical key of the rIBRCSDB tuple under audit.

4. ERROR HANDLING

If any key linkage or semantic error is encountered by this audit, the tuple in error will be recovered back to the idle list. See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: APIDKEY

Error Code:	APIDKEY
Error Description:	The audit found an rIBRCSDB tuple whose <i>acdinfokey</i> field is not RINULLKEY (such as, zero). However, the <i>acdinfokey</i> field does not point to a valid rIACDINFO tuple.
Possible Error Effect:	Possible loss of the rIACDINFO tuple resource.
Corrective Action Taken:	The <i>acdinfokey</i> field in the rIBRCSDB tuple is set to RINULLKEY. The ACDINFO related audit is scheduled to clean up the associated rIACDINFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIBRCSDB tuple.
<i>Bad Data</i> =	The value of the <i>acdinfokey</i> in the rIBRCSDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rIBRCSDB tuple.
<i>Dump</i> =	The rIBRCSDB tuple.

Table 2 Error Code: BRCSLINK

Error Code:	BRCSLINK
Error Description:	Invalid linkage found in an rIBRCSDB tuple.
Possible Error Effect:	The tuple may not be able to be accessed. Calls that may be associated with this tuple will be torn down.
Corrective Action Taken:	Zap the tuple and place it on the RLDB_HEADC idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIBRCSDB tuple.
<i>Bad Data</i> =	rIBRCSDB <i>brcsdb_mem</i> linkage field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIBRCSDB key.
<i>Dump</i> =	rIBRCSDB tuple.

Table 3 Error Code: BRCS SHARE

Error Code:	BRCS SHARE
Error Description:	It is possible that there may be no rIPCBLA linked to the rDALB in the case of a shared call. The rDALB link shows no rIPCBLA and the shared bit in the rIBRCSDB is not set.
Possible Error Effect:	A shared call may be lost or may be hung. It may not be possible to perform shared calls.

Corrective Action Taken:	The shared attribute will be cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i>	Address of the rIBRCSDB tuple.
<i>Bad Data</i> =	Attribute shared in rIBRCSDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIBRCSDB key.
<i>Dump</i> =	rIBRCSDB tuple.

Table 4 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Based on the value of the <i>acdinforekey</i> field in the rIBRCSDB tuple, the audit found a valid corresponding rIACDINFO tuple. However, the <i>brcsdbkey</i> in this rIACDINFO tuple does not point back to the rIBRCSDB tuple under audit.
Possible Error Effect:	Possible loss of the rIACDINFO tuple resource.
Corrective Action Taken:	The <i>acdinforekey</i> field in the rIBRCSDB tuple is set to RINULLKEY (such as, zero). The ACDINFO related audit is scheduled to clean up the associated rIACDINFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIBRCSDB tuple.
<i>Bad Data</i> =	The value of the <i>acdinforekey</i> in the rIBRCSDB tuple.
<i>Good Data</i> =	The value of the <i>brcsdbkey</i> in the rIACDINFO tuple.
<i>Logical Index</i> =	The logical key of the rIBRCSDB tuple.
<i>Dump 1</i> =	The rIBRCSDB tuple.
<i>Dump 2</i> =	The rIACDINFO tuple.

Table 5 Error Code: INDEX

Error Code:	INDEX
Error Description:	Index in rIBRCSDB is inconsistent with index in rIPCBLA.
Possible Error Effect:	Switch may not communicate with customer premise equipment. Calls may be incorrectly set up, and may be torn down.
Corrective Action Taken:	Zap the tuple, recovering the index field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i>	Address of the rIBRCSDB tuple.
<i>Bad Data</i> =	rIBRCSDB ltm_idx attribute.
<i>Good Data</i> =	rIPCBLA ltm_idx attribute.
<i>Logical Index</i> =	rIBRCSDB key.
<i>Dump</i> =	rIBRCSDB tuple.

Table 6 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	The rIDALB tuple corresponding to this rIBRCSDB tuple (where rIDALB.dalb = rIBRCSDB.brcsdb_lnk.lnk \em DBSFDALB) is not correctly linked to an rIPCBLA tuple. The rIDALB tuple's BRCS call reference number (rIDALB.cr_lch), if not 0, indicates that this RLDALB should be linked to an associated rIPCBLA tuple, such that rIPCBLA.procnum = (rIDALB.dalb_lnk.lnk - DBSFPCBLA) or (rIDALB.dadpm.st_link - DBSFPCBLA).
Possible Error Effect:	The switch may be unable to communicate with customer premises equipment. Calls may be incorrectly set up, and incorrectly torn down.
Corrective Action Taken:	The rIDALB and rIBRCSDB dynamic tuples will both be recovered to idle conditions, and the call terminal process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i>	The address of the rIPCBLA tuple which should have been linked to the rIDALB tuple.
<i>Bad Data</i> =	The key of the rIPCBLA tuple (procnum) which should have been linked to the rIDALB tuple.
<i>Good Data</i> =	The logical key of the rIPCBLA tuple which was calculated from the rIDALB tuple (rIDALB.dalb_lnk.lnk - DBSFPCBLA or rIDALB.dadpm.st_link - DBSFPCBLA),

	whichever was found in a non-zero condition. If both were found equal to zero, then the latter will be dumped).
<i>Logical Index</i> =	The rIBRCSDB tuple's key (brcsdb).
<i>Dump 1</i> =	The contents of the rIBRCSDB tuple.
<i>Dump 2</i> =	The contents of the rIPCBLA tuple.

Table 7 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An rIBRCSDB tuple is in the idle state but some of its other linkages are nonzero.
Possible Error Effect:	Some resource is linked to an rIBRCSDB tuple on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rIBRCSDB tuple and zeroes the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i>	Address of the rIBRCSDB tuple.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	rIBRCSDB key.
<i>Dump</i> =	rIBRCSDB tuple.

BRGDB

Software Release: 5E14 and later

1. BRIDGING DATA BLOCK (BRGDB) AUDIT

A bridging data block is used by peripheral control to store the state and type of bridge that is involved with a bridging call.

2. DATA STRUCTURES

For details on the fields contained in the RLBRGDB relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rIBRGDB tuple is in a duplex bridging state (rIBRGDB.brgstat), but the bridging type (rIBRGDB.brgtype) is not conference circuit bridging or silent bridging.
Possible Error Effect:	May cause the bridge call to fail.
Corrective Action Taken:	The audit reinitializes the rIBRGDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIBRGDB tuple.
<i>Bad Data =</i>	Bridge type (rIBRGDB.brgtype).
<i>Good Data =</i>	Bridge state (rIBRGDB.brgstat).
<i>Logical Index =</i>	Logical key (rIBRGDB.brgnum) of the rIBRGDB tuple.
<i>Dump =</i>	rIBRGDB tuple.

Table 2 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	The rIBRGDB tuple is in a busy state but is not linked to a bridging rCCBCOM tuple.
Possible Error Effect:	May cause asserts or interrupts by the application which will eventually cause the bridging call to fail.
Corrective Action Taken:	The audit reinitializes the rIBRGDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIBRGDB tuple.
<i>Bad Data =</i>	Physical key of the owning rCCBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (rIBRGDB.brgnum) of the rIBRGDB tuple.
<i>Dump =</i>	rIBRGDB tuple.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK

Error Description:	The rIBRGDB tuple is linked onto the idle list but one of the non-idle list linkages is not null (DALNKNUL).
Possible Error Effect:	May cause lost resources or asserts.
Corrective Action Taken:	The audit reinitializes the rIBRGDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIBRGDB tuple.
<i>Bad Data</i> =	Contents of the bad linkage.
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	Logical key (rIBRGDB.brgnum) of the rIBRGDB tuple.
<i>Dump</i> =	rIBRGDB tuple.

BRKPT

Software Release: 5E10 and later

1. BREAKPOINT TABLE (BRKPT) AUDIT

Two utility systems help to verify application programs and to localize and resolve problems in 5ESS® switch software. One system, called the generic access program (GRASP), resides in the administrative module (AM). The other system, called the generic utilities program (UT), resides in the switching modules (SMs), in the communications module processor (CMP), and in the packet switch unit (PSU) protocol handler (PH) model 3. Only the UT data structures are currently audited. The major data structure involved in implementation of the generic utilities program is the breakpoint table. The nodes of the breakpoint table contain linkage data, data specifying a breakpoint (including the memory address and the op code expected at that address), data specifying the utility function to be performed as part of the breakpoint clause, and status flags used by the audit.

The breakpoint table audit (BRKPT) checks the linkage between nodes and verifies the consistency of node data in the breakpoint table. Erroneous data in the breakpoint table could cause faulty operation of UT messages and/or memory mutilation. The audit prevents these effects by removing breakpoints associated with faulty nodes and freeing all associated nodes.

This audit is part of the utilities audits subgroup. It operates in the OSDS-M environment in all modes.

2. DATA STRUCTURES

Figure 1 illustrates the organization of the breakpoint table. It consists of an array (UT_nodes) of 45 nodes, each of which is a C language structure. Figure 2 illustrates how nodes are linked to form a breakpoint list, which is a circular doubly linked list. Nodes may be either active or free. Active nodes are linked in a breakpoint list; free nodes are not linked in a breakpoint list.

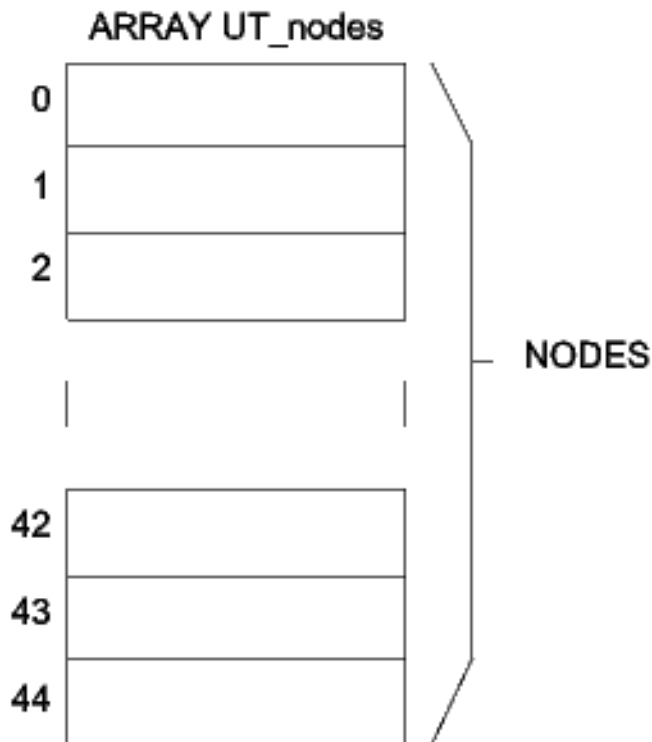
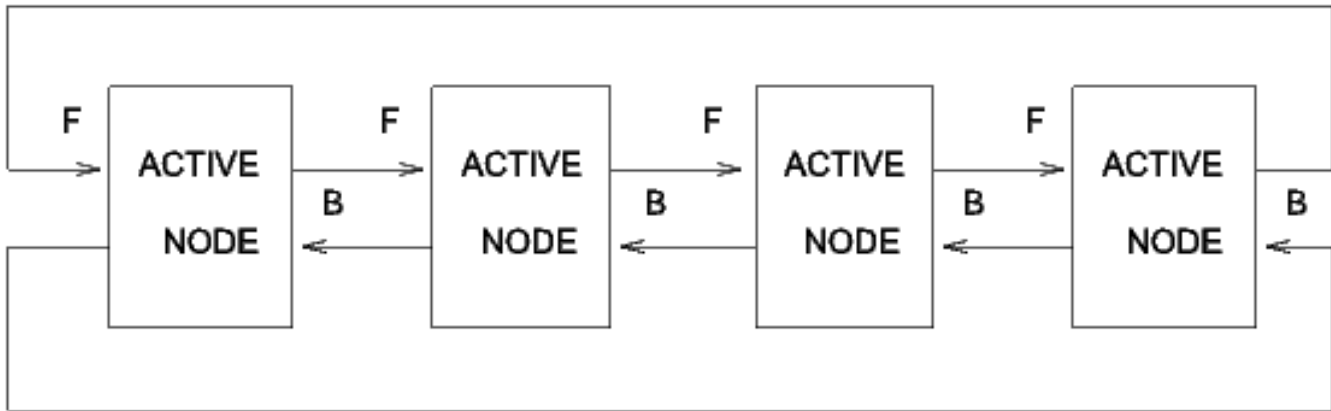


Figure 1 (BRKPT-1) Breakpoint Table Organization



F = FORWARD POINTER

B = BACKWARD POINTER

Figure 2 (BRKPT-2) Active Breakpoint List Structure

3. AUDIT CHECKS

The audit performs the following checks for each active, inhibited, and input function nodes (free nodes are not checked):

- The audit checks the range on the forward and backward pointers of each node on the list. Both pointers must be in the range 0 to 44.
- The audit compares the expected op code, which was entered when the breakpoint was created, to the op code actually found in memory. These two op codes should be identical.
- The audit examines the content of the breakpoint address specified when the breakpoint was created to see if an interrupt op code is present. If an interrupt op code is not present, the breakpoint was not planted.
- The audit performs point-to/point-back checks between nodes.
- The audit checks the range on the forward and backward pointers of each node. The backward pointer of the first node and the forward pointer of the last node must be in the range 0 to 44.

4. ERROR HANDLING

All nodes are marked as untested before the audit begins its routine (this is done either during initialization or when the audit completes a previous cycle). During the audit routine, nodes that pass all tests are marked as tested. After the breakpoint lists have been checked, the audit scans the breakpoint table looking for untested nodes. When it detects an untested node, all nodes in the breakpoint list that contain the untested node are made free. All tested nodes are then marked as untested to prepare for the next invocation of the audit. Each error is reported by means of an output message, which indicates the field found to be erroneous and dumps the contents of the faulty node. Since the nodes in a breakpoint list that contain a faulty node are made free, the breakpoint and its associated functions must be manually reentered.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The when node's type is invalid.
Possible Error Effect:	The utility functions associated with the breakpoint in error cannot be executed.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad function node.
<i>Bad Data</i> =	Backward pointer of the bad function node.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, func_idx.
<i>Dump</i> =	Dump of the entire bad function node.

Table 2 Error Code: BKDToFNC

Error Code:	BKDToFNC
Error Description:	The function node does not point to the next function node.
Possible Error Effect:	The links between function nodes are incorrect.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad function node.
<i>Bad Data</i> =	Backward pointer of the bad function node.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, func_idx.
<i>Dump</i> =	Dump of the entire bad function node.

Table 3 Error Code: BKDTOWH

Error Code:	BKDTOWH
Error Description:	The function node does not point back to the when node.
Possible Error Effect:	The audit cannot get back to the when node.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad function node.
<i>Bad Data</i> =	Backward pointer of the bad function node.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, func_idx.
<i>Dump</i> =	Dump of the entire bad function node.

Table 4 Error Code: FBKDPTR

Error Code:	FBKDPTR
Error Description:	The function node's backward pointer is out-of-range.
Possible Error Effect:	The links between function nodes are incorrect.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad function node.
<i>Bad Data</i> =	Backward pointer of the bad function node.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, func_idx.
<i>Dump</i> =	Dump of the entire bad function node.

Table 5 Error Code: FFWDPTR

Error Code:	FFWDPTR

Error Description:	The function node's forward pointer is out-of-range.
Possible Error Effect:	The links between function nodes are incorrect.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad function node.
<i>Bad Data =</i>	Forward pointer of the bad function node.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the node with the bad pointer, func_idx.
<i>Dump =</i>	Dump of the entire bad function node.

Table 6 Error Code: FWDTOWH

Error Code:	FWDTOWH
Error Description:	The last function node does not point back to the when node.
Possible Error Effect:	The audit cannot get back to the when node.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad function node.
<i>Bad Data =</i>	Forward pointer of the last function node.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the node with the bad pointer, func_idx.
<i>Dump =</i>	Dump of the entire last function node.

Table 7 Error Code: INTERRUPT

Error Code:	INTERRUPT
Error Description:	The address does not contain an interrupt.
Possible Error Effect:	There is a breakpoint somewhere in memory. There is no way to know where the breakpoint is located.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad function node.
<i>Bad Data =</i>	Address where the interrupt should have been.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the node with the bad pointer, whn_idx.
<i>Dump =</i>	Dump of the entire bad function node.

Table 8 Error Code: OPCODE

Error Code:	OPCODE
Error Description:	The stored op code is not equal to the expected op code.
Possible Error Effect:	Office personnel may not be setting the breakpoint where desired. So the breakpoint is not planted.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad function node.
<i>Bad Data =</i>	Bad op code, UTnodes[whn_idx].com.whni.ex_byt.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the node with the bad pointer, whn_idx.
<i>Dump =</i>	Dump of the entire bad function node.

Table 9 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The time control block for UT timed when node has been corrupted.
Possible Error Effect:	The utility functions associated with the breakpoint in error cannot be executed.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.

	removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad function node.
<i>Bad Data</i> =	Backward pointer of the bad function node.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, func_idx.
<i>Dump</i> =	Dump of the entire bad function node.

Table 10 Error Code: WBKDPTR

Error Code:	WBKDPTR
Error Description:	The when node's backward pointer is out-of-range.
Possible Error Effect:	The utility functions associated with the breakpoint in error cannot be executed.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad when node's backward pointer.
<i>Bad Data</i> =	Bad pointer.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, whn_idx.
<i>Dump</i> =	Dump of the entire bad function node.

Table 11 Error Code: WFWDPTR

Error Code:	WFWDPTR
Error Description:	The when node's forward pointer is out-of-range.
Possible Error Effect:	The utility functions associated with the breakpoint in error cannot be executed.
Corrective Action Taken:	The breakpoint and all utility functions associated with that breakpoint clause are removed. The entire breakpoint clause must be reentered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad function node's forward pointer.
<i>Bad Data</i> =	Bad pointer.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the node with the bad pointer, whn_idx.
<i>Dump</i> =	Dump of the entire bad function node.

BUFMSG

Software Release: 5E10 and later

1. BUFFER MESSAGE (BUFMSG) AUDIT

The Operator Services Position System (OSPS) basic automatic call distribution and call handling (BACH) feature includes all call processing software involved in logging onto a basic service terminal (BST) position as an operator. The feature handles requests for operator services and responds to the requests by routing calls to available positions. It also performs keystroke handling. The buffer message relation (RLBUFMSG) is used to queue internal messages [that is, messages between master control center (MCC) subprocesses] to the associated process block relation (RLAPB) in the feature execution control (FEX) environment.

This audit is part of the OSPS audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It operates in the OSDS-M environment in segmented and nonsegmented modes.

2. DATA STRUCTURES

This audit operates on the buffer message relation (RLBUFMSG). The RLBUFMSG relation is owned by the associated process block (RLAPB) relation. Figure 1 illustrates these data structures.

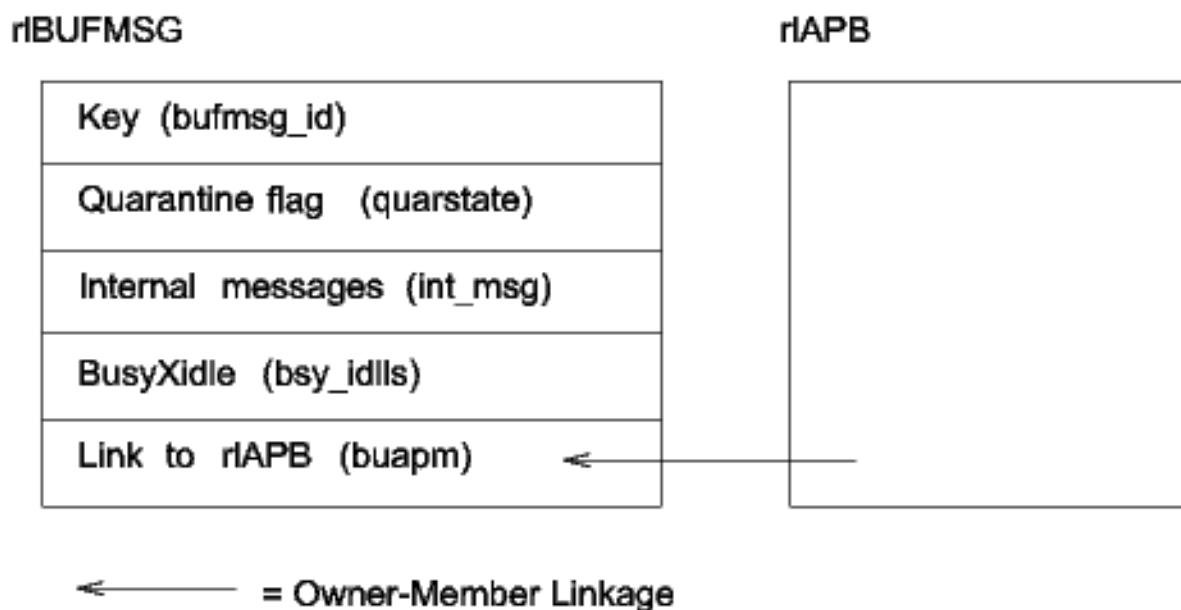


Figure 1 (BUFMSG-1) Data Structures

3. AUDIT CHECKS

To verify the linkages associated with the buffer message, the audit calls the indirect access sequential key driver (DRINASEK) and performs existence checks. After the linkages are verified, the driver calls the semantic check function. The semantic check function performs the following checks.

- (a) Verifies that when the busy/idle state is idle, the tuple is linked to the correct idle list.
- (b) Verifies that when the busy/idle state is busy, the tuple is linked to the busy list.
- (c) The audit reports an error if the tuple is not marked as busy or idle.

4. ERROR HANDLING

The audit reports errors when any linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples and links them to the appropriate idle list.

The audit schedules the associated process block audit (APB) as a related audit.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The bsy_idlIs field is set to an invalid option.
Possible Error Effect:	The rIBUFMSG will be lost to the system.
Corrective Action Taken:	The corresponding rIBUFMSG is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIBUFMSG.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIBUFMSG number.
<i>Dump =</i>	rIBUFMSG.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIBUFMSG is not busy when it should be.
Possible Error Effect:	The rIBUFMSG will be lost to the system.
Corrective Action Taken:	The corresponding rIBUFMSG is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIBUFMSG.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIBUFMSG number.
<i>Dump =</i>	rIBUFMSG.

Table 3 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIBUFMSG is not idle when it should be.
Possible Error Effect:	The rIBUFMSG will be lost to the system.
Corrective Action Taken:	The corresponding rIBUFMSG is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIBUFMSG.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	Expected owner's physical key.
<i>Logical Index =</i>	rIBUFMSG number.
<i>Dump =</i>	rIBUFMSG.

8. CA - CJ

CALINFO

Software Release: 5E14 and later

1. CALEA INFORMATION DATA BLOCK (CALINFO) AUDIT

A communications assistance for law enforcement act (CALEA) information data block is used by feature control to keep track of call data channel (CDC) chaining and CALEA specific information.

2. DATA STRUCTURES

For details on the fields contained in the RLCAL_INFO relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	An rICAL_INFO tuple is on the busy list (DBBUSYCAL_INFO) but the CALEA bridge process (CABR) process identification (rICAL_INFO.cabr_pid) in the rICAL_INFO tuple contains a process that is no longer alive.
Possible Error Effect:	May lead to lost resources and exhaustion of the RLCAL_INFO resource.
Corrective Action Taken:	The audit reinitializes the rICAL_INFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICAL_INFO tuple.
<i>Bad Data =</i>	CABR processor identification (rICAL_INFO.cabr_pid.pcrd).
<i>Good Data =</i>	CABR process number (rICAL_INFO.cabr_pid.procno).
<i>Logical Index =</i>	Logical key (rICAL_INFO.cal_idx) of the rICAL_INFO tuple.
<i>Dump =</i>	rICAL_INFO tuple.

Table 2 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The rICAL_INFO tuple is linked to an invalid rIDB_HEADC tuple.
Possible Error Effect:	May lead to lost resources and exhaustion of the RLCAL_INFO resource.
Corrective Action Taken:	The audit reinitializes the rICAL_INFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICAL_INFO tuple.
<i>Bad Data =</i>	Logical key of the owning rIDB_HEADC tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (rICAL_INFO.cal_idx) of the rICAL_INFO tuple.
<i>Dump =</i>	rICAL_INFO tuple.

Table 3 Error Code: MISSING

Error Code:	MISSING
Error Description:	The case index (rICAL_INFO.caseindex) attribute of the rICAL_INFO tuple contains an invalid index or key (rCASE_IDX.caseindex) into the static

	RLCASE_IDX relation.
Possible Error Effect:	May cause errors in monitoring the active call.
Corrective Action Taken:	The audit reinitializes the rICAL_INFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICAL_INFO tuple.
<i>Bad Data</i> =	Case index (rICAL_INFO.caseindex).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (rICAL_INFO.cal_indx) of the rICAL_INFO tuple.
<i>Dump</i> =	rICAL_INFO tuple.

Table 4 Error Code: STUCK

Error Code:	STUCK
Error Description:	Either the rICAL_INFO tuple is not linked (rICAL_INFO.crcallink.st_ownk = 0) to a rCRECORD tuple or the call move (rICAL_INFO.call_move) attribute in the rICAL_INFO tuple is set to DBYES for longer than 5 minutes.
Possible Error Effect:	May lead to lost resources and exhaustion of the RLCAL_INFO resource.
Corrective Action Taken:	The audit reinitializes the rICAL_INFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICAL_INFO tuple.
<i>Bad Data</i> =	Linkage (rICAL_INFO.crcallink.st_ownk) to the rCRECORD tuple.
<i>Good Data</i> =	Content of call move attribute (rICAL_INFO.call_move).
<i>Logical Index</i> =	Logical key (rICAL_INFO.cal_indx) of the rICAL_INFO tuple.
<i>Dump</i> =	rICAL_INFO tuple.

Table 5 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rICAL_INFO tuple is on the idle list, but one of the non-idle list linkages is not null (DALNKNNULL).
Possible Error Effect:	May lead to lost system resources.
Corrective Action Taken:	The audit reinitializes the rICAL_INFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICAL_INFO tuple.
<i>Bad Data</i> =	Non-null data found in the rICAL_INFO tuple.
<i>Good Data</i> =	DALNKNNULL.
<i>Logical Index</i> =	Logical key (rICAL_INFO.cal_indx) of the rICAL_INFO tuple.
<i>Dump</i> =	rICAL_INFO tuple.

CBDB

Software Release: 5E10 and later

1. CALL BUILDUP DATA BLOCK (CBDB) AUDIT

The RLCBDB is a dynamic relation which is used to alleviate the stack overflow problem. The call buildup data block (CBDB) tuple is linked to the head cell of the idle pools while it is not in use. It is allocated and linked to the associated process control block link area (PCBLA) tuple during the call processing time. The CBDB audit uses an indexed access sequential key driver to step through the tuples and to invoke key, quarantine bit and linkage checks. In addition, semantic checks are performed to insure that:

- The corresponding process has not given up the stack while there is a linkage between CBDB and PCBLA tuple.
- The idle CBDB tuple is accurately linked to the idle list and does not link to any other data structure.

2. DATA STRUCTURES

The RLCBDB relation is the call buildup data block. The CBDB audit is part of the feature control audits subgroup. It runs in the segmented modes and operates in the OSDS-M environment.

3. AUDIT CHECKS

The relation rICBDB is used to reduce the stack overflow problem. All the business and residential customer services (BRCS) calls (both integrated services digital network [ISDN] and non-ISDN) use this relation. The idle CBDB tuples are linked to the head cell of rIDBHEADC. While the CBDB tuple is allocated during calling processing time, it will be linked to the associated PCBLA. The CBDB audit will verify the data in the rICBDB relation. Because the rICBDB relation is a direct access relation, the indexed access sequential key driver is used to step through the tuples. The following semantic checks will be performed by the CBDB audit:

- If the CBDB tuple is linked to idle list, the other linkage fields should contain NULL.
- Verify that if the CBDB tuple is busy, the associated process should not release the stack.

4. ERROR HANDLING

When the audit encounters an error, an error message is printed on the ROP. The process associated with this error tuple is purged. The error tuple will be restored back to idle list. Audit control will schedule the related audit (PCBLA & DBHEADC) to run in the elevated mode after the error tuples have been recovered.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DEAD_PID

Error Code:	DEAD_PID
Error Description:	CBDB is linked to a PCBLA, but the owning process does not exist.
Possible Error Effect:	CBDB is tied to an inactive process and cannot be used for other calls.
Corrective Action Taken:	CBDB tuple will be put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CBDB number.
<i>Bad Data =</i>	Process number.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCBDB.
<i>Dump</i> =	NA.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An rCBDB is supposed to be idle but is not linked to the rCBDB idle list.
Possible Error Effect:	The rCBDB is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rCBDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCBDB.
<i>Bad Data</i> =	Owner link of rCBDB.
<i>Good Data</i> =	Physical key of idle head call.
<i>Logical Index</i> =	Logical key of rCBDB.
<i>Dump</i> =	NA.

Table 3 Error Code: PROC_TO

Error Code:	PROC_TO
Error Description:	The CBDB tuple is only used by terminal process, but it erroneously links to a system process.
Possible Error Effect:	This error could result in running out of CBDB tuples for terminal processes.
Corrective Action Taken:	The audit idles the rCBDB tuple. The process owning this CBDB tuple will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCBDB.
<i>Bad Data</i> =	Process number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCBDB.
<i>Dump</i> =	NA.

Table 4 Error Code: STKTEMP

Error Code:	STKTEMP
Error Description:	The process owning this CBDB tuple has released the stack but the CBDB tuple is still tied to this process.
Possible Error Effect:	This CBDB tuple will not be used by other calls. This error could result in running out of CBDB tuples for other calls.
Corrective Action Taken:	The audit idles the rCBDB tuple. The process owning this CBDB tuple will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCBDB.
<i>Bad Data</i> =	Process number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCBDB.
<i>Dump</i> =	NA.

Table 5 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An rCBDB is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to an rCBDB on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rCBDB tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCBDB.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rCBDB.

<i>Dump</i> =	NA.

CBHC-A

Software Release: 5E10 only

1. CONCENTRATED BUFFER HEAD CELL (CBHC) AUDIT

Originations and terminations to a switch may be made by lines and trunks. Lines are concentrated, which means they share hardware. Trunks are unconcentrated; they use as much hardware as they need. In both cases, there may be more inputs to the switch during a heavy call processing load than can be processed. Excess originations and terminations are buffered onto head cells. Each buffer head cell represents a one-second segment of time during which an origination or termination may have occurred; there are 20 one-second segments. Each concentrated buffer head cell owns one or more group scan data blocks. A group scan data block represents a group of eight lines. If a line origination must be buffered, the group scan data block is linked to the concentrated buffer head cell that represents the time segment in which the origination occurred. The global variable PCcbtotal contains the number of group scan data blocks that are linked to the concentrated buffer head cell.

The concentrated buffer head cell audit (CBHC) prevents the loss of queued originations and linked data blocks. This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

This audit operates on the concentrated buffer head cell relation (RLCBHC). The RLCBHC relation has an owner linkage with the group scan data block relation (RLGSDB). Figure 1 illustrates this linkage.

3. AUDIT CHECKS

To verify the linkages associated with the CBHC's, the audit calls the linkage driver (DRLINK) once for each audit segment. The audit checks each head cell in order using the physical key to step through the relation. After the linkages are verified, and the driver returns completed a check on the number of rIGSDBs linked to the rLCBHC is performed. The number should be consistent with the number in the global variable PCcbtotal.

4. ERROR HANDLING

Any rLCBHC tuples that fail these checks are reinitialized. The logical key field is set to its correct value.

If the number of rIGSDB tuples linked to an rLCBHC is inconsistent with the global variable PCcbtotal, the audit prints an error message and corrects PCcbtotal.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of scanning blocks linked to all rLCBHC busy lists is inconsistent with the contents of PCcbtotal.
Possible Error Effect:	May cause the system to go into overload.
Corrective Action Taken:	Set PCcbtotal to the number of scanning blocks linked to all rLCBHC busy lists.
Manual Action Required:	None.
Dump Description:	
Error Address =	Address of the PCcbtotal global variable.
Bad Data =	The contents of PCcbtotal.
Good Data =	The number of scanning blocks linked to all rLCBHC busy lists.
Logical Index =	NA.
Dump =	NA.

CBHC-B

Software Release: 5E11 and later

1. CONCENTRATED BUFFER HEAD CELL (CBHC) AUDIT

Originations and terminations to a switch may be made by lines and trunks. Lines are concentrated, which means they share hardware. Trunks are unconcentrated; they use as much hardware as they need. In both cases, there may be more inputs to the switch during a heavy call processing load than can be processed. Excess originations and terminations are buffered onto head cells. Each buffer head cell represents a one-second segment of time during which an origination or termination may have occurred; there are 20 one-second segments. Each concentrated buffer head cell owns one or more group scan data blocks. A group scan data block represents a group of eight lines. If a line origination must be buffered, the group scan data block is linked to the concentrated buffer head cell that represents the time segment in which the origination occurred. The global variable PCcbtotal contains the number of group scan data blocks that are linked to the concentrated buffer head cell.

The concentrated buffer head cell audit (CBHC) prevents the loss of queued originations and linked data blocks. This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

This audit operates on the concentrated buffer head cell relation (RLCBHC). The RLCBHC relation has an owner linkage with the group scan data block relation (RLGSDB). Figure 1 illustrates this linkage.

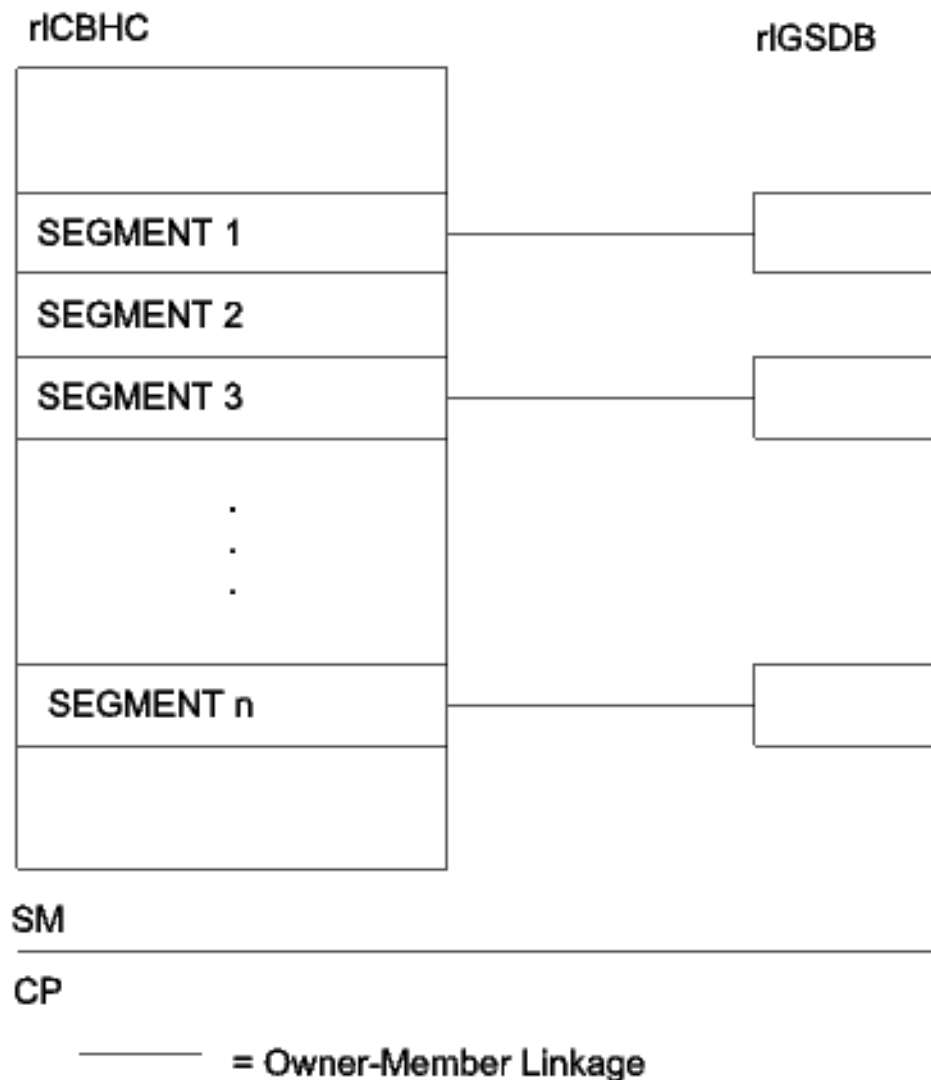


Figure 1 (CBHC-1) rICBHC Linkages

3. AUDIT CHECKS

To verify the linkages associated with the CBHC's, the audit calls the linkage driver (DRLINK) once for each audit segment. The audit checks each head cell in order using the physical key to step through the relation. After the linkages are verified, and the driver returns completed a check on the number of rIGSDBs linked to the rICBHC is performed. The number should be consistent with the number in the global variable PCcbtotal.

4. ERROR HANDLING

Any rICBHC tuples that fail these checks are reinitialized. The logical key field is set to its correct value.

If the number of rIGSDB tuples linked to an rICBHC is inconsistent with the global variable PCcbtotal, the audit prints an error message and corrects PCcbtotal.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of scanning blocks linked to all rICBHC busy lists is inconsistent with the contents of PCcbtotal.
Possible Error Effect:	May cause the system to go into overload.
Corrective Action Taken:	Set PCcbtotal to the number of scanning blocks linked to all rICBHC busy lists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCcbtotal global variable.
<i>Bad Data =</i>	The contents of PCcbtotal.
<i>Good Data =</i>	The number of scanning blocks linked to all rICBHC busy lists.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

CCATBL

Software Release: 5E10 and later

1. CONFERENCE CIRCUIT ACCESS TABLE (CCATBL) AUDIT

Conference circuits are used for conference calling and for verifying that busy lines are actually busy. The conference circuits are part of the global digital service unit (GDSU) and are resources that can be accessed from any switching module (SM) in an office. Each office has a specified number of conference circuits; therefore, it is important that these resources be operational. The conference circuit access table audit (CCATBL) verifies data in the dynamic conference circuit access table relation (RLCCATBL) against data in the static global conference circuit relation (RLGCKTI). The audit identifies semantic inconsistencies, extra tuples, and bad keys in the RLCCATBL relation.

This audit is part of the peripheral control audits subgroup. It runs in the communication module processor (CMP) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The CCATBL audit verifies the dynamic conference circuit access table relation (RLCCATBL) against the static global conference circuit relation (RLGCKTI). The RLCCATBL relation contains information about all conference circuits, such as the switching module number (im), the GDSU service group key (sg), the service group number (sgn), and the digital service unit number (dsu). The additional types of conference circuits have necessitated the addition of several fields in the rLGCKTI and rLCCATBL data structures. The index in rLGCKTI points to either the three-port or the six-port array. The total number of circuits listed in rLCCATBL lists three-port and six-port boards separately. Figure 1 shows these changes.

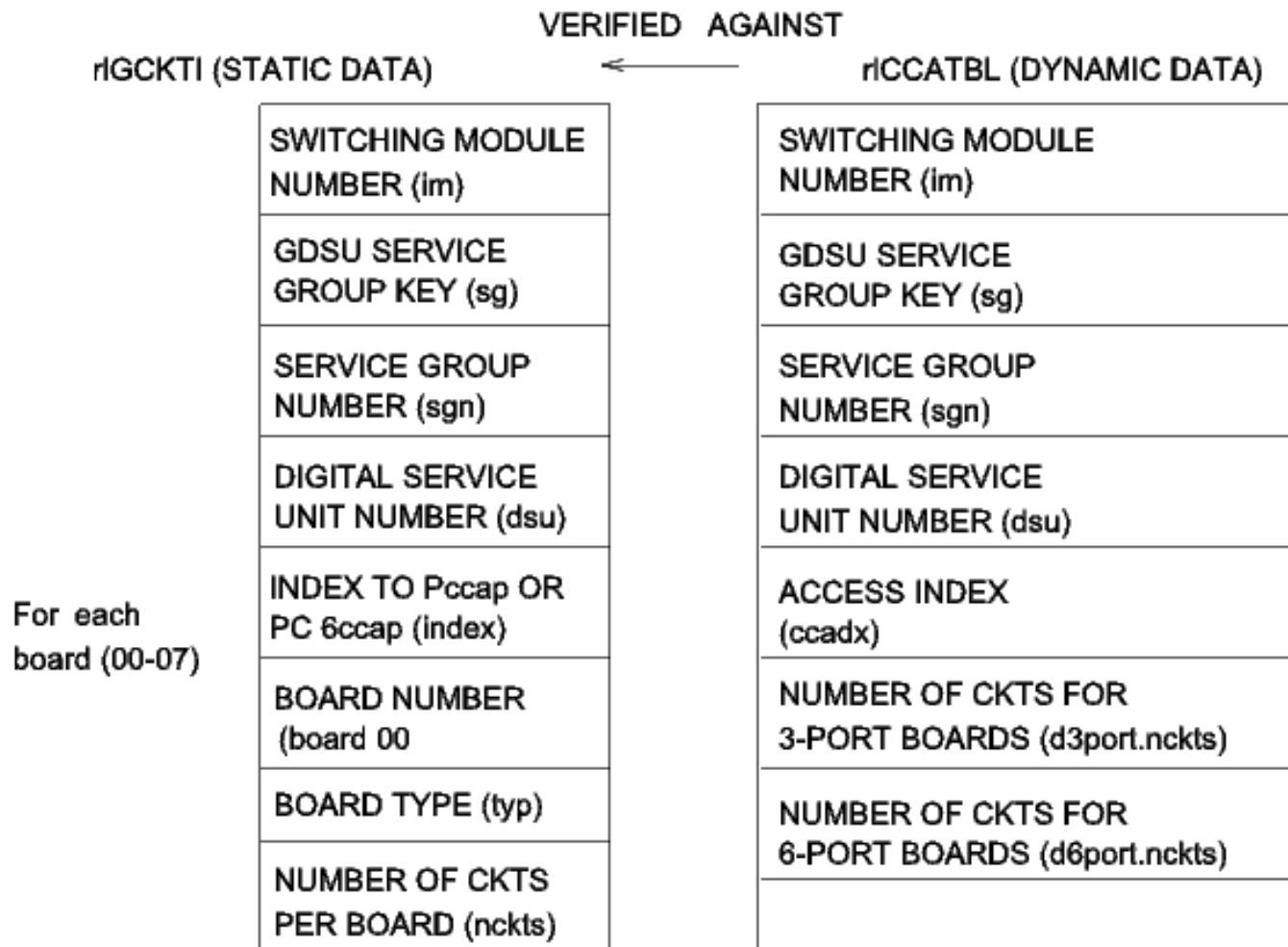


Figure 1 (CCATBL-1) CCATBL Audit

3. AUDIT CHECKS

The conference circuit access table audit uses the dynamic driver (DRDYN) to step through the RLCCATBL relation. The DRDYN driver calls a semantic check function to perform the following dynamic-to-static existence checks:

CCINDEX = Check the key of the rCCATBL tuple.

CCNON = Check the existence of a corresponding rIGCKTI tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CCINDEX

Error Code:	CCINDEX
Error Description:	The key field (ccadx) of the rCCATBL tuple does not match the corresponding

	key field (index) in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls.
Corrective Action Taken:	The rICCATBL tuple will be zeroed out, and the GCKTI audit will be scheduled to rebuild this rICCATBL tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	The key field (ccadx) from the rICCATBL tuple.
<i>Good Data =</i>	The key field (index) from the rIGCKTI tuple.
<i>Logical Index =</i>	Logical key of the rICCATBL tuple found in error.
<i>Dump =</i>	NA.

Table 2 Error Code: CCNON

Error Code:	CCNON
Error Description:	The current rICCATBL tuple under audit does not have a corresponding rIGCKTI tuple.
Possible Error Effect:	Conference calls might be routed to non-existing circuits.
Corrective Action Taken:	Remove the rICCATBL tuple from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	Logical key (ccadx) of the rICCATBL tuple found in error.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rICCATBL tuple found in error.
<i>Dump =</i>	NA.

CCBCOM-A

Software Release: 5E10 - 5E11

1. CHANNEL CONTROL BLOCK COMMON AREA (CCBCOM) AUDIT

The channel control block is an internal view of a customer's port. It consists of two regions:

- The channel control block common area, which is the background region. This region is an abstraction of the path from the customer terminal to the time slot interchanger (TSI). Figure 1 illustrates this path for a B-link (data) line. The path for an A-link (control) line goes through the control interface (CI) instead of the data interface (DI).
- The channel control block communication region, which is the foreground region. Background terminal processes communicate with foreground processes through this region.

Background and foreground processing are explained in more detail in the job list head cell (JBLH) audit description. The channel control block common area audit (CCBCOM) checks for incorrect data in the channel control block common area. Data errors can create incorrect call processing inputs, or can put the channel control block common area in a limbo condition, which results in lost resources for call processing.

This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

This audit operates on the linkage areas of the channel control block common area relation (RLCCBCOM). The RLCCBCOM relation has a one-to-one linkage with the channel data block relation (RLCHDB) and a special linkage with the high-level service circuit relation (RLHLSC). Its owner linkages are with the following members:

- The path data block relation (RLPHDB).
- The circuit queuing block relation (RLCQB).
- The circuit data block common area relation (RLCDBCOM).
- RLCHDB.

Its member linkages are with the following owners:

- The process control block linkage area relation (RLPCBLA).
- The job list head cell relation (RLJBLH).

Figure 2 illustrates the layout of a channel control block common area and Figure 3 illustrates the RLCCBCOM linkages.

3. AUDIT CHECKS

To verify the linkages associated with the channel control block common areas, the CCBCOM audit calls the linkage driver (DRLINK) once for each audit segment. DRLINK checks the one-to-one linkages with RLCHDB and the owner-member linkages with RLCDBCOM, RLCHDB, RLCQB, RLJBLH, RLPCBLA, and RLPHDB. After verifying the linkages, the DRLINK driver invokes a semantic check function that performs the following checks:

- (a) The audit checks the logical port state (lpstate) field to ensure that it is a valid state variable.
- (b) The audit follows the pointer from the channel control block foreground region to the high-level service circuit for a ringing call. The ccbusr field in the rHLSC tuple should contain the number of the channel control block that is in the ringing state. The audit does not make this point-to/point-back check for SLC[®] 96 ringing jobs because the SLC[®] 96 does not use high-level service circuits.
- (c) The audit checks the lpstate field against the channel control block common area linkages. A channel control block common area in the idle state should be linked to the idle lists so it can be used by call processing. All its linkages that are not involved in the idle list mapping must be zeroed. The linkages for a channel control block common area in a stable state should be error-free; if they are not, a bad linkage may cause an assert before the audit finds the error.
- (d) The rCCBCOM tuple should not be in the transient state for more than five minutes or a call in progress will not be able to disconnect. The audit times the lpstate field for a change from the transient state.

4. ERROR HANDLING

Any tuples that fail these checks are initialized and placed on the idle list.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	A rCCBCOM in B-channel state has different port name from that indicated by rPORTLA.
Possible Error Effect:	The bad port name may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Port name in rCCBCOM.
<i>Good Data =</i>	Key of rPORTLA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The rISPABUF tuple is linked to the list which is reserved for routine port conditioning (RPC) jobs. However the tuple is not being used for a RPC job.
Possible Error Effect:	ISPA could be preventing RPC jobs from running.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rISPABUF tuple.
<i>Bad Data =</i>	The current ISPA job type.
<i>Good Data =</i>	The ISPA job type for RPC jobs.
<i>Logical Index =</i>	Key of rISPABUF tuple.
<i>Dump =</i>	rISPABUF tuple.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	A rCCBCOM in ISLU D-channel cut through or B-channel state has an error in

	one of its linkages.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Linkage ID in the rCCBCOM.
	0 = rPCBLA owner linkage.
	1 = rCCBCOM head linkage.
	2 = rCCBCOM tail linkage.
	3 = rCHDB head linkage.
	4 = rCHDB tail linkage.
	5 = rCHDB one to one linkage.
	6 = rCDBCOM linkage not null.
	7 = rPTSB linkage not null.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

Table 4 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	The rgbus circuit name that was calculated from the port in the present rCCBCOM tuple does not correspond to the rIRGBUS tuple that owns the corresponding rCCBUSR tuple for this ISLU-Z ringing.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The rCCBCOM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM tuple.
<i>Bad Data =</i>	rgbus.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key to tuple.
<i>Dump 1 =</i>	rCCBCOM tuple.
<i>Dump 2 =</i>	rIRGBUS tuple.

Table 5 Error Code: CCB_HLSC

Error Code:	CCB_HLSC
Error Description:	The high-level service circuit (hlsc) for a ringing call does not point to this rCCBCOM.
Possible Error Effect:	A ringing call may be stuck forever.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rCCBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	rCCBCOM logical key.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump 1 =</i>	rCCBCOM tuple.
<i>Dump 2 =</i>	rHLSC tuple.

Table 6 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rICBCOM is in the idle state but is not linked to the rICBCOM idle list.
Possible Error Effect:	The rICBCOM is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rICBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICBCOM.
<i>Bad Data =</i>	Owner link of rICBCOM.
<i>Good Data =</i>	Physical key of idle head cell.
<i>Logical Index =</i>	Logical key of rICBCOM.
<i>Dump =</i>	rICBCOM tuple.

Table 7 Error Code: KEYFUNC

Error Code:	KEYFUNC
Error Description:	The tdbmap in the foreground region shows that a tdb is involved in ISLU-Z ringing, but the corresponding tuple of the PCtdbtbl for that tdb shows that it is not.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The rICBCOM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICBCOM tuple.
<i>Bad Data =</i>	CCBCOM information.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of tuple.
<i>Dump =</i>	rICBCOM tuple.

Table 8 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	The process owning rICBCOM tuple in B-channel state can not get the B-channel rIORTLA.
Possible Error Effect:	The error may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICBCOM tuple. This action terminates the call that uses the rICBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA (current process).
<i>Bad Data =</i>	Logical key or rIPCBLA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rICBCOM.
<i>Dump =</i>	rICBCOM tuple.

Table 9 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The lu_num or tdbno fields associated with the PC foreground region of the rICBCOM has out of range values. This rICBCOM is associated with the peripheral task data blocks with the lpstate equal to PCPTASK.
Possible Error Effect:	The error may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICBCOM tuple. This action terminates the PTDB that uses the rICBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The lu_num value of the PC foreground region, ccbfgr_ptr->fgwrkb.JBistask.lu_num
<i>Bad Data =</i>	The tdbno value of the PC foreground region, ccbfgr_ptr->fgwrkb.JBistask.tdb.tdbhead.tdbno
<i>Good Data =</i>	PCTOTPUNTS, total number of PCT link based peripherals.
<i>Logical Index =</i>	Logical key of rICBCOM.
<i>Dump =</i>	The PC foreground region, ccbfgr_ptr.

Table 10 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A rICBCOM in B-channel state has an error linkage to rIPCBLA (owner tuple).
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICBCOM tuple. This action terminates the call that uses the rICBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICBCOM.
<i>Bad Data =</i>	Logical key of rICBCOM.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rICBCOM.
<i>Dump =</i>	rICBCOM tuple.

Table 11 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	A rICBCOM in a stable state has an error in one of its linkages.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICBCOM tuple. This action terminates the call that uses the rICBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICBCOM.
<i>Bad Data =</i>	Linkage ID in the rICBCOM. 0 = rIPCBLA owner linkage. 1 = rICBCOM head linkage. 2 = rICBCOM tail linkage. 3 = rICHDB head linkage. 4 = rICHDB tail linkage. 5 = rICHDB one to one linkage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rICBCOM.
<i>Dump =</i>	rICBCOM tuple.

Table 12 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rICBCOM has an invalid state variable.
Possible Error Effect:	The state of the rICBCOM is unknown.
Corrective Action Taken:	The audit idles the rICBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rICBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICBCOM.
<i>Bad Data =</i>	State variable.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rICBCOM.
<i>Dump =</i>	rICBCOM tuple.

Table 13 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than 5 minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The tuple is idled and the call terminated.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 14 Error Code: TASK

Error Code:	TASK
Error Description:	A rICCB COM tuple in the peripheral control task state (rICCB COM.lstate = PCTASK) failed the point-to-point-back check between the corresponding peripheral control channel control block foreground region (PCcbfg[rICCB COM.ccb]) and the peripheral control task data block (PCtdbtbl[] array). Depending upon the loadable package, this error can also come out for the peripheral control timeslot (PCT) link based tasks. Here, an rICCB COM tuple with lstate equal to PCPTASK, failed the point to point back checks between corresponding peripheral control channel control block foreground region and the peripheral task data block (PTDB) associated with it.
Possible Error Effect:	The rICCB COM tuple will be lost to the system causing resource failures which may result in lost calls.
Corrective Action Taken:	The rICCB COM tuple will be reinitialized and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad rICCB COM tuple.
<i>Bad Data</i> =	NA. Or for PTDBs, the ccbnum field of the PCptdbtbl[][] accessed by using the logical unit and PTDB number from the PC foreground region data.
<i>Good Data</i> =	NA. Or for PTDBs, the ccbnum field of the PCptdbtbl[][] accessed by using the logical unit and PTDB number from the PC foreground region data.
<i>Logical Index</i> =	Key of the bad rICCB COM tuple.
<i>Dump 1</i> =	PC foreground region (of type struct PCCBPG).
<i>Dump 2</i> =	The PCtdbtbl[] array (not for PTDBs).

Table 15 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	Only one ISPA RPC job is allowed to be scheduled at a time. Since this rISPABUF tuple is not the first one on the list of RPC jobs, it is invalid.
Possible Error Effect:	Someone other than the RPC scheduler may be requesting RPC jobs.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad rISPABUF tuple.
<i>Bad Data</i> =	Logical key of the previous rISPABUF tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of the bad rISPABUF tuple.
<i>Dump</i> =	rISPABUF tuple.

Table 16 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rICCB COM is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rICCB COM on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rICCB COM tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

CCBCOM-B

Software Release: 5E12 only

1. CHANNEL CONTROL BLOCK COMMON AREA (CCBCOM) AUDIT

The channel control block is an internal view of a customer's port. It consists of two regions:

- The channel control block common area, which is the background region. This region is an abstraction of the path from the customer terminal to the time slot interchanger (TSI). Figure 1 illustrates this path for a B-link (data) line. The path for an A-link (control) line goes through the control interface (CI) instead of the data interface (DI).
- The channel control block communication region, which is the foreground region. Background terminal processes communicate with foreground processes through this region. Background and foreground processing are explained in more detail in the job list head cell (JBLH) audit description.

The channel control block common area audit (CCBCOM) checks for incorrect data in the channel control block common area. Data errors can create incorrect call processing inputs, or can put the channel control block common area in a limbo condition, which results in lost resources for call processing.

This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

This audit operates on the linkage areas of the channel control block common area relation (RCCBCOM). The RLCCBCOM relation has a one-to-one linkage with the channel data block relation (RLCHDB) and a special linkage with the high-level service circuit relation (RLHLSC). Its owner linkages are with the following members:

- The path data block relation (RLPHDB).
- The circuit queuing block relation (RLCQB).
- The circuit data block common area relation (RLCDBCOM).
- RLCHDB.

Its member linkages are with the following owners:

- The process control block linkage area relation (RLPCBLA).
- The job list head cell relation (RLJBLH).

Figure 2 illustrates the layout of a channel control block common area and Figure 3 illustrates the RLCCBCOM linkages.

3. AUDIT CHECKS

To verify the linkages associated with the channel control block common areas, the CCBCOM audit calls the linkage driver (DRLINK) once for each audit segment. DRLINK checks the one-to-one linkages with RLCHDB and the owner-member linkages with RLCDBCOM, RLCHDB, RLCQB, RLJBLH, RLPCBLA, and RLPHDB. After verifying the linkages, the DRLINK driver invokes a semantic check function that performs the following checks:

- (a) The audit checks the logical port state (lpstate) field to ensure that it is a valid state variable.
- (b) The audit follows the pointer from the channel control block foreground region to the high-level service circuit for a ringing call. The ccbusr field in the rHLSC tuple should contain the number of the channel control block that is in the ringing state. The audit does not make this point-to/point-back check for SLC[®] 96 ringing jobs because the SLC[®] 96 does not use high-level service circuits.
- (c) The audit checks the lpstate field against the channel control block common area linkages. A channel control block common area in the idle state should be linked to the idle lists so it can be used by call processing. All its linkages that are not involved in the idle list mapping must be zeroed. The linkages for a channel control block common area in a stable state should be error-free; if they are not, a bad linkage may cause an assert before the audit finds the error.
- (d) The rCCBCOM tuple should not be in the transient state for more than five minutes or a call in progress will not be able to disconnect. The audit times the lpstate field for a change from the transient state.

4. ERROR HANDLING

Any tuples that fail these checks are initialized and placed on the idle list.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	A rCCBCOM in B-channel state has different port name from that indicated by rPORTLA.
Possible Error Effect:	The bad port name may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Port name in rCCBCOM.
<i>Good Data =</i>	Key of rPORTLA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The rISPABUF tuple is linked to the list which is reserved for routine port conditioning (RPC) jobs. However the tuple is not being used for a RPC job.
Possible Error Effect:	ISPA could be preventing RPC jobs from running.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rISPABUF tuple.
<i>Bad Data =</i>	The current ISPA job type.
<i>Good Data =</i>	The ISPA job type for RPC jobs.
<i>Logical Index =</i>	Key of rISPABUF tuple.
<i>Dump =</i>	rISPABUF tuple.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	A rCCBCOM in ISLU D-channel cut through or B-channel state has an error in

	one of its linkages.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Linkage ID in the rCCBCOM.
	0 = rPCBLA owner linkage.
	1 = rPHDB head linkage.
	2 = rPHDB tail linkage.
	3 = rCHDB head linkage.
	4 = rCHDB tail linkage.
	5 = rCHDB one to one linkage.
	6 = rCDBCOM linkage.
	7 = rPTSB linkage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

Table 4 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	The rgbus circuit name that was calculated from the port in the present rCCBCOM tuple does not correspond to the rIRGBUS tuple that owns the corresponding rCCBUSR tuple for this ISLU-Z ringing.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The rCCBCOM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM tuple.
<i>Bad Data =</i>	rgbus.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key to tuple.
<i>Dump 1 =</i>	rCCBCOM tuple.
<i>Dump 2 =</i>	rIRGBUS tuple.

Table 5 Error Code: CCB_HLSC

Error Code:	CCB_HLSC
Error Description:	The high-level service circuit (HLSC) for a ringing call does not point to this rCCBCOM.
Possible Error Effect:	A ringing call may be stuck forever.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rCCBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	rCCBCOM logical key.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump 1 =</i>	rCCBCOM tuple.
<i>Dump 2 =</i>	rHLSC tuple.

Table 6 Error Code: HLSC

Error Code:	HLSC
Error Description:	The HLSC for a ringing call does not point to this rCCBCOM.
Possible Error Effect:	A ringing call may be stuck forever.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rCCBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Address of rHLSC.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump 1 =</i>	rCCBCOM tuple.
<i>Dump 2 =</i>	rHLSC tuple.

Table 7 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rCCBCOM is in the idle state but is not linked to the rCCBCOM idle list.
Possible Error Effect:	The rCCBCOM is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rCCBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Owner link of rCCBCOM.
<i>Good Data =</i>	Physical key of idle head cell.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

Table 8 Error Code: KEYFUNC

Error Code:	KEYFUNC
Error Description:	The tdbmap in the foreground region shows that a tdb is involved in ISLU-Z ringing, but the corresponding tuple of the PCtdbtbl for that tdb shows that it is not.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The rCCBCOM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM tuple.
<i>Bad Data =</i>	CCBCOM information.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of tuple.
<i>Dump =</i>	rCCBCOM tuple.

Table 9 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	There does not appear to be linkage from the process (rPCBLA) to the port (rIORTLA).
Possible Error Effect:	The port is lost to this process, therefore the call is dead.
Corrective Action Taken:	The rCCBCOM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPCBLA tuple.
<i>Bad Data =</i>	Key (process number) of the rPCBLA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rCCBCOM tuple.
<i>Dump =</i>	rCCBCOM tuple.

Table 10 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	The process owning rCCBCOM tuple in B-channel state can not get the B-channel rIORTLA.
Possible Error Effect:	The error may cause an assert before the audit finds the error.

Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA (current process).
<i>Bad Data</i> =	Logical key or rIPCBLA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

Table 11 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The lu_num or tdbno fields associated with the PC foreground region of the rCCBCOM has out of range values. This rCCBCOM is associated with the peripheral task data blocks with the lpstate equal to PCPTASK.
Possible Error Effect:	The error may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the PTDB that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The lu_num value of the PC foreground region, ccbfgr_ptr->fgwrkb.JBistask.lu_num
<i>Bad Data</i> =	The tdbno value of the PC foreground region, ccbfgr_ptr->fgwrkb.JBistask.tdb.tdbhead.tdbno
<i>Good Data</i> =	PCTOTPUNTS, total number of PCT link based peripherals.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	The PC foreground region, ccbfgr_ptr.

Table 12 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A rCCBCOM in B-channel state has an error linkage to rIPCBLA (owner tuple).
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	Logical key of rCCBCOM.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

Table 13 Error Code: STABLE_LINK

Error Code:	STABLE LINK
Error Description:	A rCCBCOM in a stable state has an error in one of its linkages.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	Linkage ID in the rCCBCOM. 0 = rIPCBLA owner linkage. 1 = rIPHDB head linkage. 2 = rIPHDB tail linkage. 3 = rICHDB head linkage.

	4 = rCHDB tail linkage.
	5 = rCHDB one to one linkage.
	6 = rCDBCOM head linkage.
	7 = rIPTSB linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

Table 14 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rCCBCOM has an invalid state variable.
Possible Error Effect:	The state of the rCCBCOM is unknown.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rCCBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	State variable (rCCBCOM.lstate).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

Table 15 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than 5 minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The tuple is idled and the call terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

Table 16 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The tuple has been in the ringing state for more than the time allowed by the global parameter Glringtot.
Possible Error Effect:	A call in progress has been in the ringing state too long.
Corrective Action Taken:	An MGINTERRUPT message is sent to the owning process to clean-up silently.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	length of time it's been ringing.
<i>Good Data</i> =	the value it is allowed to ring (Glringtot).
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump 1</i> =	rCCBCOM tuple.
<i>Dump 2</i> =	rPCBLA tuple.

Table 17 Error Code: TASK

Error Code:	TASK
Error Description:	A rCCBCOM tuple in the peripheral control task state (rCCBCOM.lstate = PCTASK) failed the point-to-point-back check between the corresponding peripheral control channel control block foreground region (PCccbfgr[rCCBCOM.ccb]) and the peripheral control task data block

	(PCtdbtbl[] array).
	Depending upon the loadable package, this error can also come out for the peripheral control timeslot (PCT) link based tasks. Here, an rCCBCOM tuple with lstate equal to PCPTASK, failed the point to point back checks between corresponding peripheral control channel control block foreground region and the peripheral task data block (PTDB) associated with it.
Possible Error Effect:	The rCCBCOM tuple will be lost to the system causing resource failures which may result in lost calls.
Corrective Action Taken:	The rCCBCOM tuple will be reinitialized and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rCCBCOM tuple.
<i>Bad Data =</i>	NA. Or for PTDBs, the ccbnum field of the PCptdbtbl[][] accessed by using the logical unit and PTDB number from the PC foreground region data.
<i>Good Data =</i>	NA. Or for PTDBs, the ccbnum field of the PCptdbtbl[][] accessed by using the logical unit and PTDB number from the PC foreground region data.
<i>Logical Index =</i>	Key of the bad rCCBCOM tuple.
<i>Dump 1 =</i>	PC foreground region (of type struct PCCBPG).
<i>Dump 2 =</i>	The PCtdbtbl[] array. (Not for PTDBs)

Table 18 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The high level service circuit using this CCB must point to it. However, the unit number is null.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rCCBCOM tuple will be recovered and returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Line unit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCCBCOM tuple.
<i>Dump =</i>	NA.

Table 19 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	Only one ISPA RPC job is allowed to be scheduled at a time. Since this rISPABUF tuple is not the first one on the list of RPC jobs, it is invalid.
Possible Error Effect:	Someone other than the RPC scheduler may be requesting RPC jobs.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rISPABUF tuple.
<i>Bad Data =</i>	Logical key of the previous rISPABUF tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Key of the bad rISPABUF tuple.
<i>Dump =</i>	rISPABUF tuple.

Table 20 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rCCBCOM is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rCCBCOM on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Nonzero value of linkage.
<i>Good Data =</i>	0.

<i>Logical Index</i> =	Logical key of rICBCOM.
<i>Dump</i> =	rICBCOM tuple.

CCBCOM-C

Software Release: 5E13 and later

1. CHANNEL CONTROL BLOCK COMMON AREA (CCBCOM) AUDIT

The channel control block is an internal view of a customer's port. It consists of two regions:

- The channel control block common area, which is the background region. This region is an abstraction of the path from the customer terminal to the time slot interchanger (TSI). Figure 1 illustrates this path for a B-link (data) line. The path for an A-link (control) line goes through the control interface (CI) instead of the data interface (DI).
- The channel control block communication region, which is the foreground region. Background terminal processes communicate with foreground processes through this region.

Background and foreground processing are explained in more detail in the job list head cell (JBLH) audit description.

The channel control block common area audit (CCBCOM) checks for incorrect data in the channel control block common area. Data errors can create incorrect call processing inputs, or can put the channel control block common area in a limbo condition, which results in lost resources for call processing.

This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

This audit operates on the linkage areas of the channel control block common area relation (RCCBCOM). The RLCCBCOM relation has a one-to-one linkage with the channel data block relation (RLCHDB) and a special linkage with the high-level service circuit relation (RLHLSC). Its owner linkages are with the following members:

- The path data block relation (RLPHDB).
- The circuit queuing block relation (RLCQB).
- The circuit data block common area relation (RLCDBCOM).
- RLCHDB.

Its member linkages are with the following owners:

- The process control block linkage area relation (RLPCBLA).
- The job list head cell relation (RLJBLH).

Figure 2 illustrates the layout of a channel control block common area and Figure 3 illustrates the rCCBCOM linkages.

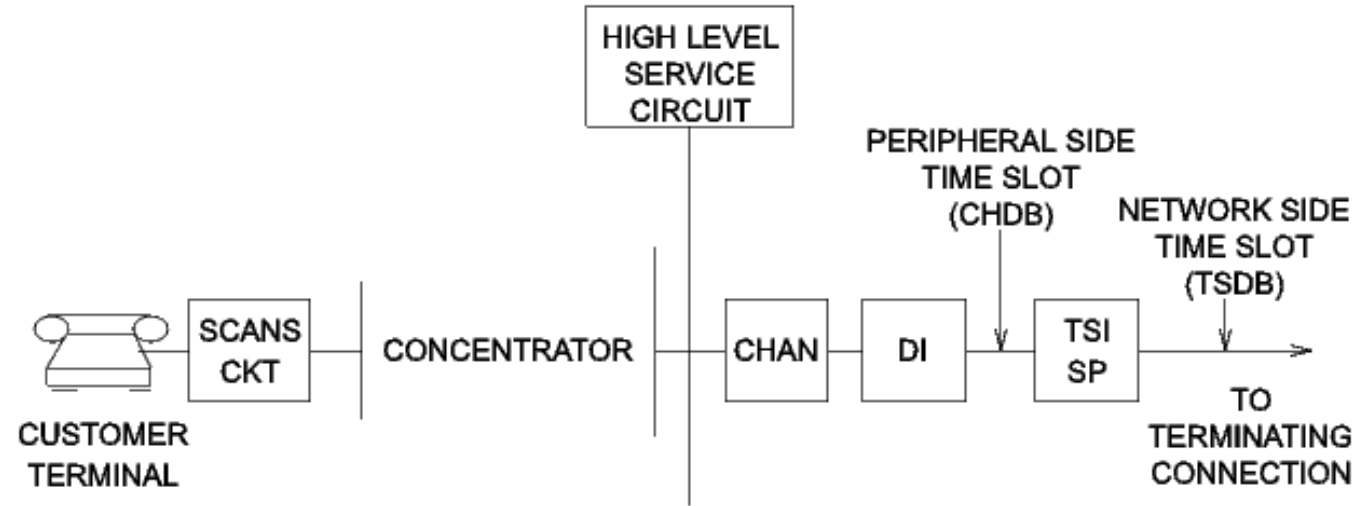


Figure 1 (CCBCOM-1) Path from Customer Terminal to TSI

LOGICAL KEY (ccbnum)
ACTIVE PORT NAME (port)
LENGTH OF TIME FOREGROUND JOB QUEUED (jobage)
LENGTH PORT TYPE (lptype)
LOGICAL PORT STATE (lpstate)

Figure 2 (CCBCOM-2) Channel Control Block Command Area

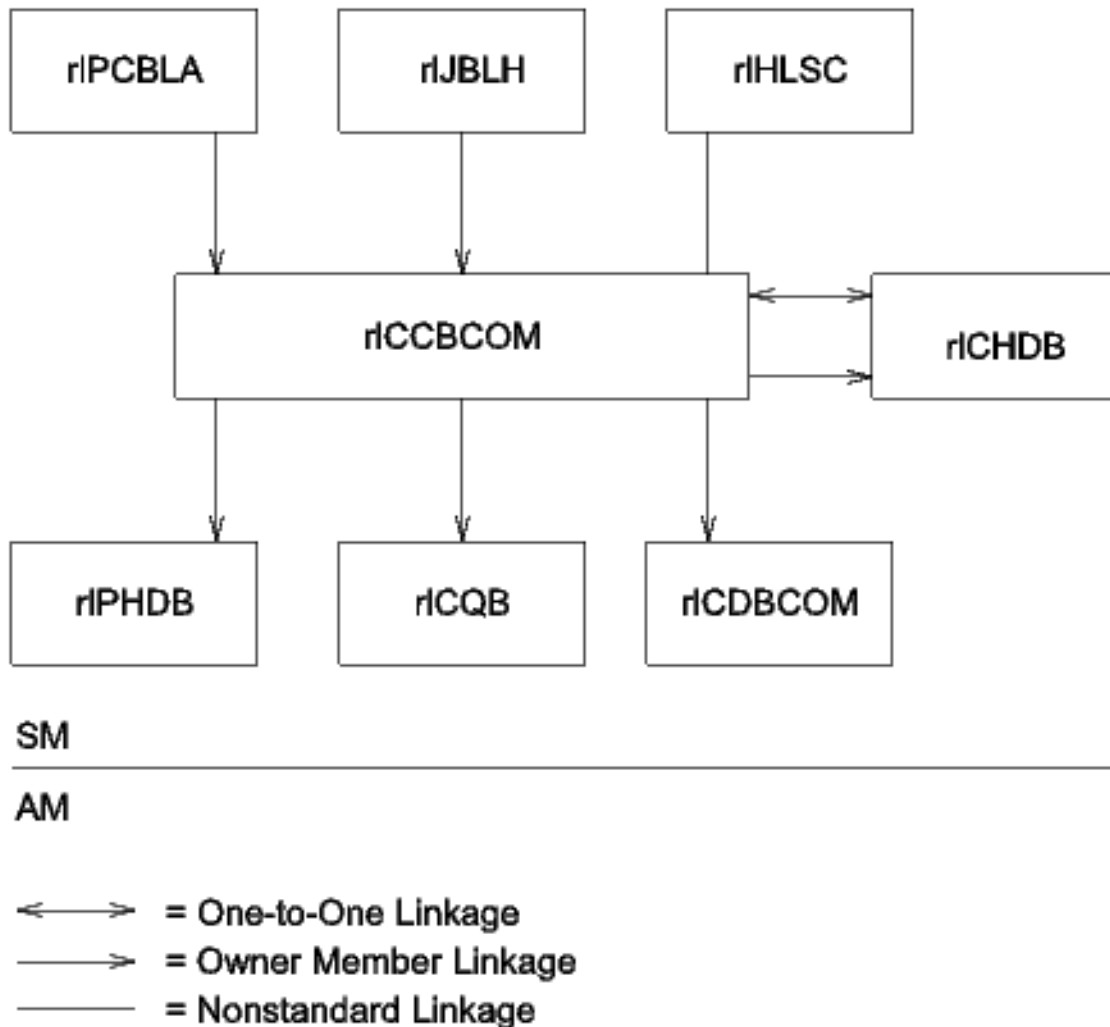


Figure 3 (CCBCOM-3) rCCBCOM Linkages

3. AUDIT CHECKS

To verify the linkages associated with the channel control block common areas, the CCBCOM audit calls the linkage driver (DRLINK) once for each audit segment. DRLINK checks the one-to-one linkages with rCHDB and the owner-member linkages with rCDBCOM, rCHDB, rCQB, rJBLH, rPCBLA, and rPHDB. After verifying the linkages, the DRLINK driver invokes a semantic check function that performs the following checks:

- The audit checks the logical port state (lpstate) field to ensure that it is a valid state variable.
- The audit follows the pointer from the channel control block foreground region to the high-level service circuit for a ringing call. The ccbusr field in the rHLSC tuple should contain the number of the channel control block that is in the ringing state. The audit does not make this point-to/point-back check for SLC[®] 96 ringing jobs because the SLC[®] 96 does not use high-level service circuits.
- The audit checks the lpstate field against the channel control block common area linkages. A channel control block common area in the idle state should be linked to the idle lists so it can be used by call processing. All its linkages that are not involved in the idle list mapping must be zeroed. The linkages for a channel control block common area in a stable state should be error-free; if they are not, a bad linkage may cause an assert before the audit finds the error.

- (d) The rCCBCOM tuple should not be in the transient state for more than five minutes or a call in progress will not be able to disconnect. The audit times the lpstate field for a change from the transient state.

4. ERROR HANDLING

Any tuples that fail these checks are initialized and placed on the idle list.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	A rCCBCOM in B-channel state has different port name from that indicated by rPORTLA.
Possible Error Effect:	The bad port name may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Port name in rCCBCOM.
<i>Good Data =</i>	Key of rPORTLA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The rISPABUF tuple is linked to the list which is reserved for routine port conditioning (RPC) jobs. However the tuple is not being used for a RPC job.
Possible Error Effect:	ISPA could be preventing RPC jobs from running.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rISPABUF tuple.
<i>Bad Data =</i>	The current ISPA job type.
<i>Good Data =</i>	The ISPA job type for RPC jobs.
<i>Logical Index =</i>	Key of rISPABUF tuple.
<i>Dump =</i>	rISPABUF tuple.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	A rCCBCOM in ISLU D-channel cut through or B-channel state has an error in one of its linkages.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rCCBCOM tuple. This action terminates the call that uses the rCCBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCCBCOM.
<i>Bad Data =</i>	Linkage ID in the rCCBCOM.
	0 = rPCBLA owner linkage.
	1 = rPHDB head linkage.
	2 = rPHDB tail linkage.

	3 = rCHDB head linkage.
	4 = rCHDB tail linkage.
	5 = rCHDB one to one linkage.
	6 = rCDBCOM linkage.
	7 = rPTSB linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

Table 4 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	The rgbus circuit name that was calculated from the port in the present rCCBCOM tuple does not correspond to the rIRGBUS tuple that owns the corresponding rCCBUSR tuple for this ISLU-Z ringing.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The rCCBCOM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM tuple.
<i>Bad Data</i> =	rgbus.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key to tuple.
<i>Dump 1</i> =	rCCBCOM tuple.
<i>Dump 2</i> =	rIRGBUS tuple.

Table 5 Error Code: CCB_HLSC

Error Code:	CCB_HLSC
Error Description:	The high-level service circuit (HLSC) for a ringing call does not point to this rCCBCOM.
Possible Error Effect:	A ringing call may be stuck forever.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rCCBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	rCCBCOM logical key.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump 1</i> =	rCCBCOM tuple.
<i>Dump 2</i> =	rHLSC tuple.

Table 6 Error Code: HLSC

Error Code:	HLSC
Error Description:	The HLSC for a ringing call does not point to this rCCBCOM.
Possible Error Effect:	A ringing call may be stuck forever.
Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved. The audit attempts to find the owner of the rCCBCOM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	Address of rHLSC.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump 1</i> =	rCCBCOM tuple.
<i>Dump 2</i> =	rHLSC tuple.

Table 7 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rICCB COM is in the idle state but is not linked to the rICCB COM idle list.
Possible Error Effect:	The rICCB COM is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rICCB COM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	Owner link of rICCB COM.
<i>Good Data</i> =	Physical key of idle head cell.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 8 Error Code: KEYFUNC

Error Code:	KEYFUNC
Error Description:	The tdbmap in the foreground region shows that a tdb is involved in ISLU-Z ringing, but the corresponding tuple of the PCtdbtbl for that tdb shows that it is not.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The rICCB COM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM tuple.
<i>Bad Data</i> =	CCB COM information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rICCB COM tuple.

Table 9 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	There does not appear to be linkage from the process (rIPCBLA) to the port (rI PORTLA).
Possible Error Effect:	The port is lost to this process, therefore the call is dead.
Corrective Action Taken:	The rICCB COM tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	Key (process number) of the rIPCBLA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM tuple.
<i>Dump</i> =	rICCB COM tuple.

Table 10 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The <i>ccbnum</i> attribute in the CCB FG (foreground) region (rICCB COM.ccbfr), the first extended CCB FG region (rICCB COM.ccbef1), or the second extended CCB FG region (rICCB COM.ccbef2) of the rICCB COM tuple is incorrectly set. These attributes should be set to the same value as the logical key (rICCB COM.ccbnum) of the rICCB COM tuple.
Possible Error Effect:	May cause asserts and or interrupts when a call is being idle.
Corrective Action Taken:	The audit will reinitialize the rICCB COM tuple and place the tuple onto the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The CCB number from the first extended CCB FG region (rICCB COM.ccbef1.ccbnum).
<i>Bad Data</i> =	The CCB number from the second extended CCB FG region (rICCB COM.ccbef2.ccbnum).
<i>Good Data</i> =	The CCB number from the CCB FG region (rICCB COM.ccbfr.ccbnum).
<i>Logical Index</i> =	Logical key of the rICCB COM tuple.
<i>Dump</i> =	rICCB COM tuple.

Table 11 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	The process owning rICCB COM tuple in B-channel state can not get the B-channel rI PORTLA.
Possible Error Effect:	The error may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICCB COM tuple. This action terminates the call that uses the rICCB COM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA (current process).
<i>Bad Data</i> =	Logical key or rIPCBLA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 12 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The lu_num or tdbno fields associated with the PC foreground region of the rICCB COM has out of range values. This rICCB COM is associated with the peripheral task data blocks with the lpstate equal to PCPTASK.
Possible Error Effect:	The error may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICCB COM tuple. This action terminates the PTDB that uses the rICCB COM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The lu_num value of the PC foreground region, ccbfq_ptr→fgwrkb.JBistask.lu_num
<i>Bad Data</i> =	The tdbno value of the PC foreground region, ccbfq_ptr→fgwrkb.JBistask.tdb.tdbhead.tdbno
<i>Good Data</i> =	PCTOTPUNTS, total number of PCT link based peripherals.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	The PC foreground region, ccbfq_ptr.

Table 13 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A rICCB COM in B-channel state has an error linkage to rIPCBLA (owner tuple).
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICCB COM tuple. This action terminates the call that uses the rICCB COM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	Logical key of rICCB COM.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 14 Error Code: STABLE_LINK

Error Code:	STABLE LINK
Error Description:	A rICCB COM in a stable state has an error in one of its linkages.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICCB COM tuple. This action terminates the call that uses the rICCB COM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	Linkage ID in the rICCB COM. 0 = rIPCBLA owner linkage.

	1 = rIPHDB head linkage.
	2 = rIPHDB tail linkage.
	3 = rICHDB head linkage.
	4 = rICHDB tail linkage.
	5 = rICHDB one to one linkage.
	6 = rICDBCOM head linkage.
	7 = rIPTSB linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 15 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rICCB COM has an invalid state variable.
Possible Error Effect:	The state of the rICCB COM is unknown.
Corrective Action Taken:	The audit idles the rICCB COM tuple and zeros the linkages involved. The audit attempts to find the owner of the rICCB COM and purges the process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	State variable (rICCB COM.lstate).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 16 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than 5 minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The tuple is idled and the call terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump</i> =	rICCB COM tuple.

Table 17 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The tuple has been in the ringing state for more than the time allowed by the global parameter Glringtot.
Possible Error Effect:	A call in progress has been in the ringing state too long.
Corrective Action Taken:	An MGINTERRUPT message is sent to the owning process to clean-up silently.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCB COM.
<i>Bad Data</i> =	length of time it's been ringing.
<i>Good Data</i> =	the value it is allowed to ring (Glringtot).
<i>Logical Index</i> =	Logical key of rICCB COM.
<i>Dump 1</i> =	rICCB COM tuple.
<i>Dump 2</i> =	rPCBLA tuple.

Table 18 Error Code: TASK

Error Code:	TASK
Error Description:	A rCCBCOM tuple in the peripheral control task state (rCCBCOM.lstate = PCTASK) failed the point-to-point-back check between the corresponding peripheral control channel control block foreground region (PCccbfgr[rCCBCOM.ccb]) and the peripheral control task data block (PCtdtbl[] array). Depending upon the loadable package, this error can also come out for the peripheral control timeslot (PCT) link based tasks. Here, an rCCBCOM tuple with lstate equal to PCPTASK, failed the point to point back checks between corresponding peripheral control channel control block foreground region and the peripheral task data block (PTDB) associated with it.
Possible Error Effect:	The rCCBCOM tuple will be lost to the system causing resource failures which may result in lost calls.
Corrective Action Taken:	The rCCBCOM tuple will be reinitialized and linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rCCBCOM tuple.
<i>Bad Data =</i>	NA. Or for PTDBs, the ccbnum field of the PCptdbtbl[][] accessed by using the logical unit and PTDB number from the PC foreground region data.
<i>Good Data =</i>	NA. Or for PTDBs, the ccbnum field of the PCptdbtbl[][] accessed by using the logical unit and PTDB number from the PC foreground region data.
<i>Logical Index =</i>	Key of the bad rCCBCOM tuple.
<i>Dump 1 =</i>	PC foreground region (of type struct PCCBPG).
<i>Dump 2 =</i>	The PCtdtbl[] array. (Not for PTDBs)

Table 19 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The high level service circuit using this CCB must point to it. However, the unit number is null.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rCCBCOM tuple will be recovered and returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Line unit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCCBCOM tuple.
<i>Dump =</i>	NA.

Table 20 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	Only one ISPA RPC job is allowed to be scheduled at a time. Since this rIISPABUF tuple is not the first one on the list of RPC jobs, it is invalid.
Possible Error Effect:	Someone other than the RPC scheduler may be requesting RPC jobs.
Corrective Action Taken:	The rIISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rIISPABUF tuple.
<i>Bad Data =</i>	Logical key of the previous rIISPABUF tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Key of the bad rIISPABUF tuple.
<i>Dump =</i>	rIISPABUF tuple.

Table 21 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rCCBCOM is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rCCBCOM on the idle list and may be unusable by call processing.

Corrective Action Taken:	The audit idles the rCCBCOM tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCCBCOM.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rCCBCOM.
<i>Dump</i> =	rCCBCOM tuple.

CCBUSR

Software Release: 5E10 and later

1. CHANNEL CONTROL BLOCK USER (CCBUSR) AUDIT

The integrated services line unit (ISLU) Z analog interface (ISLU-Z) provides analog call processing on the ISLU, including plain old telephone service (POTS) calls, multiparty calls, and coin phone operations on the ISLU. Several audits check the data structures used by the ringing generators (RGs), metallic access network (MAN), and sparing of faulty line cards. The audits include the peripheral control (PC) ringing occupancy table audit (RGOcup), the PC ringing task data block audit (RGTDDB), the ringing generator bus audit (RgBUS), the channel control block user audit (CCBUSR), the ISLU-Z access network bus audit (ISANBUS), the spare line card audit (SPRCRD), and the ISLU transaction audit (ISLUTRAN). The CCBUSR audit ensures the accuracy of the channel control block numbers for ringing processes. The CCBUSR audit is part of the Peripheral Control audits subgroup. It is part of the ISLU application loadable package (ISLUAPP). It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The CCBUSR audit uses the following data structures: the channel control block user relation (RLCCBUSR), the channel control block common area relation (RLCCBCOM), and the ringing generator bus relation (RLRgBUS). The RLCCBUSR relation is a member of the direct access RLRgBUS relation. The RLCCBUSR relation stores the channel control block numbers used for ISLU-Z ringing. These data structures are shown in Figure 1.

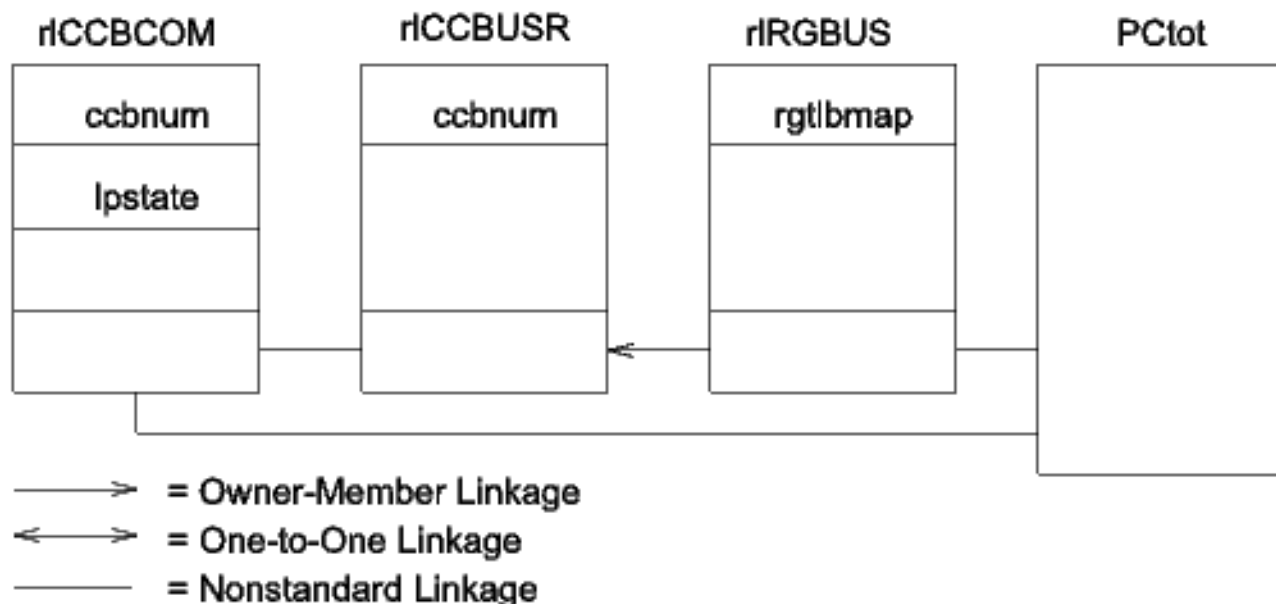


Figure 1 (CCBUSR-1) Data Structures of the CCBUSR Audit

3. AUDIT CHECKS

The CCBUSR audit calls the indirect access sequential key driver (DRINASEK) to verify the linkages between the RLCCBUSR and RLRgBUS relations. When the linkages are verified, the semantic check function checks that the local port state field (lpstate) in RLCCBCOM, which corresponds to the same channel control block number in RLCCBUSR, has a value of PCRING or PCPXSTATE.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	A rICCBUSR tuple involved in ringing corresponds to an rICCBCOM tuple whose lpstate is not PCRING or PCPXSTATE, and the rICCBCOM is not idle.
Possible Error Effect:	Ringing data has been garbaged.
Corrective Action Taken:	The rICCBUSR tuple is recovered, reinitialized, and put back on the idle list. The corresponding rICCBCOM tuple is quarantined.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCBUSR tuple.
<i>Bad Data =</i>	ccbnum from rICCBUSR.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rICCBUSR tuple.
<i>Dump 1 =</i>	rICCBUSR tuple.
<i>Dump 2 =</i>	rICCBCOM tuple.

CCEST-A

Software Release: 5E10 - 5E12

1. COMMON CONTROL EQUIPMENT STATUS TABLE (CCEST) AUDIT

The CCEST audit is responsible for ensuring that the maintenance status and ownership of digital networking unit-synchronous optical network (SONET) (DNU-S) circuits is synchronized between the switching module (SM) and the DNU-S CC.

Basic maintenance status (such as in-service, out-of-service, and so forth) is kept for all DNU-S circuitry in the DNU-S CC processor. Ownership status for certain circuits (such as CC, CD, and TMUX) is also maintained there. A task interface is used to obtain this data from the DNU-S CC and compare it to maintenance status kept in the switching module. Multiple tasks are sent down to the DNU-S CC to collect ownership and maintenance status information. The HWDCAUSET task is used in two ways. First the HWDCAU_HW type is used to collect ownership and maintenance status for CDs, SFIs, TMUXs. Next the HWDCAU_FAC type is used on a per-STE basis to collect maintenance status for an STE and its STS1s, VT1.5s, and DS1s. The audit allows any discrepancy to exist for one execution of the audit.

2. DATA STRUCTURES

Data for CDs, SFIs, and TMUXs is collected in the hwdCHARDWARE structure type. Data for an STE and its STS1s, VT1.5s and DS1s is collected in the hwdCFACILITY structure type. Ownership status for CDs, SFIs and TMUXs is compared against the *stat_act* field relation RLSMEST in the switching module. All maintenance status is compared against the dynamic relation *bas_stat* and *qual_2* fields of RLSMEST in the switching module.

3. AUDIT CHECKS

CCBCOM_LINK = Verify an rCCBCOM tuple was obtained.

CDBCOM = Verify an rCDBCOM tuple was obtained.

DBOWNERIDLE = Verify that a process owns the circuit if the CCEST data indicates the circuit is owned.

MAINT_STATE = Verify that maintenance status in rLSMEST agrees with that in CCEST.

ZAP = The audit was scheduled as a result of a memory interrupt.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	Could not obtain an associated rCCBCOM tuple.
Possible Error Effect:	Various asserts or other failures from PC/SMIM.
Corrective Action Taken:	Schedule the CKTDATA audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address returned from DA.
<i>Bad Data</i> =	The <i>ccbnum</i> of the rCCBCOM tuple.
<i>Good Data</i> =	The <i>cdccm.st_ownk</i> field of the rCDBCOM tuple - DBSFCCBCOM.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	The rCCBCOM tuple.
<i>Dump 2</i> =	The rCDBCOM tuple.

Table 2 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	Could not obtain an associated rCDBCOM tuple.
Possible Error Effect:	Various asserts or other failures from PC/SMIM.
Corrective Action Taken:	Schedule the CKTDATA audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address returned from DA.
<i>Bad Data =</i>	The <i>name</i> field of the rCDBCOM tuple.
<i>Good Data =</i>	The <i>unit_id</i> field of the rSMEST tuple.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	The rCDBCOM tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 3 Error Code: DBOWNERIDLE

Error Code:	DBOWNERIDLE
Error Description:	A DNU-S circuit is marked as "owned" in the DNU-S CC, but no valid owner exists in the switching module.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	A task is sent to the DNU-S to release ownership of the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The data group number in which the circuit resides.
<i>Bad Data =</i>	The <i>stat_act</i> field of the rSMEST tuple.
<i>Good Data =</i>	DBBUSY (1).
<i>Logical Index =</i>	The index used to the task return array.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The entire task return structure.

Table 4 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance status of a DNU-S circuit does not agree between the DNU-S CC and the switching module.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	The circuit is placed on the MRA recovery queue for remove and restore.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The data group number in which the circuit resides.
<i>Bad Data =</i>	The <i>bas_stat</i> and <i>qual_2</i> field of the circuit in CCEST.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The index used to the task return array.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The entire task return structure.

CCEST-B

Software Release: 5E13 and later

1. COMMON CONTROL EQUIPMENT STATUS TABLE (CCEST) AUDIT

The digital networking unit-synchronous optical network (SONET) (DNU-S) is a high capacity digital transport peripheral. Each DNU-S common control (CC) processor circuit contains an equipment status table (EST) for all DNU-S circuitry. The switching module (SM) also contains an EST for all DNU-S circuitry which is stored in the RLSMEST relation. The CCEST audit ensures that the ownership status, maintenance status, and interrupt inhibit administrator (IIA) status of DNU-S circuits is synchronized between the SM and the DNU-S CCs. The CCEST audit runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLSMEST relation, the hwdCHARDWARE structure, and the hwdCFACILITY structure, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	A DNU-S circuit's <i>qual_2</i> is marked "PROOS" in the DNU-S CCEST data, but the circuit is not busy in the DNU-S or the SM.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	The circuit is placed on the MRA recovery queue for remove and restore.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The data group number in which the circuit resides.
<i>Bad Data</i> =	The <i>qual_2</i> of the circuit in CCEST.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index used to the task return array.
<i>Dump 1</i> =	The rLSMEST tuple.
<i>Dump 2</i> =	The entire task return structure.

Table 2 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	Could not obtain an associated rCCBCOM tuple.
Possible Error Effect:	Various asserts or other failures from PC/SMIM.
Corrective Action Taken:	Schedule the CKTDATA audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address returned from DA.
<i>Bad Data</i> =	The <i>ccbnum</i> of the rCCBCOM tuple.
<i>Good Data</i> =	The <i>cdccm.st_ownk</i> field of the rCDBCOT tuple - DBSFCCBCOM.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	The rCCBCOM tuple.
<i>Dump 2</i> =	The rCDBCOT tuple.

Table 3 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	Could not obtain an associated rCDBCOM tuple.
Possible Error Effect:	Various asserts or other failures from PC/SMIM.
Corrective Action Taken:	Schedule the CKTDATA audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address returned from DA.
<i>Bad Data =</i>	The <i>name</i> field of the rCDBCOM tuple.
<i>Good Data =</i>	The <i>unit_id</i> field of the rSMEST tuple.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	The rCDBCOM tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 4 Error Code: DBOWNERIDLE

Error Code:	DBOWNERIDLE
Error Description:	A DNU-S circuit is marked as "owned" in the DNU-S CC, but no valid owner exists in the switching module.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	A task is sent to the DNU-S to release ownership of the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The data group number in which the circuit resides.
<i>Bad Data =</i>	The <i>stat_act</i> field of the rSMEST tuple.
<i>Good Data =</i>	DBBUSY (1).
<i>Logical Index =</i>	The index used to the task return array.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The entire task return structure.

Table 5 Error Code: IIA

Error Code:	IIA
Error Description:	The IIA status of the DNU-S circuit is inconsistent between the DNU-S and the SM.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	The circuit is placed on the MRA recovery queue for remove and restore.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The data group number in which the circuit resides.
<i>Bad Data =</i>	The IIA state of the circuit in CCEST.
<i>Good Data =</i>	The <i>pinhype</i> field of the rSMEST tuple.
<i>Logical Index =</i>	The <i>unit_id</i> field of the rSMEST tuple.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The entire task return structure.

Table 6 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Could not obtain a static rITSP2APP tuple for the DNU-S DS1.
Possible Error Effect:	The DNU-S which is a transport may not be able to communicate with its application.
Corrective Action Taken:	None.
Manual Action Required:	A static rITSP2APP tuple should exist for every DNU-S DS1 circuit. Refer to the Logical Index information for the circuit name which is the key to the missing rITSP2APP tuple.
Dump Description:	
<i>Error Address =</i>	The data group number in which the circuit resides.
<i>Bad Data =</i>	The relation ID for RLTPSP2APP.
<i>Good Data =</i>	The <i>unit_id</i> field of the rSMEST tuple.
<i>Logical Index =</i>	The <i>unit_id</i> field of the rSMEST tuple.
<i>Dump =</i>	The rSMEST tuple.

Table 7 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance status of a DNU-S circuit does not agree between the DNU-S CC and the switching module.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	The circuit is placed on the MRA recovery queue for remove and restore.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The data group number in which the circuit resides.
<i>Bad Data</i> =	The <i>bas_stat</i> and <i>qual_2</i> field of the circuit in CCEST.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index used to the task return array.
<i>Dump 1</i> =	The rSMEST tuple.
<i>Dump 2</i> =	The entire task return structure.

Table 8 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The peripheral control and timing (PCT) busy bit is stuck indicating busy.
Possible Error Effect:	The circuit can not be obtained for circuit administration.
Corrective Action Taken:	A command is sent to the DNU-S to clear the PCT busy bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The data group number in which the circuit resides.
<i>Bad Data</i> =	The value of the PCT busy bit (SMPBITON).
<i>Good Data</i> =	The value the PCT busy bit should be (SMPBITOFF).
<i>Logical Index</i> =	The index used to the task return array.
<i>Dump</i> =	The rSMEST tuple.

CCGINFO

Software Release: 5E11 and later

1. SCCP GLOBAL INFORMATION TABLE (CCGINFO) AUDIT

CCGINFO audit steps through the RLFIX_DG (fix digit global title translation) static relation and checks the data in CC7GTT_INFO table. Error in the CC7GTT_INFO table will result in rebuilding the CC7GTT_INFO table.

2. DATA STRUCTURES

For details on the fields contained in the RLFIX_DG relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The number of valid entries in the CC7GTT_INFO table (as counted by the audit) exceeds the entry count in the table dictionary.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with protocol handler (PH) updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Table ID (CC7GTT_INFO).
<i>Bad Data =</i>	Number of entries from the table dictionary.
<i>Good Data =</i>	Number counted by the audit.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The size of the CC7GTT_INFO table has grown beyond the maximum size that signaling connection control part (SCCP) audits allow (AUCCGBITS). Change the definition of AUCCGBITS to an appropriate larger value.
Possible Error Effect:	This table is not audited.
Corrective Action Taken:	The audit returns to audit control without performing any checks on the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Maximum table size allowed by SCCP audits
<i>Good Data =</i>	Maximum table size from the table dictionary
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: INFO_ERR

Error Code:	INFO_ERR
Error Description:	A CCGT_INFO table entry has been found to contain an erroneous field. The <i>Error Address</i> value can help to determine which particular field the audit is complaining about. The following list matches the <i>Error Address</i> value with the field in error: 2 = num_pgs 5 = pg_selct 6 = page_size
Possible Error Effect:	Unknown.
Corrective Action Taken:	The incorrect field is corrected, and the PH copies are updated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	See previous Error Description.
<i>Bad Data</i> =	Incorrect field value.
<i>Good Data</i> =	Correct field value.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire dynamic entry.

Table 4 Error Code: INSERT_FAILED

Error Code:	INSERT_FAILED
Error Description:	An attempt to insert an entry into the CCGT_INFO table failed. It exists in RLfix_dig, but not in the table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 5 Error Code: INV

Error Code:	INV
Error Description:	A CCGT_TRAN page owned by a CCGT_INFO entry overlaps with a page owned by another such entry.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the table entry that is in error.
<i>Bad Data</i> =	Offset field value of the above entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The block number offset that overlaps.
<i>Dump</i> =	Contents of the erroneous table entry.

Table 6 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A static RLfix_dig relation tuple has been found to have no counterpart in the CCGT_INFO table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	An attempt is made to insert the entry (with values drawn from the static tuple).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.

<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 7 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	An entry has been found in the CCGT_INFO table that does not exist in the static relation RLfix_dig.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The entry is deleted, and the PH copies are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the beginning of the non-null entry.
<i>Bad Data</i> =	Offset (into table) of the non-null entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire non-null entry.

Table 8 Error Code: NOT_SORTED

Error Code:	NOT_SORTED
Error Description:	The CCGT_INFO table is not sorted.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the first table entry that is greater than or equal to the one following it.
<i>Bad Data</i> =	Index of the <i>Error Address</i> entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Contents of the table entry in <i>Error Address</i> .
<i>Dump</i> =	Contents of the table entry that follows the entry in <i>Error Address</i> .

Table 9 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The offset field of a CCGT_INFO table entry is out of range.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the table entry that is in error.
<i>Bad Data</i> =	Offset field value of the <i>Error Address</i> entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Contents of the erroneous table entry.

CCGTBL

Software Release: 5E11 and later

1. SCCP GLOBAL TITLE TRANSLATION TABLE (CCGTBL) AUDIT

CCGTBL audit steps through the static relation RLTRAN_GP (global title translation group) and checks the data in CC7GTT_TBL table. Error in the table will result in CC7GTT_TBL being rebuilt.

2. DATA STRUCTURES

For details on the fields contained in the RLTRAN_GP relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The number of valid entries in the CC7GTT_TBL table (as counted by the audit) does not match the entry count in the table dictionary.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with protocol handler (PH) updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Table ID (CC7GTT_TBL).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The size of the CC7GTT_TBL table has grown beyond the maximum size that signaling connection control part (SCCP) audits allow (AUCCGBITS). Change the definition of AUCCGBITS to an appropriate larger value.
Possible Error Effect:	This table is not audited.
Corrective Action Taken:	The audit returns to audit control without performing any checks on the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Maximum table size allowed by SCCP audits.
<i>Good Data =</i>	Maximum table size from the table dictionary.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: INSERT_FAILED

Error Code:	INSERT_FAILED
Error Description:	An attempt to insert an entry into the CC7GTT_TBL table failed. It exists in rTRAN_GP, but not in the table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 4 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A static rTRAN_GP relation tuple has been found to have no counterpart in the CC7GTT_TBL table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	An attempt is made to insert the entry (with values drawn from the static tuple).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 5 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-null entry has been found (in the CC7GTT_TBL table) that does not exist in the static relation RLTRAN_GP.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The entry is nulled out (deleted), and the PH copies are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the beginning of the non-null entry.
<i>Bad Data</i> =	Offset (into table) of the non-null entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire non-null entry.

Table 6 Error Code: NOT_SORTED

Error Code:	NOT_SORTED
Error Description:	The CC7GTT_TBL table is not sorted.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the first table entry that is greater than or equal to the one following it.
<i>Bad Data</i> =	Index of the entry in the error address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Contents of the table entry in the error address.
<i>Dump 2</i> =	Contents of the table entry that follows the entry in the error address.

Table 7 Error Code: TBL_ERR

Error Code:	TBL_ERR
Error Description:	A CC7GTT_TBL table entry has been found to differ from its counterpart in rTRAN_GP. The error address value can help to determine which particular field the audit is complaining about. The following list matches the error address

	value with the field in error. 5 = st_octet 6 = dest_net
Possible Error Effect:	Unknown.
Corrective Action Taken:	The incorrect field is corrected from static data, and the corresponding update is made on all PHs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	See previous error description.
<i>Bad Data</i> =	Dynamic field value.
<i>Good Data</i> =	Static field value.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire dynamic entry.

CCGTT

Software Release: 5E11 and later

1. SCCP GLOBAL TITLE TRANSLATION TYPE TABLE (CCGTT) AUDIT

CCGTT audit steps through the static relation RLGTT_MASK (global title translation for address digits) and check the data in the CCTRAN_TYP and CCGTT_TRAN table. Error in the table will result in the table being rebuilt.

2. DATA STRUCTURES

For details on the fields contained in the RLGTT_MASK relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error codes.

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The number of valid entries in the CCGTT_TT table (as counted by the audit) does not match the entry count in the table dictionary.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with protocol handler (PH) updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Table ID (CCGTT_TT).
<i>Bad Data =</i>	Dictionary version of the number of entries.
<i>Good Data =</i>	The number of entries found by the audit.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The size of the CCGTT_TT table has grown beyond the maximum size that signaling connection control part (SCCP) audits allow (AUCCGBITS). Change the definition of AUCCGBITS to an appropriate larger value.
Possible Error Effect:	This table is not audited.
Corrective Action Taken:	The audit returns to audit control without performing any checks on the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Maximum table size allowed by SCCP audits.
<i>Good Data =</i>	Maximum table size from the table dictionary.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: INSERT_FAILED

Error Code:	INSERT_FAILED
Error Description:	An attempt to insert an entry into the CCGTT_TT table failed. It exists in rIGTT_MASK but not in the table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild with PH updates, and then it stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 4 Error Code: MISSING

Error Code:	MISSING
Error Description:	The CCGTT_TT entry has no corresponding entry in the CCGT_INFO table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	A request is made to rebuild the CCGTT_TRAN table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire CCGTT_TT table entry.

Table 5 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A static rIGTT_MASK relation tuple has been found to have no counterpart in the CCGTT_TT table.
Possible Error Effect:	Unknown.
Corrective Action Taken:	An attempt is made to insert the entry (with values drawn from the static tuple).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-null entry has been found in the CCGTT_TT table that does not exist in the static relation RLGTT_MASK.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The entry is nulled out (deleted), and the PH copies are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the beginning of the non-null entry.
<i>Bad Data</i> =	Offset (into table) of the non-null entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire non-null entry.

Table 7 Error Code: TT_ERR

Error Code:	TT_ERR
Error Description:	A CCGTT_TT table entry has been found to differ from its counterpart in rIGTT_MASK. The Error Address value can help to determine which particular field the audit is complaining about. The following list matches the <i>Error Address</i> value with the field in error:

	0 = Indicates that a CCGT_INFO tuple has been found with a key matching that of the static RLgtt_mask tuple, but whose num_ent field does not match the static num_fds field. 2 = num_fix 3 = num_req 4 = num_opt
Possible Error Effect:	Unknown.
Corrective Action Taken:	The incorrect field is corrected from static data, and the corresponding update is made on all PHs. If the error address = 0, the corrective action is to request a rebuild of the CCGTT_TRAN table and to refresh the PH copies.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	See the previous Error Description.
<i>Bad Data</i> =	Dynamic field value.
<i>Good Data</i> =	Static field value.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire dynamic entry.

CCSLINK

Software Release: 5E13 and later

1. ELIMINATION OF FALSE CCS LINK CONGESTION (CCSLINK) AUDIT

The CCSLINK audit was developed for the SS7MRE feature. The elimination of false link congestion is defined through the use of T31 timer. T31 timer will be started/stopped/restarted according to the transmit/re-transmit buffer congestion levels. The new timer T31 is audited by CCSLINK audit which runs in CCS PHs PH3 and beyond.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanations in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The timer T31 status is not consistent with the congestion level and timer monitor status.
Possible Error Effect:	The possible error effect is one of the following: • If CCS link is in congestion state and the timer T31 is not running then call processing may be denied. • If CCS link is not in congestion state and the timer T31 is running then eventually link will be takeout of service.
Corrective Action Taken:	This audit will report the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PSCLV3.tidx
<i>Bad Data =</i>	An internal value indicating the error type.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LINK index.
<i>Dump =</i>	NA.

Table 2 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	A timer index in an PSCLV3 is out of range or indexes a timer which is not owned by the PSCLV3 or is of an unexpected type.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit sets the timer index field in the PSCLV3 to the null value (PSNULLTE) or takes other special recovery based on the timer type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used to indicate which timer index was being checked.
<i>Bad Data =</i>	An internal value used to indicate the type of error.
<i>Good Data =</i>	An internal value indicating which entry in an array of timers (as indicated by the <i>Error Address</i> value) was being checked.

<i>Logical Index</i> =	The timer element index.
<i>Dump</i> =	The PSCLV3.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	This error will be printed on the receive-only printer (ROP) and the audit will return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

CCSSTPDB

Software Release: 5E10 and later

1. COMMON CHANNEL SIGNALING 7 SETUP DATA BLOCK (CCSSTPDB) AUDIT

The CCS information is much too much for the stack anymore; the relation RLCCSSTPDB was created to hold the information during call setup.

2. DATA STRUCTURES

The RLCCSSTPDB dynamic relation serves as a holding area for the CCS information (previously on the stack) and is used only during call setup. The CCSSTPDB audit references the following dynamic relations:

RLDB_HEADC = One tuple of this relation acts as the idle list headcell for relation RLCCSSTPDB, linked through **ccsstpdb_mem**.

RLPCBLA = A busy rLCCSSTPDB tuple is linked through **ccsstpdb_mem.st_link** to an rLPCBLA tuple.

The RLCCSSTPDB data structure is presented in Figure 1.

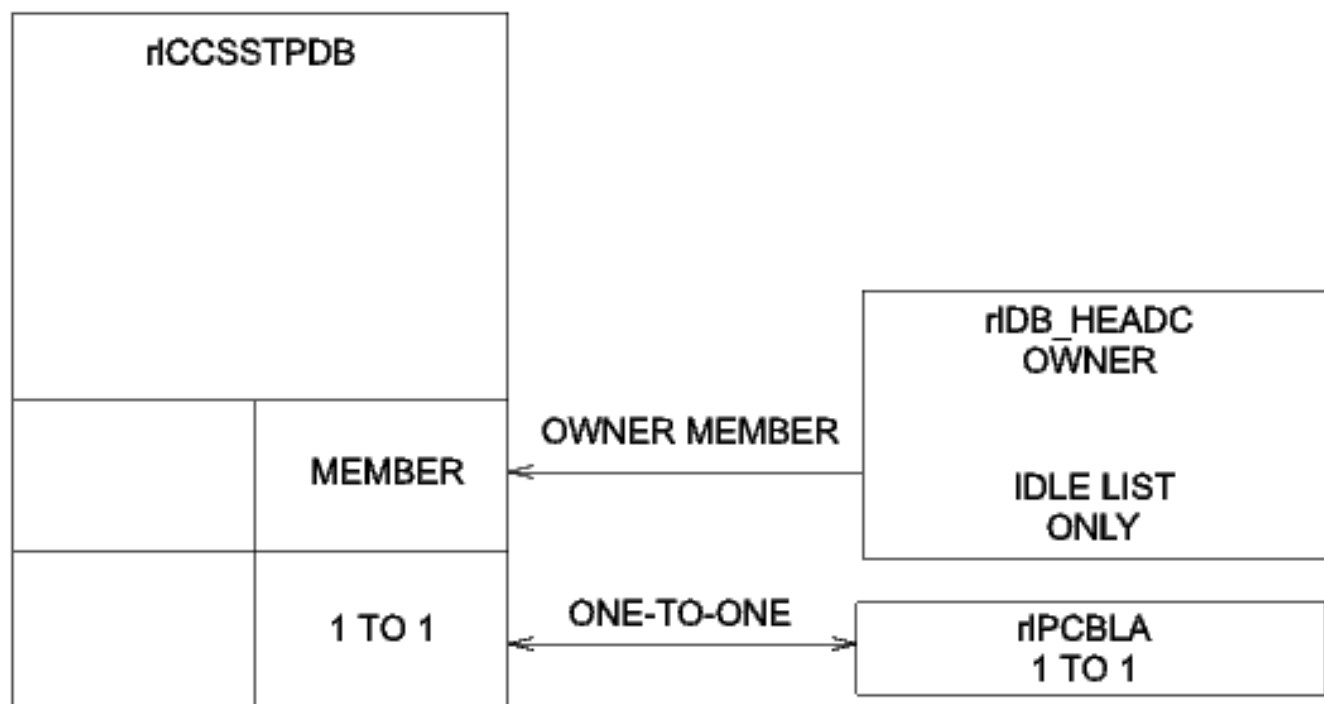


Figure 1 (CCSSTPDB-1) Data Structures of the CCSSTPDB Audit

3. AUDIT CHECKS

The CCSSTPDB audit uses the DRINASEK audit driver. For each rLCCSSTPDB tuple found in the relation, the audit makes the following checks:

DEADPID = The tuple should be linked to an rLPCBLA tuple, when busy.

IDLE_LINK = The owning linkage of the rLCCSSTPDB tuple must be 0 or attached to the idle headcell in RLDB_HEADC.

ZERO_LINK = If the rICCSSTPDB tuple is idle, other linkages should be NULL.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s) and those generated by the DRINASEK driver.

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rICCSSTPDB tuple is not linked to its headcell nor to a rIPCBLA tuple. It must be linked to one or the other.
Possible Error Effect:	The rICCSSTPDB tuple is unusable by call processing.
Corrective Action Taken:	The rICCSSTPDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICCSSTPDB tuple where the error was found.
<i>Bad Data =</i>	The process owning the rICCSSTPDB tuple (rICCSSTPDB.ccsstpdb_mem.st_link).
<i>Good Data =</i>	The rICCSSTPDB tuple link to the idle list (rICCSSTPDB.ccsstpdb_mem.st_ownk = 0).
<i>Logical Index =</i>	The key to the rICCSSTPDB tuple (rICCSSTPDB.stpdb_num).
<i>Dump =</i>	The rICCSSTPDB tuple where the error was found.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rICCSSTPDB tuple is not linked to its headcell nor to a rIPCBLA tuple. It must be linked to one or the other.
Possible Error Effect:	The rICCSSTPDB tuple is unusable by call processing.
Corrective Action Taken:	The rICCSSTPDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICCSSTPDB tuple where the error was found.
<i>Bad Data =</i>	The rICCSSTPDB tuple link to the idle list (rICCSSTPDB.ccsstpdb_mem.st_ownk).
<i>Good Data =</i>	The number of the idle head cell.
<i>Logical Index =</i>	The key to the rICCSSTPDB tuple (rICCSSTPDB.stpdb_num).
<i>Dump =</i>	The rICCSSTPDB tuple where the error was found.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rICCSSTPDB tuple is not linked to its headcell nor to a rIPCBLA tuple. It must be linked to one or the other.
Possible Error Effect:	The rICCSSTPDB tuple is unusable by call processing.
Corrective Action Taken:	The rICCSSTPDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICCSSTPDB tuple where the error was found.
<i>Bad Data =</i>	(baddata).
<i>Good Data =</i>	The rICCSSTPDB tuple link to the idle headcell in RLDB_HEADC (rICCSSTPDB.ccsstpdb_mem.st_ownk).
<i>Logical Index =</i>	The key to the rICCSSTPDB tuple (rICCSSTPDB.stpdb_num).
<i>Dump =</i>	The rICCSSTPDB tuple where the error was found.

CDBCOM-A

Software Release: 5E10 - 5E11

1. CIRCUIT DATA BLOCK COMMON AREA (CDBCOM) AUDIT

The switching resource allocator (SRA) program in the peripheral control subsystem uses circuit data blocks to allocate and deallocate circuits to call processes, idle lists, and out-of-service lists. Each allocatable circuit is assigned a circuit data block in the dynamic circuit data block common area relation (RLCDBCOM). The CDBCOM audit performs a dynamic-to-static existence check with the circuit data relation (RLCKTDATA) to ensure that the circuit data blocks are allocated properly. The CDBCOM audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The CDBCOM audit verifies the existence of tuples in the dynamic circuit data block common area relation (RLCDBCOM), by checking static data in the circuit data relation (RLCKTDATA). As shown in Figure 1, each tuple contains the circuit name, availability information, linkages to the circuit queuing block (RLCQB), the circuit control common area (RLCCBCOM), and the circuit data (RLCKTDATA) relations.

3. AUDIT CHECKS

The audit first checks whether RLCKTDATA is being updated by a recent change/verify transaction. If it is being updated, the CDBCOM audit quits and returns control to the audit control program. If RLCKTDATA is not being updated, the audit invokes the static driver (DRSTATIC) to step through the tuples of RLCDBCOM.

- The audit performs a range check to ensure the validity of each tuple's key.
- If the key is valid, the audit checks whether a corresponding tuple exists in RLCKTDATA. The name fields of each relation are used to make this check.
- If the tuple exists in RLCKTDATA, the audit checks whether it is equipped by reading the equipage state (eqstat) field of RLCKTDATA.

If these checks pass, the audit checks the next tuple in RLCDBCOM. If the tuple does not exist in RLCKTDATA or exists but is unequipped, an error is flagged.

4. ERROR HANDLING

Errors flagged during the checks listed are handled as follows.

- If the audit finds an erroneous key in RLCDBCOM, the RLCDBCOM tuple is updated with the correct key from RLCKTDATA and an error message is formatted.
- If an RLCDBCOM tuple exists for an RLCKTDATA tuple that is unequipped, the RLCDBCOM tuple is removed from the database and an error message is formatted.

If the audit finds no corresponding tuple in RLCKTDATA for the RLCDBCOM tuple being checked, the audit removes the RLCDBCOM tuple.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rSMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rSMEST tuple.

Table 2 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A rCHDB tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	RLCHDB relation ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rCHDB tuple's channel number (chnum).
<i>Dump</i> =	rCHDB tuple.

Table 3 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The associated common board circuit is not transient, indicating peripheral control's (PC) bit map is wrong.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The bit map will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Common board unit index.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Idle list on the common board.
<i>Dump</i> =	NA.

Table 4 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	There should be a dedicated linkage between a tuple of RLCDBCOM (rLCDBCOM.cd ch.link) and a tuple of RLCHDB (rCHDB.ch cd).
Possible Error Effect:	A call will not be able to be set up on a circuit.
Corrective Action Taken:	The audit recovers the rLCDBCOM tuple and reinitializes the RLCDBCOM-to-RLCHDB linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLCDBCOM tuple.
<i>Bad Data</i> =	Circuit type (rCKTDATA.type).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rLCDBCOM tuple.
<i>Dump</i> =	rLCDBCOM tuple.

Table 5 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICDBCOM is associated with a system tone but is not on the DBCADGEN idle list.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDGBCOM tuple.
<i>Bad Data =</i>	The owner of the rICDBCOM tuple.
<i>Good Data =</i>	DBCADGEN head cell value.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump =</i>	rICDBCOM tuple.

Table 6 Error Code: DBCHAN

Error Code:	DBCHAN
Error Description:	The dbchan field of the rICKTDATA static tuple does not match the rICHDB dynamic key.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	As a part of recovering the rICDBCOM tuple, the rICDB will also be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICDBCOM tuple.
<i>Bad Data =</i>	The rICDBCOM to rICHDB link.
<i>Good Data =</i>	The dbchan field of the rICKTDATA field.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 7 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rICDBCOM tuple exists for rICKTDATA.
Possible Error Effect:	Eventual loss of call processing resources.
Corrective Action Taken:	The extra rICDBCOM tuple is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICDBCOM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	rICKTDATA tuple equipage status.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rICDBCOM tuple.

Table 8 Error Code: IDLPSU

Error Code:	IDLPSU
Error Description:	A PSU time slot circuit cdbcom is linked to a chdb which does not have a valid inproc for this type of circuit.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and chdb and re-establishes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	chdb address.
<i>Bad Data =</i>	chdb inproc.
<i>Good Data =</i>	PCIDLPSU.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	cdbcom and chdb tuples.

Table 9 Error Code: ISDNTUP

Error Code:	ISDNTUP
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB

	specified in the logical index field of this dump, but the rIACT_SPR tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIACT_SPR tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIACT_SPR tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: MISSING

Error Code:	MISSING
Error Description:	During reconstruction of a rICDBCOM tuple, an expected associated rICHDB tuple was missing.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The CHDB audit will be run as a related audit of CDBCOM to clean up any rICHDB tuple problem and the rICDBCOM tuple reconstruction will complete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rICHDB tuple.
<i>Bad Data</i> =	The rICHDB tuple's channel number (chnum).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rICHDB tuple's channel number (chnum).
<i>Dump</i> =	rICDBCOM tuple.

Table 11 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous had cell (DBIDLMISC).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Idle link list physical key.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump 1</i> =	CKTDATA tuple.
<i>Dump 2</i> =	CDBCOM tuple.

Table 12 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Manual asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 13 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Linkage (from cdbcom).
<i>Good Data =</i>	Linkage (expected).
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	Circuit type CDBCOM.

Table 14 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rICDBCOM and its associated rITONECH tuple failed a point-to/point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	The constructed logical key of the rICDBCOM tuple using the contents of the rITONECH tuple.
<i>Good Data =</i>	The actual logical key of the rICDBCOM tuple.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump 1 =</i>	rICDBCOM tuple.
<i>Dump 2 =</i>	rITONECH tuple.

Table 15 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The rICDBCOM tuple has been on the transient list since the last time the audit ran. It must be stuck.
Possible Error Effect:	Possible lost resources.
Corrective Action Taken:	The rICDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	The actual idle list physical key.
<i>Good Data =</i>	The physical key of the correct owner.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump 1 =</i>	rICDBCOM tuple.

Table 16 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Actual idle list physical key.
<i>Good Data =</i>	Physical key of correct idle list.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	CDBCOM tuple.

CDBCOM-B

Software Release: 5E12 and later

1. CIRCUIT DATA BLOCK COMMON AREA (CDBCOM) AUDIT

The switching resource allocator (SRA) program in the peripheral control subsystem uses circuit data blocks to allocate and deallocate circuits to call processes, idle lists, and out-of-service lists. Each allocatable circuit is assigned a circuit data block in the dynamic circuit data block common area relation (RLCDBCOM). The CDBCOM audit performs a dynamic-to-static existence check with the circuit data relation (RLCKTDATA) to ensure that the circuit data blocks are allocated properly. The CDBCOM audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The CDBCOM audit verifies the existence of tuples in the dynamic circuit data block common area relation (RLCDBCOM), by checking static data in the circuit data relation (RLCKTDATA). As shown in Figure 1, each tuple contains the circuit name, availability information, linkages to the circuit queuing block (RLCQB), the circuit control common area (RLCCBCOM), and the circuit data (RLCKTDATA) relations.

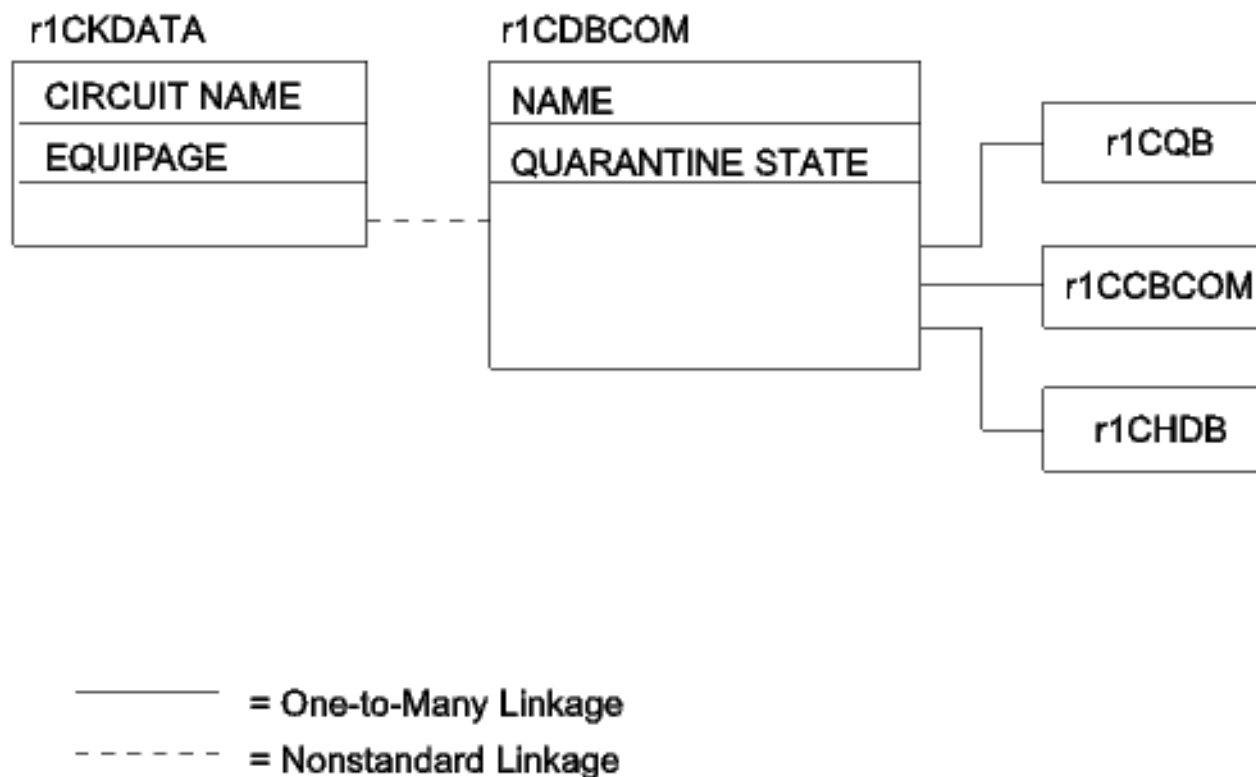


Figure 1 (CDBCOM-1) Data Structures of the CDBCOM Audit

3. AUDIT CHECKS

The audit first checks whether RLCKTDATA is being updated by a recent change/verify transaction. If it is being updated, the CDBCOM audit quits and returns control to the audit control program. If RLCKTDATA is not being updated, the audit invokes the static driver (DRSTATIC) to step through the tuples of RLCDBCOM.

- The audit performs a range check to ensure the validity of each tuple's key.

- If the key is valid, the audit checks whether a corresponding tuple exists in rICKTDATA. The name fields of each relation are used to make this check.
- If the tuple exists in rICKTDATA, the audit checks whether it is equipped by reading the equipage state (eqstat) field of rICKTDATA.

If these checks pass, the audit checks the next tuple in RLCDDBCOM. If the tuple does not exist in RLCKTDATA or exists but is unequipped, an error is flagged.

4. ERROR HANDLING

Errors flagged during the checks listed are handled as follows.

- If the audit finds an erroneous key in rLCDBCOM, the rLCDBCOM tuple is updated with the correct key from rICKTDATA and an error message is formatted.
- If an rLCDBCOM tuple exists for an rICKTDATA tuple that is unequipped, the rLCDBCOM tuple is removed from the database and an error message is formatted.

If the audit finds no corresponding tuple in RLCKTDATA for the rLCDBCOM tuple being checked, the audit removes the rLCDBCOM tuple.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rSMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rSMEST tuple.

Table 2 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A rICHDB tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	RLCHDB relation ID.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rICHDB tuple's channel number (chnum).
<i>Dump =</i>	rICHDB tuple.

Table 3 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The associated common board circuit is not transient, indicating peripheral control's (PC) bit map is wrong.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The bit map will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Common board unit index.
<i>Bad Data =</i>	Circuit type.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Idle list on the common board.
<i>Dump =</i>	NA.

Table 4 Error Code: CDB2CKT

Error Code:	CDB2CKT
Error Description:	The <i>unit</i> , <i>utype</i> , <i>ckt_indx</i> , <i>type</i> , <i>pidb</i> , <i>picb</i> , <i>dbchan</i> , or <i>chnasn</i> fields of this dynamic rICDBCOM tuple do not match the corresponding <i>unit</i> , <i>utype</i> , <i>ckt_indx</i> , <i>type</i> , <i>pidb</i> , <i>baseadd</i> right shifted 8 bits which is the <i>picb</i> , <i>dbchan</i> , or <i>chnasn</i> fields of the associated static rICKTDATA tuple.
Possible Error Effect:	There may be a loss of use or erroneous use of the related circuit.
Corrective Action Taken:	The audit will recover the dynamic rICDBCOM tuple which initializes the rICDBCOM tuple fields to be consistent with the corresponding static rICKTDATA tuple fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rICDBCOM tuple address.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump 1 =</i>	rICDBCOM tuple.
<i>Dump 2 =</i>	rICKTDATA tuple.

Table 5 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	There should be a dedicated linkage between a tuple of RLCDDBCOM (rICDBCOM.cd ch.link) and a tuple of RLCHDB (rICHDB.ch cd).
Possible Error Effect:	A call will not be able to be set up on a circuit.
Corrective Action Taken:	The audit recovers the rICDBCOM tuple and reinitializes the RLCDDBCOM-to-RLCHDB linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICDBCOM tuple.
<i>Bad Data =</i>	Circuit type (rICKTDATA.type).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump =</i>	rICDBCOM tuple.

Table 6 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICDBCOM is associated with a system tone but is not on the DBCADGEN idle list.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDGBCOM tuple.
<i>Bad Data =</i>	The owner of the rICDBCOM tuple.
<i>Good Data =</i>	DBCADGEN head cell value.

<i>Logical Index</i> =	Logical key of the rICDBCOM tuple.
<i>Dump</i> =	rICDBCOM tuple.

Table 7 Error Code: DBCHAN

Error Code:	DBCHAN
Error Description:	The dbchan field of the rICKTDATA static tuple does not match the rICHDB dynamic key.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	As a part of recovering the rICDBCOM tuple, the rICDB will also be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICDBCOM tuple.
<i>Bad Data</i> =	The rICDBCOM to rICHDB link.
<i>Good Data</i> =	The dbchan field of the rICKTDATA field.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 8 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rICDBCOM tuple exists for rICKTDATA.
Possible Error Effect:	Eventual loss of call processing resources.
Corrective Action Taken:	The extra rICDBCOM tuple is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICDBCOM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	rICKTDATA tuple equipage status.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICDBCOM tuple.

Table 9 Error Code: IDLPSU

Error Code:	IDLPSU
Error Description:	A PSU time slot circuit cdbcom is linked to a chdb which does not have a valid inproc for this type of circuit.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and chdb and re-establishes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	chdb address.
<i>Bad Data</i> =	chdb inproc.
<i>Good Data</i> =	PCIDLPSU.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	cdbcom and chdb tuples.

Table 10 Error Code: ISDNTUP

Error Code:	ISDNTUP
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIACT_SPR tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIACT_SPR tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIACT_SPR tuple.

<i>Dump</i> =	NA.

Table 11 Error Code: MISSING

Error Code:	MISSING
Error Description:	During reconstruction of a rCDBCOM tuple, an expected associated rCHDB tuple was missing.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The CHDB audit will be run as a related audit of CDBCOM to clean up any rCHDB tuple problem and the rCDBCOM tuple reconstruction will complete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rCHDB tuple.
<i>Bad Data</i> =	The rCHDB tuple's channel number (chnum).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rCHDB tuple's channel number (chnum).
<i>Dump</i> =	rCDBCOM tuple.

Table 12 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous had cell (DBIDLMISC).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Idle link list physical key.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump 1</i> =	CKTDATA tuple.
<i>Dump 2</i> =	CDBCOM tuple.

Table 13 Error Code: ODDLOSTTUP (1)

Error Code:	ODDLOSTTUP (1)
Error Description:	During rCDBCOM tuple recovery, a static rCKTDATA tuple was unexpectedly found missing for an RISLU2/ISLU2 line card circuit or a static rUNIT_CKT tuple was unexpectedly found missing for an RISLU2/ISLU2 line group controller circuit.
Possible Error Effect:	Asserts and single process purges may be taken due to the static data inconsistencies.
Corrective Action Taken:	None.
Manual Action Required:	The missing rCKTDATA tuple specified by the logical index or the missing rUNIT_CKT tuple whose keys would need to be derived from the logical index should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	rCKTDATA address.
<i>Bad Data</i> =	Line card circuit name.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Line card circuit name.
<i>Dump 1</i> =	rCKTDATA tuple.
<i>Dump 2</i> =	rIACT_SPR tuple.

Table 14 Error Code: ODDLOSTTUP (2)

Error Code:	ODDLOSTTUP (2)
Error Description:	During rCDBCOM tuple recovery, a static rISLU2LG tuple was unexpectedly found missing for an RISLU2/ISLU2 line group controller circuit.
Possible Error Effect:	Asserts and single process purges may be taken due to the static data inconsistencies.

Corrective Action Taken:	None.
Manual Action Required:	The missing rISLU2LG tuple specified by the logical index should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	rICKTDATA address.
<i>Bad Data</i> =	Line card circuit name.
<i>Good Data</i> =	2.
<i>Logical Index</i> =	Line group controller circuit name.
<i>Dump 1</i> =	rICKTDATA tuple.
<i>Dump 2</i> =	rIACT SPR tuple.

Table 15 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Manual asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 16 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Linkage (from cdbcom).
<i>Good Data</i> =	Linkage (expected).
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	Circuit type CDBCOM.

Table 17 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rCDBCOM and its associated rTONECH tuple failed a point-to/point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCDBCOM tuple.
<i>Bad Data</i> =	The constructed logical key of the rCDBCOM tuple using the contents of the rTONECH tuple.
<i>Good Data</i> =	The actual logical key of the rCDBCOM tuple.
<i>Logical Index</i> =	Logical key of the rCDBCOM tuple.
<i>Dump 1</i> =	rCDBCOM tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 18 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The rCDBCOM tuple has been on the transient list since the last time the audit ran. It must be stuck.

Possible Error Effect:	Possible lost resources.
Corrective Action Taken:	The rICDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	The actual idle list physical key.
<i>Good Data =</i>	The physical key of the correct owner.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump 1 =</i>	rICDBCOM tuple.

Table 19 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Actual idle list physical key.
<i>Good Data =</i>	Physical key of correct idle list.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	CDBCOM tuple.

CFLNTRM

Software Release: 5E10 and later

1. CALL FORWARDING LINE TERMINATION (CFLNTRM) AUDIT

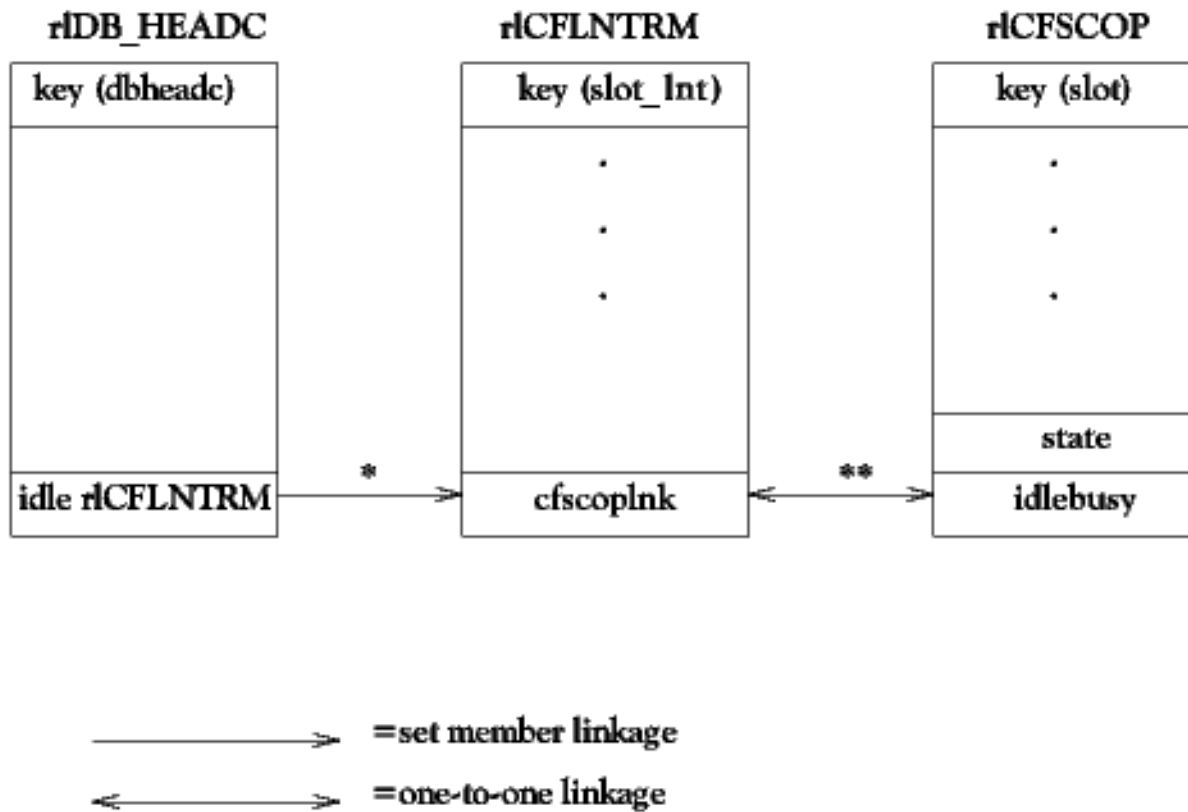
Call forwarding is a feature of the business and residential custom services (BRCS) package that allows one telephone to be forwarded to another telephone. It is assigned in the local office and stored in office-dependent data (ODD). This audit verifies the consistency of the dynamic relation RLCFLNTRM.

2. DATA STRUCTURES

The RLCFLNTRM dynamic relation serves as a holding area for CFDA and CFPF to store necessary portions of the MGLN_TERM message until forwarding is attempted. It is a compacted indexed relation. The CFLNTRM audit references the following dynamic relations:

RLCFSCOP	= Tuples of this relation acts as the busy list headcell for relation RLCFLNTRM, linked through rICFLNTRM.cfscoplInk .
RLDB_HEADC	= One tuple of this relation acts as the idle list headcell for relation RLCFLNTRM, linked through rICFLNTRM.cfscoplInk .

Refer to Figure 1 .



***The rIDB HEADC tuple with the key DBCFLNTIDL is the idle list headcell for RLCFLNTRM.**

****Busy rCFLNTRM tuples go through one-to-one linkage to a rICFSCOP tuple.**

Figure 1 (CFLNTRM-1) Data Structures of the CFLNTRM Audit

3. AUDIT CHECKS

The CFLNTRM audit uses the DRINASEK audit driver. For each rCFLNTRM tuple found in the relation, the audit makes the following checks:

IDLE_LINK = The owning linkage of the rCFLNTRM tuple must be zero or attached to the idle headcell in RLDB_HEADC.

NOTIDLE = If the rCFLNTRM tuple is idle, other linkages should be NULL.

If the rCFLNTRM tuple is busy:

INV_STATE = The rICFSCOP tuple should have a state of Don't Answer or CFPF.

NOTBUSY = The tuple should be linked to an rICFSCOP tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s) and those generated by the DRINASEK driver.

Table 1 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The tuple is not linked correctly to the idle rIDB_HEADC tuple.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rCFLNTRM tuples.
Corrective Action Taken:	The rCFLNTRM tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rCFLNTRM tuple where the error was found.
<i>Bad Data =</i>	The invalid linkage to the rIDB_HEADC tuple (rCFLNTRM.cfscoplнк.st_ownк).
<i>Good Data =</i>	The correct linkage to the rIDB_HEADC tuple.
<i>Logical Index =</i>	The key to the rCFLNTRM tuple (rCFLNTRM.slot_Int).
<i>Dump =</i>	The rCFLNTRM tuple where the error was found.

Table 2 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The rCFLNTRM tuple is linked to a rCFSCOP tuple with a state other than Call Forwarding Don't Answer or Call Forwarding over Private Facilities (rCFSCOP.state != CF_DAWTO or CF_CFPFRRG).
Possible Error Effect:	This resource is unnecessary for this call. This may cause possible call failures if the system exhausts the number of available rCFLNTRM tuples.
Corrective Action Taken:	The rCFLNTRM tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rCFLNTRM tuple where the error was found.
<i>Bad Data =</i>	The state of the rCFSCOP tuple (rCFSCOP.state).
<i>Good Data =</i>	The linkage to the rCFSCOP tuple.
<i>Logical Index =</i>	The key to the rCFLNTRM tuple (rCFLNTRM.slot_Int).
<i>Dump 1 =</i>	The rCFLNTRM tuple where the error was found.
<i>Dump 2 =</i>	The rCFSCOP tuple linked to the rCFLNTRM.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The tuple is not linked to a rCFSCOP tuple; nor to a rIDB_HEADC tuple.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rCFLNTRM tuples.
Corrective Action Taken:	The rCFLNTRM tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rCFLNTRM tuple where the error was found.
<i>Bad Data =</i>	The linkage to the rCFSCOP tuple (rCFLNTRM.cfscoplнк.st_link).
<i>Good Data =</i>	The linkage to the rIDB_HEADC tuple (rCFLNTRM.cfscoplнк.st_ownк).
<i>Logical Index =</i>	The key to the rCFLNTRM tuple (rCFLNTRM.slot_Int).
<i>Dump =</i>	The rCFLNTRM tuple where the error was found.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The tuple is linked incorrectly to both an idle rIDB_HEADC tuple and to a rCFSCOP tuple.
Possible Error Effect:	Since this resource is on the idle list, it may be retrieved at any time and used

	for a call. If this resource is already linked up for a different call, that information will be lost causing call failure.
Corrective Action Taken:	The rCFLNTRM tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rCFLNTRM tuple where the error was found.
<i>Bad Data</i> =	The linkage to the rIDB_HEADC tuple (rCFLNTRM.cfscoplnk.st_ownk).
<i>Good Data</i> =	The linkage to the rCFSCOP tuple (rCFLNTRM.cfscoplnk.st_link).
<i>Logical Index</i> =	The key to the rCFLNTRM tuple (rCFLNTRM.slot_int).
<i>Dump</i> =	The rCFLNTRM tuple where the error was found.

CFSCOP

Software Release: 5E10 and later

1. CALL FORWARDING SYSTEM CAMP-ON PROCESS (CFSCOP) AUDIT

Call forwarding is a feature of the business and residential custom services (BRCS) package that allows one telephone to be forwarded to another telephone. It is assigned in the local office and stored in office-dependent data (ODD).

This audit verifies the consistency of the dynamic relation RLCFSCOP in all SMS, in routine and elevated segmented modes.

2. DATA STRUCTURES

The RLCFSCOP dynamic relation is used by the call forwarding system process (SCOP) mainly to manage the call record information associated with the forwarded leg of a call. It also contains information about the base station and previous and subsequent legs of a forwarded call. The RLCFSCOP entries are populated by a forwarding terminal process and maintained by the SCOP after a routing message has been sent to forward the call. It is a compacted indexed relation.

The CFSCOP audit performs checks on RLCFSCOP semantic data based on the following fields:

<i>bcid</i>	= The call ID associated with this leg of a forwarded call.
<i>bfeatid</i>	= The feature ID of the feature used to forward the call.
<i>bltm_idx</i>	= The ltm index of the forwarder (base station).
<i>bpid</i>	= The process ID of the originator.
<i>idlebusy</i>	= The set-member linkages from the rICFSCOP tuple to the rIDB_HEADC tuple, either to the idle list or to one of the 16 busy lists.
<i>idlebusy.st_link</i>	= The one-to-one linkage from the rICFSCOP tuple to an optional rICFLNTRM tuple.
<i>slot</i>	= The key of the rICFSCOP tuple.
<i>state</i>	= The state of the rICFSCOP tuple, identifying whether the tuple is on the idle list or what type of forwarding is occurring.

The CFSCOP audit references the following dynamic relations:

RLCFLNTRM	= Tuples of this relation are linked when the state of the rICFSCOP tuple is either call forwarding don't answer, waiting for time-out, or call forwarding over private facilities doing digit analysis. They are linked through the one-to-one linkage field of <i>rICFSCOP.idlebusy</i> .
RLDB_HEADC	= One tuple of this relation acts as the idle list headcell for relation RLCFSCOP. Sixteen tuples of this relation act as the busy list headcells for relation RLCFSCOP. All are linked through the set-member linkage of <i>rICFSCOP.idlebusy</i> .

Figure 1 shows the relationship of these data structures.

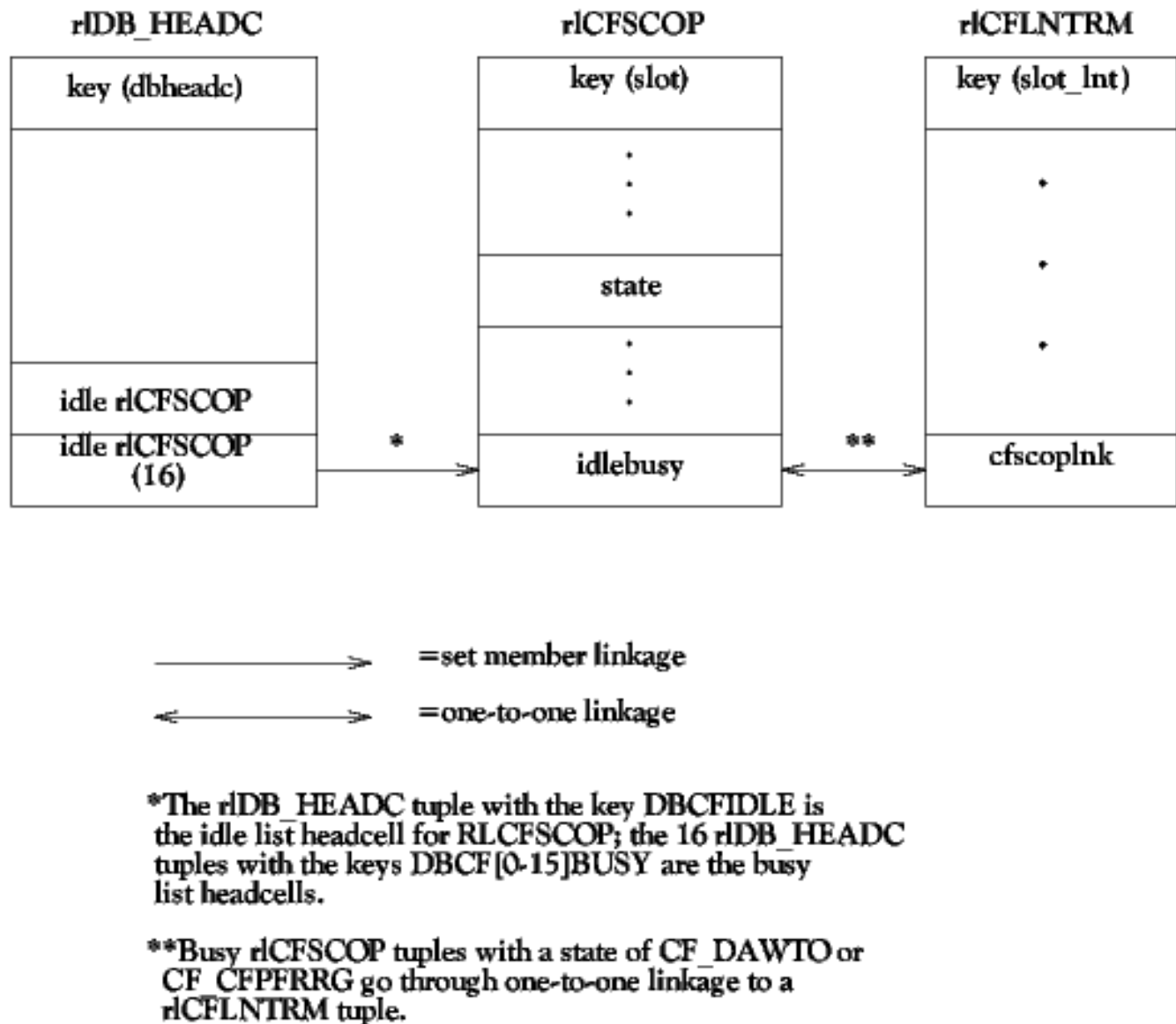


Figure 1 (CFSCOP-1) Data Structures of the CFSCOP Audit

3. AUDIT CHECKS

The CFSCOP audit uses the DRINASEK audit driver. For each rICFSCOP tuple found in the relation, the audit makes the following checks:

- NOTBUSY = When the state of the rICFSCOP tuple is other than inactive, the tuple must be on the correct busy list.
- NOTIDLE = The owning linkage of the rICFSCOP tuple must be attached to the idle headcell in RLDB_HEADC, and the link to rICFLNTRM must be null when the state of the rICFSCOP tuple is CFINACTIVE.

If the rICFSCOP tuple is busy:

- CR_DATA = This rICFSCOP tuple should have a valid rICRECORD tuple associated with it.
- PID = When the originating process ID is on the current SM, the process should be alive.
- UNEQUAL = The rICFSCOP tuple should be linked to an rICFLNTRM tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CR_DATA

Error Code:	CR_DATA
Error Description:	This rICFSCOP tuple does not have a valid rICRECORD tuple associated with it, where rICFSCOP.bcid is equal to rICRECORD.cnum.
Possible Error Effect:	The customer may not be correctly billed for the call.
Corrective Action Taken:	The rICFSCOP tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICFSCOP tuple where the error was found.
<i>Bad Data =</i>	The key of the rICRECORD tuple (rICRECORD.cnum).
<i>Good Data =</i>	The index to the rICRECORD tuple (rICFSCOP.bcid.crindx).
<i>Logical Index =</i>	The key to the rICFSCOP tuple (rICFSCOP.slot).
<i>Dump 1 =</i>	The rICFSCOP tuple where the error was found.
<i>Dump 2 =</i>	The rICRECORD tuple where the error was found.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The state of the rICFSCOP tuple is other than inactive, but the rICFSCOP tuple is not on the correct busy list.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rICFSCOP tuples.
Corrective Action Taken:	The rICFSCOP tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICFSCOP tuple where the error was found.
<i>Bad Data =</i>	The busy list the rICFSCOP tuple should be on (hc_key).
<i>Good Data =</i>	The inaccurate busy list (rICFSCOP.idlebusy.st_ownk - DBSFDB_HEA).
<i>Logical Index =</i>	The key to the rICFSCOP tuple (rICFSCOP.slot).
<i>Dump =</i>	The rICFSCOP tuple where the error was found.

Table 3 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The state of the rICFSCOP tuple is <i>CFINACTIVE</i> , but the owning linkage of the rICFSCOP tuple is either not attached to the idle headcell in RLDB_HEADC, or the link to rICFLNTRM is not null.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rICFSCOP tuples.
Corrective Action Taken:	The rICFSCOP tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICFSCOP tuple where the error was found.
<i>Bad Data =</i>	The linkage to the rICFLNTRM tuple (rICFSCOP.idlebusy.st_link).
<i>Good Data =</i>	The linkage to the rIDB_HEADC tuple (rICFSCOP.idlebusy.st_ownk -

	DBSFDB HEA).
<i>Logical Index</i> =	The key to the rICFSCOP tuple (rICFSCOP.slot).
<i>Dump</i> =	The rICFSCOP tuple where the error was found.

Table 4 Error Code: PID

Error Code:	PID
Error Description:	The rICFSCOP tuple is linked to a busy list and the originating process ID (or process ID of a previous forwarder) is on the current SM, but the process is no longer alive.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rICFSCOP tuples.
Corrective Action Taken:	The rICFSCOP tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICFSCOP tuple where the error was found.
<i>Bad Data</i> =	The process ID whose process no longer exists. (rICFSCOP.bpid.procno << 16 rICFSCOP.bpid.pcrd << 8 rICFSCOP.bpid.uniq).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rICFSCOP tuple (rICFSCOP.slot).
<i>Dump</i> =	The rICFSCOP tuple where the error was found.

Table 5 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The state of the rICFSCOP tuple is waiting for CFDA timeout or CFPF re-routing (rICFSCOP.state = CF_DAWTO or CF_CFPFRRG), but the tuple is not linked to an rICFLNTRM tuple.
Possible Error Effect:	This will cause call failures when the process later looks for a rICFLNTRM tuple.
Corrective Action Taken:	The rICFSCOP tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICFSCOP tuple where the error was found.
<i>Bad Data</i> =	The key of the rICFLNTRM tuple (rICFLNTRM.slot Int).
<i>Good Data</i> =	The state of the rICFSCOP tuple (rICFSCOP.state).
<i>Logical Index</i> =	The key to the rICFSCOP tuple (rICFSCOP.slot).
<i>Dump 1</i> =	The rICFSCOP tuple where the error was found.
<i>Dump 2</i> =	The rICFLNTRM tuple where the error was found.

CGACKT

Software Release: 5E10 and later

1. CARRIER GROUP ALARM CIRCUIT (CGACKT) AUDIT

The CGACKT audit is responsible for ensuring the integrity of tuples in the carrier group alarm circuit (RLCGACKT) relation. This relation contains carrier group alarm (CGA) status for circuit based facilities capable of having CGAs. There is a tuple in this relation for every CGA source in the CGA status (RLCGA_STAT) relation that is circuit based.

2. DATA STRUCTURES

The RLCGACKT relation is a dynamic relation that is indexed compacted and is keyed by the **ckt** field. Tuples in the RLCGACKT relation can exist in the switching module (SM) environment. The audit can run both the SM or an SM-2000. The audit can run in segmented, elevated, and directed modes. This audit resides in the BASE package.

When the audit finds a CGA error, it will place the corresponding representative rICGACKT tuple on the TMCGA_SERVICE queue for processing by the terminal maintenance (TM) CGA system process. For non digital networking unit-synchronous optical unit (SONET) (DNU_S) facilities, each facility is its own representative. For DNU_S facilities, there is a hierarchy of facilities.

The highest level is the SONET termination equipment (STE) followed by the synchronous transport signal (STS-1). The child circuit of the STS-1 is the virtual tributary 1.5 (VT1.5) which has the digital signal one (DS1) as its child circuit. Each STE may have one STS-1 underneath it.

Under the STS1, there can be up to twenty eight VT1.5s with each VT1.5 having only one DS1 underneath it. STE and VT1.5 facilities may be representative circuits that can be placed on the TMCGA_SERVICE queue. The STS-1 and DS1 are considered non-representative circuits for CGA processing. If there is no operational VT1.5 on a master virtual peripheral interface (VPIDB), the STE is considered the representative for its STS-1.

If there is an operational VT1.5 on a master VPIDB, it is considered the representative facility for the STE, STS-1, and DS1 on that VPIDB. For facilities that are not on a master VPIDB, only a VT1.5 can be a representative of its associated DS1. DNU_S facilities only exist on a SM2000.

If the CGACKT audit finds an error on a non-representative circuit, it will recover the non-representative circuit to an idle state and mark the representative circuit for recovery. If the representative circuit has yet to be audited in this pass of the CGACKT audit, it will be recovered immediately and placed on the TMCGA_SERVICE queue. If the audit has already audited the representative circuit before finding an error in the non-representative circuit, one of two possibilities will occur.

Either the representative circuit will be recovered on the second pass of the CGACKT audit, or, if the representative circuit is already on the TMCGA_SERVICE or TMCGA_INPROG queues, the TM CGA system process will be sensitive to the fact that the circuit is marked for audit recovery and treat the processing of this circuit like it was a request for audit recovery. This will mean that the TM CGA System Process will read the F bits for this facility to determine the proper CGA state. Facilities that are their own representative, will be recovered immediately and placed on the TMCGA_SERVICE queue for processing.

If the CGACKT audit finds an error outside of a selective initialization of the SM, it will send an OSWAKEUP message to the TM CGA system process to process any CGA requests related to an audit error.

3. AUDIT CHECKS

The CGACKT audit uses the DRINDEX driver to step through the RLCGACKT relation and perform the semantic checks on the rICGACKT tuples. For each tuple, the CGACKT audit will make the following checks: If the CGACKT audit is entered:

EXTRATUP	= Check to make sure that the rICGACKT tuple has a corresponding rICGA_STAT tuple.
INHNORM	= Check to make sure that the noau field in the corresponding rICGACKT tuple is DBNO for non-operational facilities.
LOST_CKTDATA	= Check to make sure that there is a valid VT1.5 circuit name associated with a DS1 facility given the fact that the DS1 facility is populated in the RLCGACKT relation.
RPTRBAD	= Check to make sure that the audit can obtain a pointer to the representative rICGACKT tuple.
SETMAP	= Check to make sure the src_map field in corresponding rICGACKT tuple matches the CGACKT audit calculated representative circuit name.
TYPE	= Check to make sure the src_type field in the rICGACKT tuple matches the type specified in the corresponding rICGA_STAT tuple.

The following checks are done if the representative rICGACKT tuple is not marked for audit recovery.

CGAQY_BITN	= Check to make sure that the noau field in the rICGACKT tuple is set to DBYES if its representative rICGACKT tuple is on either the TMCga_SERVICE or TMCga_INPROG queues.
CGAQN_BITY	= Check to make sure that the noau field in the rICGACKT tuple is set to DBNO if its representative rICGACKT tuple is not on either the TMCga_SERVICE or TMCga_INPROG queues.
STUCKBUSY	= Check to make sure the noau field in the rICGACKT tuple has not been left set to DBYES for more than 5 minutes.

The following check is made only if the representative rICGACKT tuple is marked for audit recovery.

NOSPECIALBIT	= Check to make sure the noau field in the rICGACKT tuple is set to DBYES if the representative rICGACKT tuple is marked for audit recovery.

The following check is made only if the facility represented by this rICGACKT indicates that it is operational (that is, its **src_map** is not GLNULCKT).

ODDLOSTTUP	= Check to make sure that there is a corresponding static rICGAREL tuple associated with the DNU_S facility represented by the rICGACKT tuple under audit.

The following check is made only if the facility represented by this rICGACKAT tuple indicates that it is operational and that its representative rICGACKT tuple is not marked for audit recovery.

READSMEST	= Check to make sure the audit can obtain an associated rISMEST tuple for both the circuit represented by the rICGACKT under audit and its representative circuit.

The following checks are made only if the circuit represented by the rICGACKT tuple under audit is not operational (that is, in GROW) or its associated rISMEST tuple or its representative circuits rISMEST does not indicate a busy condition.

ALARM	= Check to make sure that the TM queue to which this rICGACKT tuple is linked to corresponds

to the CGA alarm status in the associated rISMEST tuple.

WRONGLIST = Check to make sure the this rICGACKT tuple is linked to a valid TM queue.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALARM

Error Code:	ALARM
Error Description:	The CGACKT audit has detected an rICGACKT tuple that is linked to an improper TM CGA queue based on its CGA alarm status in the associated rISMEST tuple.
Possible Error Effect:	The CGA status for this facility represented by this rICGACKT is incorrect. Therefore, facility may be marked out of service with a CGA when it is really in service or visa versa.
Corrective Action Taken:	The rICGACKT tuple recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCGA_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The physical key of the rICGALIST for which this tuple has member linkage or 0.
<i>Good Data =</i>	Valid values are: 0 = If the rICGACKT tuple is linked to the TMCGA_ALARM list but its status in rISMEST indicates no alarms. 1 = If the rICGACKT tuple is not linked to any list but has CGA alarm status in its rISMEST.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple.
<i>Dump 2 =</i>	The corresponding rISMEST tuple.

Table 2 Error Code: CGAQN_BITY

Error Code:	CGAQN_BITY
Error Description:	The CGACKT audit has found an rICGACKT tuple whose noau field is set to DBYES when the representative rICGACKT tuple is not linked to TMCGA_SERVICE or TMCGA_INPROG queues.
Possible Error Effect:	CGA status handling will not be correct for the DNU_S facility under audit.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCGA_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The corrupted value of the noau field in the rICGACKT tuple.
<i>Good Data =</i>	The correct value of the noau field in the rICGACKT tuple.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The rICGACKT tuple of the representative facility.

Table 3 Error Code: CGAQY_BITN

Error Code:	CGAQY_BITN
Error Description:	The CGACKT audit has found an rICGACKT tuple whose noau field is set to

	DBNO when the representative rICGACKT tuple is linked to TMCGA_SERVICE or TMCGA_INPROG queues.
Possible Error Effect:	The audit may audit rICGACKT tuples in a busy state. This may lead to incorrectly reporting audit errors for the facility in question.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCGA_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The corrupted value of the noau field in the rICGACKT tuple.
<i>Good Data =</i>	The correct value of the noau field in the rICGACKT tuple.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The rICGACKT tuple of the representative facility.

Table 4 Error Code: INHNORM

Error Code:	INHNORM
Error Description:	The CGACKT audit has found an rICGACKT tuple whose noau field is set to DBYES when this rICGACKT has no representative rICGACKT (that is, its src_map is GLNULCKT).
Possible Error Effect:	CGA processing may not be functioning properly for this facility.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The corrupted value of the noau field in the rICGACKT tuple.
<i>Good Data =</i>	The correct value of the noau field in the rICGACKT tuple.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The corresponding rICGA_STAT tuple.

Table 5 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The CGACKT audit could not obtain a valid DNU_S STE or VT1.5 circuit name for the DNU_S facility under audit.
Possible Error Effect:	CGA status handling will not be correct for the DNU_S facility under audit.
Corrective Action Taken:	None.
Manual Action Required:	These names are derived by calling macros that try to read the corresponding rICKTDATA tuple for the STE or VT1.5. Thus, there is a missing rICKTDATA tuple for this VT1.5 or STE DNU_S facility. The good data field in the dump indicates if the STE or VT1.5 rICKTDATA tuple is missing. It will also indicate where in the CGACKT audit code the error occurred. This missing tuple must be inserted into the database.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The value of GLNULCKT.
<i>Good Data =</i>	0 = If the missing rICKTDATA tuple is for the STE circuit. 1 = If the missing rICKTDATA tuple is for the VT1.5 circuit. 2 = If the missing rICKTDATA tuple is for the VT1.5 circuit 3 = If the missing rICKTDATA tuple is for the VT1.5 circuit.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple under audit.
<i>Dump 2 =</i>	The corresponding rICGA_STAT tuple.

Table 6 Error Code: NOSPECIALBIT

Error Code:	NOSPECIALBIT
Error Description:	The CGACKT audit has found an rICGACKT tuple whose noau field is set to DBNO when the representative rICGACKT tuple is marked for audit recovery.
Possible Error Effect:	CGA status handling will not be correct for the facility under audit.
Corrective Action Taken:	The rICGACKT tuple under audit will have its noau bit set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The corrupted value of the noau field in the rICGACKT tuple.
<i>Good Data =</i>	The correct value of the noau field in the rICGACKT tuple.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The rICGACKT tuple of the representative facility.

Table 7 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The CGACKT audit has found an rICGACKT tuple that represents a DNU_S facility. However, it could not obtain the corresponding rICGAREL tuple for this facility.
Possible Error Effect:	CGA status handling will not be correct for the facility under audit.
Corrective Action Taken:	None.
Manual Action Required:	This missing rICGAREL tuple must be inserted into the database. The key of the missing rICGAREL tuple is given in the <i>Bad Data</i> field of the audit dump.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The logical key of the missing rICGAREL tuple.
<i>Good Data =</i>	The relation ID of rICGAREL.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The corresponding rICGA_STAT tuple.

Table 8 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The CGACKT audit was unable to read the corresponding rISMEST tuple for either the facility under audit or its representative facility.
Possible Error Effect:	CGA status handling will not be correct for the facility under audit.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCGA_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	0 = If the audit could not read rISMEST for this facility. 1 = If the audit could not read rISMEST for the representative facility.
<i>Good Data =</i>	The logical key of the rICGACKT tuple.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The rICGACKT tuple of the representative facility.

Table 9 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	The CGACKT audit was unable to read the corresponding rICGACKT tuple for the representative facility of the facility under audit.
Possible Error Effect:	CGA status handling will not be correct for the facility under audit.
Corrective Action Taken:	None. The key of the representative rICGACKT tuple is probably bad. Therefore, it should be corrected when the CGACKT audit audits the representative

	rICGACKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rICGACKT tuple.
<i>Bad Data</i> =	The logical key of the representative rICGACKT tuple.
<i>Good Data</i> =	The value of the src_map field in the rICGACKT tuple.
<i>Logical Index</i> =	The logical key of the rICGACKT tuple.
<i>Dump</i> =	The rICGACKT tuple currently under audit.

Table 10 Error Code: SETMAP

Error Code:	SETMAP
Error Description:	The CGACKT audit has found that the src_map field in the rICGACKT tuple under audit is corrupt.
Possible Error Effect:	CGA status handling will not be correct for the facility under audit.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCga_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rICGACKT tuple.
<i>Bad Data</i> =	The value of the src_type field in the rICGACKT tuple.
<i>Good Data</i> =	The value of the correct src_map field in the rICGACKT tuple.
<i>Logical Index</i> =	The logical key of the rICGACKT tuple.
<i>Dump 1</i> =	The rICGACKT tuple currently under audit.
<i>Dump 2</i> =	The corresponding rICGA_STAT tuple.

Table 11 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The CGACKT audit has detected an rICGACKT tuple that has its noau bit set to "DBYES" for more than 5 minutes. This bit is used to keep audits from auditing the CGA status represented by the information in the rICGACKT tuple while the TM CGA system process is working on this tuple.
Possible Error Effect:	The rICGACKT tuple is stuck in a busy state. This may be due to the fact that the TM CGA system process is not active and needs to be sent a WAKEUP message.
Corrective Action Taken:	The rICGACKT tuple recovered to an idle state, and the TM CGA system process is sent a WAKEUP message.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rICGACKT tuple.
<i>Bad Data</i> =	The value of the noau field.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The logical key of the rICGACKT tuple.
<i>Dump 1</i> =	The rICGACKT tuple.
<i>Dump 2</i> =	The corresponding rICGA_STAT tuple.

Table 12 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit has detected a wrong src_type in an rICGACKT tuple.
Possible Error Effect:	The CGA processing for this port type facility will not function properly.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCga_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICGACKT tuple.
<i>Bad Data</i> =	The value of the corrupted src_type field.
<i>Good Data</i> =	The value of the type field in the corresponding rICGA_STAT tuple.
<i>Logical Index</i> =	The logical key of the rICGACKT tuple.
<i>Dump 1</i> =	The rICGACKT tuple.
<i>Dump 2</i> =	The corresponding rICGA_STAT tuple.

Table 13 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The CGACKT audit has found an rICGACKT tuple with an unknown or incorrect member linkage to the RLCGALIST relation.
Possible Error Effect:	CGA status handling will not be correct for the facility under audit.
Corrective Action Taken:	The rICGACKT tuple under audit is recovered to an idle state and its representative rICGACKT tuple will be placed on the TMCGA_SERVICE queue for CGA processing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGACKT tuple.
<i>Bad Data =</i>	The value of the list_lnk.st_ownd linkage in the rICGACKT tuple.
<i>Good Data =</i>	The logical key of the corresponding rSMEST tuple.
<i>Logical Index =</i>	The logical key of the rICGACKT tuple.
<i>Dump 1 =</i>	The rICGACKT tuple currently under audit.
<i>Dump 2 =</i>	The corresponding rSMEST tuple.

CGALIST

Software Release: 5E10 and later

1. CARRIER GROUP ALARM LIST (CGALIST) AUDIT

The terminal maintenance (TM) subsystem must know the status of facilities that can have carrier group alarms (CGA). To keep track of CGA transitions, special tuples called head cells are used. A head cell indicates the first and last facility on a CGA list. Facilities that currently have no CGAs are not linked to any head cell. The CGA list relation (RLCGALIST) consists of facility list head cells. Three list head cells are used. The TMCGA_SERVICE head cell contains a list of facilities that are waiting to have CGAs serviced by the TM subsystem. The TMCGA_INPROG head cell contains a list of facilities that are currently having their CGA status updated by the TM subsystem. Finally, TMCGA_ALARM contains a list of facilities that currently have stable CGAs. The members that are linked to each list in the RLCGALIST relation may be of two types. Digital facilities that may have CGAs are linked to the various lists using carrier group alarm circuit relation (RLCGACKT) tuples. Facilities that represent metallic service unit (MSU) scan points are linked to the various lists using carrier group alarm port relation (RLCGAPORT) tuples. The linkage between the RLCGAPORT and RLCGALIST relations is not used.

The CGALIST audit verifies the linkage between the head cells in the RLCGALIST relation and the first and last facilities represented by rICGACKT tuples in that list. The head tuple in rICGALIST links to the first facility on the list, which links to other facilities with the same CGA status. The last facility on the list links to the tail tuple in rICGALIST and completes the chain. If this chain is broken, Terminal Maintenance would not be able to determine the CGA status of the facilities.

This audit is part of the carrier group alarm audits subgroup, and resides in the BASE package. The audit can run in segmented, elevated, or demand modes. It can run in both the switching module (SM) or the SM-2000.

2. DATA STRUCTURES

The data structures checked by the CGALIST audit are the carrier group alarm list relation (RLCGALIST) and the carrier group alarm circuit relation (RLCGACKT). The rICGALIST tuple has owner linkages with rICGACKT tuples.

3. AUDIT CHECKS

To verify the linkages in the RLCGALIST relation, the audit calls the linkage driver (DRLINK) once for each audit segment. The DRLINK driver verifies that each head cell in rICGALIST has a valid owner-member linkage to an rICGACKT tuple. Since the RLCGALIST head cells contain only linkage, no semantic checks are performed.

4. ERROR HANDLING

If one of the head cells contains erroneous data, it will be zeroed and the list reestablished when the carrier group alarm circuit (CGACKT) audit runs.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

CGAPORT

Software Release: 5E10 and later

1. CARRIER GROUP ALARM PORT (CGAPORT) AUDIT

The CGAPORT audit is responsible for ensuring the integrity of tuples in the carrier group alarm port (RLCGAPORT) relation. This relation contains carrier group alarm (CGA) status for port based metallic service unit (MSU) scan points. There is a tuple in this relation for every CGA source in the CGA status (RLCGA_STAT) relation that is port based.

2. DATA STRUCTURES

The RLCGAPORT relation is a dynamic relation that contains information related to the CGA status of port based facilities. The RLCGAPORT relation is an indexed compaction access relation that is keyed by the **port** field. Tuples in the RLCGAPORT relation can exist in the switching module (SM) environment. The audit can run both the SM or the SM-2000. The audit can run in segmented or elevated modes. This audit resides in the BASE package.

If the audit finds an error outside of a selective initialization of an SM, it will send an OSWAKEUP message to the terminal maintenance (TM) CGA system process to process any CGA requests related to an audit error.

3. AUDIT CHECKS

The CGAPORT audit uses the DRINDEX driver to step through the RLCGAPORT relation and perform the semantic checks on the rICGAPORT tuples. For each tuple, the CGAPORT audit will make the following checks:

EXTRATUP	= Check to make sure that the rICGAPORT tuple has a corresponding rICGA_STAT tuple.
STUCKBUSY	= Check to see if the noau field in the rICGAPORT tuple has been set to DBYES for more than 5 minutes.
TYPE	= Check to make sure the src_type field in the rICGAPORT tuple matches the type specified in the corresponding rICGA_STAT tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The CGAPORT audit has detected an rICGAPORT tuple that has its noau bit set to "DBYES" for more than 5 minutes. This bit is used to keep audits from auditing the CGA status represented by the information in the rICGAPORT tuple while the TM CGA system process is working on this tuple.
Possible Error Effect:	The rICGAPORT tuple is stuck in a busy state. This maybe due to the fact that the TM CGA system process is not active and needs to be sent a WAKEUP message.
Corrective Action Taken:	The rICGAPORT tuple recovered to an idle state, and the TM CGA system process is sent a WAKEUP message.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the corresponding rICGAPORT tuple.
<i>Bad Data =</i>	The value of the noau field.

<i>Good Data</i> =	0.
<i>Logical Index</i> =	The logical key of the rICGAPORT tuple.
<i>Dump 1</i> =	The rICGAPORT tuple.
<i>Dump 2</i> =	The corresponding rICGA_STAT tuple.

Table 2 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit has detected a wrong src_type in an rICGAPORT tuple.
Possible Error Effect:	The CGA processing for this port type facility will not function properly.
Corrective Action Taken:	The src_type field is corrected using the value specified in the corresponding rICGA_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICGAPORT tuple.
<i>Bad Data</i> =	The value of the corrupted src_type field.
<i>Good Data</i> =	The value of the type field in the corresponding rICGA_STAT tuple.
<i>Logical Index</i> =	The logical key of the rICGAPORT tuple.
<i>Dump 1</i> =	The rICGAPORT tuple.
<i>Dump 2</i> =	The corresponding rICGA_STAT tuple.

CGASTAT

Software Release: 5E10 and later

1. CARRIER GROUP ALARM STATUS (CGASTAT) AUDIT

The carrier group alarm status (RLCGA_STAT) relation contains information that indicates if a carrier group alarm (CGA) is to be reported, ignored, or conditioned for a given facility. It also specifies the type of facility that had the CGA. There is one rICGA_STAT tuple for each port or circuit based facility that is capable of having CGA status.

2. DATA STRUCTURES

The RLCGA_STAT static relation contains information that specifies which facilities can have CGA status. If a facility is port based, the *spn* field in the corresponding rICGA_STAT tuple will be set to DBYES. In this case, there will be a corresponding dynamic carrier group alarm port (rICGAPORT) tuple for this facility. If the *spn* field indicates DBNO in the rICGA_STAT tuple, this facility is circuit based and will have a corresponding dynamic carrier group alarm circuit (rICGACKT) tuple. The RLCGA_STAT is an indexed compaction access relation that is keyed by the *circuit* field which represents the facility.

The purpose of the CGASTAT audit is to insure that there is a one for one correspondence between the static rICGA_STAT tuples and the dynamic rICGACKT and rICGAPORT tuples in the switching module (SM). This audit can run in segmented or elevated modes. The CGASTAT audit resides in the BASE software package. It can run in both the SM and the SM-2000. On an audit error that occurs outside of an SM selective initialization, the CGASTAT audit will send an OSWAKEUP message to the terminal maintenance (TM) CGA system process to wake up the process so that it may process related CGAs. Only circuit based facilities that have rICGACKT tuples are placed on the TM CGA servicing queue.

3. AUDIT CHECKS

The CGASTAT audit uses the STATIC audit driver to traverse through each tuple in the RLCGA_STAT relation. For each tuple, the CGASTAT audit will make the following checks:

LOSTTUP = Check to make sure that the rICGA_STAT tuple has a corresponding rICGACKT or rICGAPORT tuple depending on the *spn* field in the rICGA_STAT tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Depending on the value of <i>spn</i> field in rICGA_STAT tuple, the audit did not find a corresponding dynamic rICGAPORT or rICGACKT tuple.
Possible Error Effect:	CGA processing will not function properly for this facility.
Corrective Action Taken:	The audit will try to insert the missing dynamic rICGAPORT or rICGACKT tuple. In the case of circuit based facilities, the rICGACKT tuple will be placed on the TMCGA_SERVICE queue in the carrier group alarm list (RLCGALIST) relation so the TM CGA system process may process its CGA status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICGA_STAT tuple.
<i>Bad Data</i> =	The circuit name in the buffer to read the dynamic relation.
<i>Good Data</i> =	The relation ID of the missing dynamic tuple (either RLCGAPORT or

	RLCGACKT).
<i>Logical Index</i> =	The logical key of the rICGA_STAT tuple.
<i>Dump</i> =	The rICGA_STAT tuple.

CHALLOC

Software Release: 5E10 and later

1. CHANNEL ALLOCATION (CHALLOC) AUDIT

The channel allocation map contains information about which peripheral side time slots (channels) are busy. The channel allocation audit (CHALLOC) verifies the busy/idle status of the channel allocation map against the status of the associated channel data blocks. Recovery prevents resources from becoming lost to the system. This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structures checked by the CHALLOC audit are the channel data block relation (RLCHDB), the circuit data block common area relation (RLCDBCOM), and the channel allocation array, (pcCHALLOC). Figure 1 illustrates the layout of the channel allocation array, and Figure 2 illustrates the pcCHALLOC linkages. The audit also checks the peripheral interface data bus (PIDB) time slot block relation (RLPTSB) and the ISLU idle time slot summary (PCdichidle) array.

pcCHALLOC

Busy_idle bits
Count of even time slots that are idle (evencnt) (per group of 8)
Count of odd time slots that are idle (oddcnt) (per group of 8)
Count of even time slots that are in use be ISLU (islu_use)
Count of odd time slots that are in use be ISLU (islu_use)

Figure 1 (CHALLOC-1) Channel Allocation Array

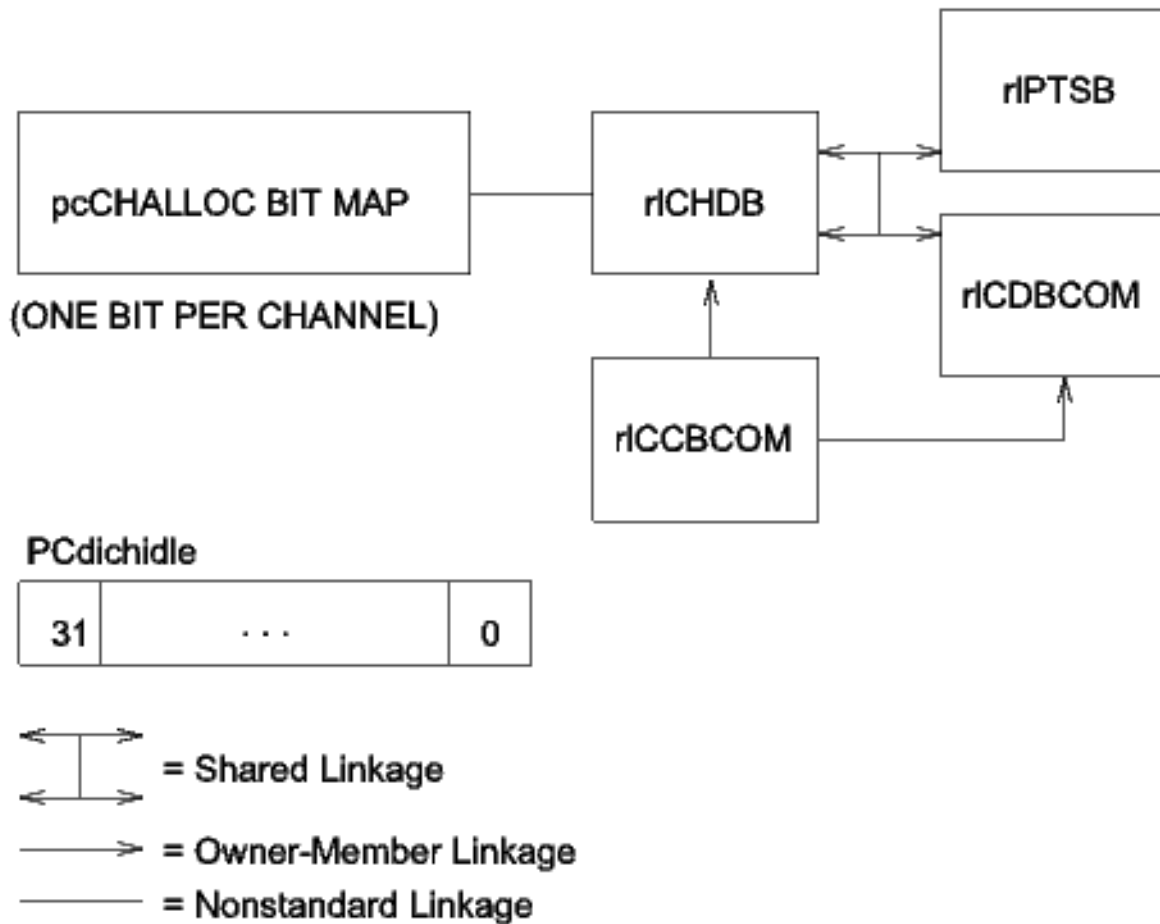


Figure 2 (CHALLOC-2) pcCHALLOC Linkages

3. AUDIT CHECKS

The CHALLOC audit steps through the pcCHALLOC array checking the busy/idle status of the channel allocation against the status of the associated channel data block. The audit checks the busy/idle bit for a channel against the channel data block linkages to the channel control block common area and the circuit data block common area. Channel data blocks, except for RSM control time slots in a host switching module (HSM), that are linked to the channel data block idle list and are not linked to a circuit data block common area (RLCDBCOM), should be marked idle in pcCHALLOC. All other channel data blocks should be marked busy in pcCHALLOC. The audit also checks the even count (evencnt) and odd count (oddcnt) fields to ensure that they agree with the number of busy/idle bits marked idle in each group of eight even-and odd-numbered channels.

The global variables for indicating the time slots designated for audible ringing (Glaudring) and sourcing idle code (Glidlech) are checked to ensure that they are marked busy in the pcCHALLOC array.

The even and odd counts for time slots available for ISLU usage (islu_use) are checked by verifying the shared linkage between rCHDB and rCDBCOM or rIPTSB. If the rCHDB tuple is linked to rIPTSB, the time slot is in use by ISLU. Also, PCdichidle[] is checked for consistency after each group of eight time slots is audited.

4. ERROR HANDLING

If an inconsistency exists between pcCHALLOC and RLCHDB, time slots (except RSM control time slots) in pcCHALLOC are changed to agree with RLCHDB. The RSM control time slots in an HSM must stay busy in the

channel allocation map.

Recovery includes the updating of all appropriate count fields.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	1 = A busy rICHDB tuple does not have its associated bit marked correctly in the channel allocation maps. A map bit value of 0 indicates busy while a map bit value of 1 indicates idle. 2 = The audible ringing or idle code channel is not marked busy.
Possible Error Effect:	The busy rICHDB could be seized by another process.
Corrective Action Taken:	The audit will busy the time slot in the channel allocation map.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of rICHDB tuple.
<i>Bad Data</i> =	Map contents (bit 1 = istatus, bit 0 = ostatus).
<i>Good Data</i> =	0 (istatus = idle, ostatus = idle).
<i>Logical Index</i> =	Logical key of rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 2 Error Code: EVEN_COUNT

Error Code:	EVEN_COUNT
Error Description:	The count of the number of busy channels in the channel allocation map is in error.
Possible Error Effect:	There may be more or less channels available.
Corrective Action Taken:	The audit corrects the busy channel count in the channel allocation map.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of the counter.
<i>Bad Data</i> =	Bad count value.
<i>Good Data</i> =	Correct count value.
<i>Logical Index</i> =	Channel allocation row index.
<i>Dump 1</i> =	Channel allocation incoming map row.
<i>Dump 2</i> =	Channel allocation outgoing map row.

Table 3 Error Code: IDL_STATE

Error Code:	IDL_STATE
Error Description:	The contents of PCdichidle[] is not correct.
Possible Error Effect:	There may be more or less channels available.
Corrective Action Taken:	The PCdichidle[] will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCdichidle[].
<i>Bad Data</i> =	The contents of PCdichidle[].
<i>Good Data</i> =	The ISLU allocation count.
<i>Logical Index</i> =	Channel allocation row index.
<i>Dump</i> =	NA.

Table 4 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	An rICHDB tuple that is idle does not have its associated bit marked correctly in the channel allocation maps.

Possible Error Effect:	The channel allocation maps are out of sync.
Corrective Action Taken:	The audit idles the bit in the channel allocation map.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address =</i>	Address of rICHDB tuple.
<i>Bad Data =</i>	Linkage contents (bit 1 = istatus, bit 0 = ostatus).
<i>Good Data =</i>	3 (istatus = 1, ostatus = 1).
<i>Logical Index =</i>	Logical key of rICHDB tuple.
<i>Dump =</i>	rICHDB tuple.

Table 5 Error Code: ISEUSG_CNT

Error Code:	ISEUSG_CNT
Error Description:	The count of the number of ISLU busy channels in the channel allocation map is in error.
Possible Error Effect:	There may be more or less channels available.
Corrective Action Taken:	The audit corrects the ISLU busy channel count in the channel allocation map.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address =</i>	Address of the counter.
<i>Bad Data =</i>	Bad count value.
<i>Good Data =</i>	Correct count value.
<i>Logical Index =</i>	Channel allocation row index.
<i>Dump 1 =</i>	Channel allocation incoming map row.
<i>Dump 2 =</i>	Channel allocation outgoing map row.

Table 6 Error Code: ISOUSG_CNT

Error Code:	ISOUSG_CNT
Error Description:	The count of the number of ISLU busy channels in the channel allocation map is in error.
Possible Error Effect:	There may be more or less channels available.
Corrective Action Taken:	The audit corrects the ISLU busy channel count in the channel allocation map.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address =</i>	Address of the counter.
<i>Bad Data =</i>	Bad count value.
<i>Good Data =</i>	Correct count value.
<i>Logical Index =</i>	Channel allocation row index.
<i>Dump 1 =</i>	Channel allocation incoming map row.
<i>Dump 2 =</i>	Channel allocation outgoing map row.

Table 7 Error Code: ODD_COUNT

Error Code:	ODD_COUNT
Error Description:	The count of the number of busy channels in the channel allocation map is in error.
Possible Error Effect:	There may be more or less channels available.
Corrective Action Taken:	The audit corrects the busy channel count in the channel allocation map.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address =</i>	Address of the counter.
<i>Bad Data =</i>	Bad count value.
<i>Good Data =</i>	Correct count value.
<i>Logical Index =</i>	Channel allocation row index.
<i>Dump 1 =</i>	Channel allocation incoming map row.
<i>Dump 2 =</i>	Channel allocation outgoing map row.

Table 8 Error Code: RSMCHAN

Error Code:	RSMCHAN
Error Description:	Invalid DB return when accessing the RLRSCHAN relation; possible ODD problem.
Possible Error Effect:	Relation can not be accessed.
Corrective Action Taken:	None.
Manual Action Required:	If error repeats investigate the ODD data.

Dump Description:	
<i>Error Address</i> =	Address of rICHDB tuple.
<i>Bad Data</i> =	Bad DB return value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key used to access the RLRSMCHAN relation.
<i>Dump</i> =	rLRSMCHAN buffer.

Table 9 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets, and or ASSERTS.
Corrective Action Taken:	Zero the rICHDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Channel number.
<i>Dump</i> =	NA.

CHAN

Software Release: 5E10 and later

1. LINE UNIT CHANNEL CIRCUIT (CHAN) AUDIT

Channels are time slots that carry data and signaling information to the peripheral side of the time slot interchanger (TSI) and signal processor. The channel circuit relation (RLCHAN) contains dynamic information about all line unit channel circuits used by call processing. An rCHAN tuple is updated by peripheral control when a call is being processed. Each channel circuit has a one-to-one correspondence with a B-link in the line unit concentrator path. The CHAN audit only performs key checks, circuit name checks, and dynamic-to-static existence checks against RLCKTDATA in order to find extra rCHAN tuples.

If extra tuples collect in rCHAN, hardware growth is inhibited; therefore, the CHAN audit is needed to identify inconsistencies, delete extra tuples, and correct bad keys.

This audit is part of the peripheral control audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The CHAN audit uses data in the static circuit data relation (RLCKTDATA) to verify data in the dynamic channel circuit relation (RLCHAN). The rCHAN relation contains information such as the key (made up of the line unit number, service group number, and channel circuit number), circuit name, and the busy/idle bit. Figure 1 shows the fields in rCHAN checked by the audit. The CHAN audit only performs a key check and circuit name check against RLCKTDATA and performs a dynamic-to-static existence check to detect extra rCHAN tuples. The LINE audit verifies the rCHAN, rICDBCOM, and B-link status map consistency.

rCHAN

CHANNEL CIRCUIT (chanckt)
CIRCUIT NAME (name)
.
.
.
BUSY/IDLE STATUS (stg0gd)

rCKTDATA

CIRCUIT NAME (name)
EQUIPAGE STATE (eqstat)

_____ = Nonstandard linkage

Figure 1 (CHAN-1) Data Structures

3. AUDIT CHECKS

The channel audit uses the static driver (DRSTATIC) to step through the RLCHAN relation.

- (a) Initially, the audit calls a dynamic-to-static existence check function, which includes a key check. A key check is performed by any audit with a double-indexed relation that invokes the DRSTATIC driver. It verifies that the key passed to the existence check function is correct. If the key is correct, the audit performs the following existence checks.
 - (1) For each tuple in rCHAN, it verifies that a corresponding tuple exists in the static rCKTDATA relation.
 - (2) It checks for any extra rCHAN tuples.
 - (3) It checks the equipage state of the corresponding rCKTDATA tuple. If the state is unequipped (UNEQ), no rCHAN tuple should exist for that corresponding circuit name.

4. ERROR HANDLING

If the key passed to the existence check function is in error, it is corrected according to data in RLCKTDATA.

Any tuple in RLCHAN that does not correspond to a tuple in RLCKTDATA, or has a corresponding rCKTDATA tuple in the unequipped (UNEQ) state, is deleted and an error is reported.

The LINE, CKTDATA, CHDB, CDBCOM audits are scheduled as related audits.

The CHAN audit produces the following error codes and those generated by the DRSTATIC driver.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The channel circuit name in the rICHAN tuple is not the same as the calculated name.
Possible Error Effect:	This would cause loss of call processing resources, asserts, and improper database accesses.
Corrective Action Taken:	The circuit name is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICHAN tuple.
<i>Bad Data =</i>	Actual circuit name in rICHAN.
<i>Good Data =</i>	Calculated circuit name.
<i>Logical Index =</i>	Key of rICHAN tuple.
<i>Dump =</i>	rICHAN tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	There is an extra tuple in the rICHAN relation. This tuple is not supported by static data in rCKTDATA.
Possible Error Effect:	This tuple is taking up space which could be used for growth.
Corrective Action Taken:	The extra tuple is removed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICHAN Tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rICHAN Tuple.
<i>Dump 1 =</i>	Contents of Extra rICHAN Tuple.
<i>Dump 2 =</i>	Contents of the rCKTDATA Tuple.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	The key of the rICHAN tuple is not the same as the calculated key.
Possible Error Effect:	The bad key would cause loss of call processing resources and possible asserts from PC.
Corrective Action Taken:	The key in the tuple is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICHAN tuple.
<i>Bad Data =</i>	Actual key of the rICHAN tuple.
<i>Good Data =</i>	Calculated key of rICHAN tuple.
<i>Logical Index =</i>	Calculated key of rICHAN tuple.
<i>Dump =</i>	rICHAN tuple.

CHDB-A

Software Release: 5E10 only

1. CHANNEL DATA BLOCK (CHDB) AUDIT

The channel data block (CHDB) is an abstraction of the peripheral side channel of the network. The channel data block is used to allocate channels to lines for call processing. A line is assigned a channel when the customer goes off-hook. Trunk circuits have dedicated channels. If there is an error in the channel data block data associated with an active call, it will interfere with successful call processing. An error in the channel data block data associated with an idle trunk channel may result in lost supervision.

Time available on the link from the switching module (SM) to a remote switching module (RSM) is divided into time slots. Each time slot carries a voice signal, which is composed of 16 bit pulses. Eight of the bits contain the encoding of the voice signal, and eight bits are used for signaling. One of these eight signaling bits, the E signaling bit, confirms the connection between the originating and terminating processes. The possible bit values are continuity and discontinuity.

The channel data block audit (CHDB) ensures the integrity of the channel data block relation (RLCHDB). The audit performs validity checks on the head cells in the relation and on the rCHDB tuples.

2. DATA STRUCTURES

The data structures checked by the CHDB audit are the channel data block relation (RLCHDB), the unconcentrated buffer head cell relation (RLUBHC), the channel control block common area relation (RLCCBCOM), the circuit data block common area relation (RLCDBCOM) and the PIDB time slot block relation (RLPTSB). The following data structures are used for SM-2000 processors only:

ldsfs_ts = A field in the rCHDB relation that represents the LDSF TSI time slot associated with the CHDB.
rLNK_USE = The TSI link type relation structure.

Figure 1 illustrates the layout of rCHDB and the linkages associated with it.

3. AUDIT CHECKS

To verify the linkages associated with the channel data blocks, the CHDB audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the DRLINK driver executes a semantic check function that performs the following checks based on the state of the channel.

- (a) The audit checks the channel state (inproc) field against the channel data block linkages. Idle channel data blocks should be owned by the channel data block idle list head cell or else the channel data blocks will be unavailable for call processing. An idle channel data block should have zero linkages except for those channel blocks dedicated to trunks. If an idle channel block has linkages, other resources cannot be used for call processing. An idle channel data block used for a trunk application should have a valid linkage to the trunk circuit data block. Otherwise, the trunk circuit cannot be used by call processing.
- (b) If the inproc value indicates a special idle state for a given trunk type, the audit verifies the following:
 - (1) The rCHDB tuple is properly linked to the channel data block idle list.
 - (2) The rCHDB tuple is marked busy in the channel allocation maps.
 - (3) The rCHDB tuple is properly linked to a circuit data block (trunk circuits).

- (c) If an idle rCHDB tuple is associated with an HSM T1 facility, the audit verifies that the E-bit discontinuity is sent to the RSM.
- (d) Any channel data block (except RSM control time slots in HSMs) associated with an idle channel (whether dedicated to a trunk or available for connection to a line) should be receiving from the time slot 0 (idle code) to prevent the signal of another telephone call from being transmitted to more than one destination. The audit performs a hardware read to determine which time slot is connected to the channel.
- (e) If the inproc value indicates that the channel data block is in a stable busy state, the audit verifies the following:
 - (1) The rCHDB tuple is properly linked to an rCCBCOM tuple owned by a terminal process.
 - (2) The rCHDB tuple is properly linked to an rUBHC tuple.
 - (3) The rCHDB tuple is marked busy in the channel allocation maps.
 - (4) The rCHDB tuple is properly linked to rCDBCOM tuple or rPTSB tuple.
- (f) If a stable rCHDB tuple is associated with an HSM T1 facility, the audit verifies that E-bit changes are being scanned. The audit also verifies that scanning detects continuity unless the tuple is on a hit-timing list.
- (g) Sometimes the inproc value indicates that the state of the channel is expected to change soon. This situation usually occurs during transient states such as trunk signaling, digit collection for lines and trunks, and flash and hit timing for lines and trunks. If the inproc value indicates this case, the audit verifies the following:
 - (1) The rCHDB tuple is marked busy in the channel allocation maps.
 - (2) The rCHDB tuple is not in a transient state for more than five minutes.
- (h) A few channels are used for special system purposes, such as audible ringing tone. If a transmission test facility (TTF) is used, three channels are dedicated to it. In this case, the inproc value indicates that the PC subsystem should ignore supervision of the channels, and the audit verifies the following:
 - (1) The rCHDB tuples are properly linked to a special list.
 - (2) The rCHDB tuples are marked busy in the channel allocation maps.
- (i) The audit checks the four_wire bit in the rCHDB tuple. The four_wire bit in the rCHDB should only be set if the trunk using this time slot is 4-WIRE trunk or a precedence and preemption trunk.
- (j) To save the call on hold during the selective initialization, the CHDB audit checks the path state in the PHDB. If the path is open and the inproc state is stable, then the call on hold will be saved.
- (k) For TSIU to PSU nailed up channel in the idle state, verify that the CHDB tuple is properly linked to idle list (CCBCOM), linked to the logical time slot circuit (CDBCOM) and not linked to PTSB (since an idle channel has not been connected to the time slot in the PIDB).
- (l) For TSIU to PSU nailed up channel in the stable state, verify that the CHDB tuple is properly linked to an active list (CCBCOM), linked to the logical time slot circuit (CDBCOM) and also linked to PTSB (since the channel has been connected to the time slot in the PIDB).
- (m) The following checks are made only in the SM-2000 environment. For DNU related circuits, the audit checks that there is an associated rPLI_LNK tuple. It also checks whether the scanning of the F bits should be

allowed or not.

4. ERROR HANDLING

If any errors are found, the channel data block is reinitialized and placed on the idle list, unless the channel is dedicated to a trunk. Stuck calls are terminated.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BSYPSU

Error Code:	BSYPSU
Error Description:	The inproc attribute indicates that the rCHDB tuple is being used for a busy TSIU to PSU nailed up channel. However the linkages are not correct for this busy state.
Possible Error Effect:	The rCHDB tuple could be incorrectly set up for a TSIU to PSU nailed up channel. This will cause the nailed up channel listening to a different time slot. When the time slot is accessed by the call, assert may occur.
Corrective Action Taken:	The audit recovers the rCHDB to the idle state and attempt to preserve the linkage to rCDBCOM if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of rCHDB tuple.
<i>Bad Data =</i>	1 = Not linked to rCCBCOM active list. 2 = Not linked to rCDBCOM. 3 = Not linked to rPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump =</i>	The rCHDB tuple.

Table 2 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	The rCHDB tuple associated with a system tone failed a point-to-point-back check against the rCDBCOM tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Link to the rCDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump 1 =</i>	rCHDB tuple.
<i>Dump 2 =</i>	rCDBCOM tuple.

Table 3 Error Code: CHDB_TONE_FIELD

Error Code:	CHDB_TONE_FIELD
Error Description:	A CHDB's <i>inproc</i> field was not in the PCTONES state but its tone field was found to contain something but PCTNULL. Only PCTONES-type CHDBs should have a non-NULL value in the tone field.
Possible Error Effect:	An illegal CHDB tuple was lying around. Indicative of a process not properly cleaning up after itself.

Corrective Action Taken:	The offending tuple's tone field was set to PCTNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Bad tone field value.
<i>Good Data =</i>	PCTNULL value.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump =</i>	rCHDB tuple.

Table 4 Error Code: CRAM_E_UPPERBITS

Error Code:	CRAM_E_UPPERBITS
Error Description:	The value in the TSIU3-specific bits of control RAM E (CRAM-E) is invalid for the current set of options enabled for this TSIU model 3.
Possible Error Effect:	Unpredictable effects can occur when an option is enabled and the associated (and needed) hardware is not equipped.
Corrective Action Taken:	The correct value of the options covered in this audit has been written into CRAM-E.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the faulty CHDB tuple.
<i>Bad Data =</i>	A composite of the errors found during the check of the TSIU3-specific bits in CRAM-E. Any bit=1 is in error.
<i>Good Data =</i>	The value of the global TSIU3 option indicator, Gltsiopt. This register uses negative logic. 0 = Option ENABLED. 1 = Option DISABLED.
<i>Logical Index =</i>	The key of the faulty CHDB tuple.
<i>Dump 1 =</i>	The faulty CHDB tuple.
<i>Dump 2 =</i>	The entire structure of type tsCHNCONFIG. See file hdr/smim/SMts_emctl.n for definition of CRAM-E. See file hdr/smim/SMts_cmd3.h for definition of Gltsiopt.

Table 5 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	The input processing code marked in the rCHDB tuple (rCHDB.inproc) was marked to a state of PCASI_FLSH, but the advanced service interface feature bit (rCHDB.proxy_use) was not set. The PCASI_FLSH state is used for advanced service interface trunk signaling. For this state, the <i>proxy_use</i> field in the rCHDB tuple should always be set (marked to DBYES).
Possible Error Effect:	Call failures from loss of channel data, and failures during port conditioning activities.
Corrective Action Taken:	The rCHDB tuple, and all associated channel data are recovered to their idle states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	The state of the <i>proxy_use</i> bit (DBNO).
<i>Good Data =</i>	The correct state of the <i>proxy_use</i> bit (DBYES).
<i>Logical Index =</i>	The key of the rCHDB tuple (rCHDB.chnum).
<i>Dump =</i>	The contents of the rCHDB tuple.

Table 6 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rCHDB tuple that is associated with a system tone is not linked to the DBCADGEN head cell.

Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICHDB tuple will be recovered and linked back onto its idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Owner of the rICHDB tuple.
<i>Good Data</i> =	Value of DBCADGEN head cell.
<i>Logical Index</i> =	Index of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 7 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit could not read the rUNIT_CKT tuple for the signaling terminal equipment (STE) circuit, which is a fourth level parent of the digital signal level 0 (DS0) circuit for a DNU-S facility or the virtual tributary (VT1.5) circuit which is the second level circuit for a DNU-S facility. Check the <i>Error Address</i> field as to determine which circuit could not be read. The rICHDB tuple under audit represents this DS0 circuit and the DS0 is the first (0th) DS0 on the facility. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The rICHDB tuple in error is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = Could not read the VT1.5 circuit. 2 = Could not read the STE circuit.
<i>Bad Data</i> =	The circuit index of the VT1.5 or STE circuit that audit could not read.
<i>Good Data</i> =	If audit could not read the VT1.5 circuit, then the VT1.5 number else the number of the STE circuit(0-5).
<i>Logical Index</i> =	Key of the rICHDB tuple, <i>chnum</i> .
<i>Dump 1</i> =	The rICHDB tuple in error.
<i>Dump 2</i> =	The rICDBCOM tuple this rICHDB is linked to.

Table 8 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	A stable HSM rICHDB which should be scanning for E-bit discontinuity from the periphery is not.
Possible Error Effect:	The call may be stuck and never tear itself down since it will not see E-bit discontinuity.
Corrective Action Taken:	The rICHDB is initialized. As part of initialization, E-bit discontinuity is sent towards the periphery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 9 Error Code: E_BITSEND

Error Code:	E_BITSEND
Error Description:	An idle HSM rICHDB associated with a T1 facility is not sending E-bit discontinuity toward the periphery.
Possible Error Effect:	Calls originating or terminating in an RSM will fail when they attempt to use this channel.
Corrective Action Taken:	The rICHDB is initialized. As part of initialization, E-bit discontinuity is sent towards the periphery.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 10 Error Code: IDLE_CHDB

Error Code:	IDLE_CHDB
Error Description:	An idle rCHDB's associated channel is not set to the idle state in the TSI hardware.
Possible Error Effect:	The rCHDB's associated channel is introducing noise into the time slot.
Corrective Action Taken:	The audit initializes the rCHDB and places it on the idle list. As part of initialization, the audit forces the channel to receive from time slot 0 (idle code) and set all related TSI hardware to the idle state.
Manual Action Required:	Request the TSDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of rCHDB.
<i>Bad Data</i> =	Valid values are: 0 = Assignable channel. 1 = Nailed-up channel.
<i>Good Data</i> =	Time slot 0.
<i>Logical Index</i> =	Key of the rCHDB and channel (they have the same key).
<i>Dump</i> =	rCHDB pc_chn_config structure.

Table 11 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	An rCHDB whose state field indicates that it is idle does not have as its owner the rCHDB idle list head cell.
Possible Error Effect:	The rCHDB is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit places the rCHDB on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Actual rCHDB owner's physical key.
<i>Good Data</i> =	Expected rCHDB owner's physical key.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 12 Error Code: IDLPSU

Error Code:	IDLPSU
Error Description:	The inproc attribute indicates that the rCHDB tuple is being used for an idle TSIU to PSU nailed up channel. However the linkages are not correct for this idle state.
Possible Error Effect:	The rCHDB tuple could be incorrectly set up for a TSIU to PSU nailed up channel. This error will prevent the nailed up channel from being used by call processing or maintenance.
Corrective Action Taken:	The audit recovers the rCHDB to the idle state and attempts to preserve the linkage to rCDBCOM if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Valid values are: 1 = Not linked to rCCBCOM idle list. 2 = Not linked to rCDBCOM.

	3 = Linked to rIPTSB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	The rCHDB tuple.

Table 13 Error Code: INV

Error Code:	INV
Error Description:	The <i>ccstrk</i> field indicates whether or not this channel is associated with a common channel signal (CCS) trunk. The <i>ccstrk</i> field (rCHDB.ccstrk) has been found to be incorrect.
Possible Error Effect:	Trunk offhook reports may be incorrect. The trunk may be incorrectly queued in the wrong unconcentrated buffer.
Corrective Action Taken:	The audit sets the bit to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	<i>ccstrk</i> field of the rCHDB tuple.
<i>Good Data</i> =	<i>lp_type</i> of the trunk.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	The rCHDB tuple.

Table 14 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	The <i>dcstrk</i> field indicates whether or not this channel is associated with a digital AMPS (D-AMPS) trunk. The <i>dcstrk</i> field (rCHDB.dcstrk) has been found to be incorrect.
Possible Error Effect:	Trunk offhook reports may be incorrect. The trunk may be incorrectly queued in the wrong unconcentrated buffer.
Corrective Action Taken:	The audit sets the bit to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	<i>dcstrk</i> field of the rCHDB tuple.
<i>Good Data</i> =	<i>lp_type</i> of the trunk.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	The rCHDB tuple.

Table 15 Error Code: LINKS

Error Code:	LINKS
Error Description:	The rCHDB tuple is linked to rCDBCOM tuple through its <i>chubm.st_link</i> field. If this field is not null, that is not DALNKNUL, then an rCDBCOM tuple is attached to it. This error comes out when audit tried to access this rCDBCOM tuple and failed. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The rCHDB tuple in error is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>cdccm.st_link</i> field of rCDBCOM tuple which shows the linkage to rCHDB tuple.
<i>Bad Data</i> =	The <i>chubm.st_link</i> field minus DBSFCDBCOM which indicates the logical key of the rCDBCOM tuple the rCHDB tuple is attached to and audit failed to access.
<i>Good Data</i> =	Key of the rCDBCOM tuple, <i>name</i> , that audit could not access.
<i>Logical Index</i> =	Key of the rCHDB tuple, <i>chnum</i> .
<i>Dump</i> =	The rCHDB tuple in error.

Table 16 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	This error code indicates an inconsistency between two or more of the following

	data structures: PCtsicn_ch[] = Time slot interchange (TSI) constant channel array. This array is indexed by rICHDB.cnst and contains either a channel number or PCNULLCHAN. PCtsicon[] = TSI constant bit map array. The i'th bit indicates whether the channel in the i'th entry of PCtsicn_ch[] is busy (bit=0) or idle (bit=1). PCtsiap_ch[] = (not for SM-2000) TSI access port channel array. This array is indexed by rICHDB.acspt and contains either a channel number or PCNULLCHAN. PCtsiap[] = (not for SM-2000) TSI access port bit map array. The i'th bit indicates whether the channel in the i'th entry of PCtsiap_ch[] is busy (bit=0) or idle (bit=1). An rICHDB tuple = The tuple's constant channel number (rICHDB.cnst) is an index into PCtsicn_ch[]. Its access port field (rICHDB.acspt) is an index into PCtsiap_ch[]. The index number in this description corresponds to the <i>Logical Index</i> value. Based on the <i>Logical Index</i>, one of the following items will apply: 1 = The index into the PCtsicn_ch[] containing the channel number is not equal to the <i>cnst</i> field in corresponding rICHDB tuple (rICHDB.cnst). 2 = The index into the PCtsiap_ch[] containing the channel number is not equal to the <i>acspt</i> field in corresponding rICHDB tuple (rICHDB.acspt). 3 = The channel number in PCtsicn_ch[rICHDB.cnst] is corrupted. 4 = The channel number in PCtsiap_ch[rICHDB.acspt] is corrupted.
Possible Error Effect:	May not be able to use diagnostics.
Corrective Action Taken:	Valid values are: 1,3 = The audit sets the channel number in PCtsicn_ch[rICHDB.cnst] to PCNULLCHAN and sets the corresponding status bit in PCtsicon[] to idle. 2,4 = The audit sets the channel number in PCtsiap_ch[rICHDB.acspt] to PCNULLCHAN and sets the corresponding status bit in PCtsiap[] to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1,3 = Address of PCtsicn_ch[0] element. 2,4 = Address of PCtsiap_ch[] element.
<i>Bad Data</i> =	Valid values are: 1,3 = Channel number stored in PCtsicn_ch[]. 2,4 = Channel number stored in PCtsiap_ch[].
<i>Good Data</i> =	Index into PC channel array.

<i>Logical Index</i> =	1, 2, 3, or 4 as stated in the Error Description .
<i>Dump 1</i> =	Valid values are: 1,3 = Element of PC bit array, PCtsicon[]. 2,4 = Element of PC bit array, PCtsiapt[].
<i>Dump 2</i> =	Valid values are: 1,3 = Element of PC constant channel, PCtsicn_ch[]. 2,4 = Element of PC access port channel array, PCtsiapt[].
<i>Dump 3</i> =	The rICHDB tuple.

Table 17 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	This error code indicates an inconsistency between two or more of the following data structures: PCtsicn_ch[] = TSI constant channel array. Always, PCtsicn_ch[] = PCNULLCHAN. PCtsicon[] = TSI constant bit map array. Its 0th bit of the first entry is marked busy (bit=0). PCtsiap_ch[] = (not for SM-2000) TSI access port channel array. Always, PCtsiap_ch[] = PCNULLCHAN. PCtsiapt[] = (not for SM-2000) TSI access port bit map array. Its 0th bit of the first entry is marked busy (bit = 0). The index number in this description corresponds to the <i>Logical Index</i> value. Based on <i>Logical Index</i>, one of the following items will apply: 1 = Either PCtsicn_ch[] is not equal to PCNULLCHAN, or the corresponding status bit in PCtsicon[] (the 0th bit) is marked idle. 2 = Either PCtsiap_ch[] is not equal to PCNULLCHAN, or the corresponding status bit in PCtsiapt[] (the 0th bit) is marked idle.
Possible Error Effect:	May not be able to use diagnostics.
Corrective Action Taken:	Valid values are: 1 = The audit sets PCtsicn_ch[] = PCNULLCHAN and the corresponding bit in constant bit map array (PCtsicon[]) to busy. 2 = The audit sets PCtsiap_ch[] = PCNULLCHAN and the corresponding bit in access port bit map array (PCtsiapt[]) to busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = Address of PCtsicn_ch[]. 2 = Address of PCtsiap_ch[].

<i>Bad Data</i> =	Valid values are: 1 = Channel number stored in PCtsicn_ch[]. 2 = Channel number stored in PCtsiap_ch[].
<i>Good Data</i> =	Valid values are: 1 = Status bit for this channel number (PCtsicon[]). 2 = Status bit for this channel number (PCtsiapt[]).
<i>Logical Index</i> =	1 or 2 as stated in the Error Description .
<i>Dump 1</i> =	PCtsicon[].
<i>Dump 2</i> =	PCtsiapt[].

Table 18 Error Code: NOSUPV

Error Code:	NOSUPV
Error Description:	The channel has supervision incorrectly turned off.
Possible Error Effect:	The associated port cannot originate or disconnect.
Corrective Action Taken:	The rICHDB tuple is reinitialized and any associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 19 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit could not read the corresponding rIPLI_LNK tuple for a rICHDB tuple related to a DNU_S circuit. The audit will delete the rICHDB tuple and report a manual action error because the entry in PCLNK2USE exists which indicates the static rILNK_USE also exists. The PCLNK2USE is an array that contains all the links and the associated usage. It is built off of data in both the RLLNK_USE and rIPLI_LNK tuples. Therefore, it must have existed at one point. The <i>tsi_link</i> field in the dump is the key of the missing rIPLI_LNK tuple. It should be noted that this error will only be seen in the SM-2000 environment.
Possible Error Effect:	DNU_S facilities may be degraded since some static data is missing.
Corrective Action Taken:	The rICHDB tuple is deleted.
Manual Action Required:	The missing rIPLI_LNK tuple should be inserted.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	The value of the <i>tsi_link</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 20 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The index number in this description corresponds to the good data value. So, based on good data, one of the following items will apply: 0 = The rICHDB tuple's constant channel number field (rICHDB.cnst) is not set to the upper bound (PCNUMCN). 1 = (not for SM-2000) The rICHDB tuple's access port field (rICHDB.acspt) is not set to the upper bound (PCNUMAP).

Possible Error Effect:	May not be able to run diagnostics that require the use of a access or constant TSI port.
Corrective Action Taken:	The constant channel number field (rICHDB.cnst) is set to the upper bound (PCNUMCN) and the access port field (rICHDB.acspt) is set to the upper bound (PCNUMAP).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Valid values are: 0 = rICHDB tuple's <i>cnst</i> field. 1 = rICHDB tuple's <i>acspt</i> field.
<i>Good Data =</i>	0 or 1 as stated in the Error Description .
<i>Logical Index =</i>	Key of the rICHDB tuple (rICHDB.chnum).
<i>Dump =</i>	rICHDB tuple.

Table 21 Error Code: PC_QUAR

Error Code:	PC_QUAR
Error Description:	The input processing code (rICHDB.inproc) indicates that peripheral control encountered signaling, timing, or signal processing errors associated with this rICHDB tuple. When peripheral control encounters one of these types of errors the input processing code is set to peripheral control quarantine (rICHDB.inproc == PCQUARAN) indicating that the rICHDB tuple should be recovered and reinitialized by the audit.
Possible Error Effect:	The state of the rICHDB tuple is unknown and this may cause line and trunks from originating.
Corrective Action Taken:	The rICHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Input processing doce (rICHDB.inproc = PCQUARAN)
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	rICHDB tuple.

Table 22 Error Code: RSMCHAN

Error Code:	RSMCHAN
Error Description:	Invalid DB return when accessing RLRSCHAN relation, possible ODD problem. This error will continue to print out until fixed by the office personnel.
Possible Error Effect:	Relation cannot be accessed.
Corrective Action Taken:	None.
Manual Action Required:	If error repeats, investigate the ODD data.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLRSCHAN relation.
<i>Dump =</i>	rIRSMCHAN buffer.

Table 23 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	A bad linkage has been found in a stable rICHDB. One of the following errors has occurred: - The rICHDB is linked to a head cell (owner's logical key<0). - The rICHDB is linked to a dummy tuple (owner's logical key=0).

	The linkage is not being used (linkage physical key in the rICHDB=0).
	All of the links in the rICHDB should be active in a stable state.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICHDB. This action terminates the active call that uses the rICHDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Linkage identification.
	0 = rUBHC slave linkage.
	1 = rCCBCOM sink linkage.
	2 = rCDBCOM linkage.
	3 = rCCBCOM source linkage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump 1 =</i>	Linkage identification.
<i>Dump 2 =</i>	rICHDB tuple.
	The last 4 bytes of the CHDB dump indicate to which CDBCOM tuple is linked. The key of CDBCOM tuple represents the actual circuit or trunk name. This key can be used to access FC_TRUNK or CKTDATA to obtain the detailed information about that circuit or trunk.

Table 24 Error Code: STATE

Error Code:	STATE
Error Description:	Illegal state in the SP. The SP is not scanning for off-hooks for this channel.
Possible Error Effect:	Either the SP ignore RAM is wrong or the rep.ag field in the rICHDB is wrong.
Corrective Action Taken:	This channel cannot be used for phone calls.
Manual Action Required:	This rICHDB will be put on the idle list at which time scanning will be turned on for off-hooks.
Dump Description:	None.
<i>Error Address =</i>	Address of rICHDB.
<i>Bad Data =</i>	SP ignore RAM.
<i>Good Data =</i>	Expected value.
<i>Logical Index =</i>	rICHDB channel number.
<i>Dump =</i>	rICHDB tuple.

Table 25 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rICHDB tuple has an invalid state indicator in the inproc field.
Possible Error Effect:	The state of the rICHDB is unknown.
Corrective Action Taken:	The rICHDB tuple is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Bad state variable.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	rICHDB tuple.

Table 26 Error Code: STATUS

Error Code:	STATUS
Error Description:	The <i>four_wire</i> bit of a rICHDB tuple has incorrectly been turned on or off. The

	<i>four_wire</i> bit should be set according to the service class in rIFC_PORTTYP tuple.
Possible Error Effect:	The rICHDB is incorrectly set up for <i>four_wire</i> access. This will impact the origination and termination from/to 4-wire trunk.
Corrective Action Taken:	The CHDB audit sets the bit to a correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	The <i>four_wire</i> field of the rICHDB tuple.
<i>Good Data</i> =	Not Applicable.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 27 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than five minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rICHDB tuple is idled and the call is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 28 Error Code: TONE_CHDB

Error Code:	TONE_CHDB
Error Description:	The rICHDB tuple that is associated with a system tone failed a point-to-point-back check with the rTONECH tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Tone field of the rICHDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump 1</i> =	rICHDB tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 29 Error Code: TRUNK_LINK

Error Code:	TRUNK_LINK
Error Description:	An idle rICHDB tuple, used for a trunk application, has an invalid rCDBCOM tuple linkage.
Possible Error Effect:	The trunk circuit is unusable.
Corrective Action Taken:	The audit initializes the rICHDB and places it on the idle list. When the CDBCOM audit runs, the audit links the rICHDB to the proper rCDBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Linkage ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump 1</i> =	Linkage ID.
<i>Dump 2</i> =	rICHDB tuple.

Table 30 Error Code: TSOWNED

Error Code:	TSOWNED
Error Description:	An rICHDB tuple's local digital services function (LDSF) tone generator time slot does not match the time slot number held for it in write-protected data

	(rICHDB.lsf ts != PCdsc3ckts[unit].pcdsc3ts[chan]).
Possible Error Effect:	The system tone associated with the rICHDB tuple's channel is not usable.
Corrective Action Taken:	The rICHDB tuple is reinitialized. Any associated processes are purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The associated system tone's ldsuclnn (rITONECH.ldsuclnn). This data is used to derive the unit and channel numbers used to index into PCdsc3ckts[].pcdsc3ts[].
<i>Bad Data</i> =	The rICHDB tuple's LDSF time slot (rICHDB.lsf ts).
<i>Good Data</i> =	The time slot from PCdsc3ckts[] (PCdsc3ckts[unit].pcdsc3ts[chan]).
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump 1</i> =	The rICHDB tuple.
<i>Dump 2</i> =	The associated rITONECH tuple.

Table 31 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The <i>rep_f</i> field of the rICHDB tuple, which shows whether the F-bit scanning should be allowed or not, did not match the same value calculated by the audit for a DNU-S facility. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The <i>rep_f</i> field of rICHDB tuple is corrected with the value calculated by audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The unit type, <i>utype</i> field of rCDBCOM tuple linked to rICHDB tuple. This should be SMDNU S.
<i>Bad Data</i> =	The <i>rep_f</i> field of the rICHDB tuple.
<i>Good Data</i> =	The value of the <i>rep_f</i> field calculated by audit.
<i>Logical Index</i> =	Key of the rICHDB tuple, <i>chnum</i> .
<i>Dump 1</i> =	The rICHDB tuple in error.
<i>Dump 2</i> =	The rCDBCOM tuple linked to rICHDB tuple.
<i>Dump 3</i> =	The rSMEST tuple of the STE circuit, if accessed by audit.

Table 32 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rICHDB has nonzero linkage in one of its linkage blocks.
Possible Error Effect:	Some resource is linked to a rICHDB on the idle link list and may be unusable by call processing.
Corrective Action Taken:	The audit initializes the rICHDB and places it on the idle list. As part of initialization, it zeros the linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Bad linkage contents.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

CHDB-B

Software Release: 5E11 - 5E12

1. CHANNEL DATA BLOCK (CHDB) AUDIT

The channel data block (CHDB) is an abstraction of the peripheral side channel of the network. The channel data block is used to allocate channels to lines for call processing. A line is assigned a channel when the customer goes off-hook. Trunk circuits have dedicated channels. If there is an error in the channel data block data associated with an active call, it will interfere with successful call processing. An error in the channel data block data associated with an idle trunk channel may result in lost supervision.

Time available on the link from the switching module (SM) to a remote switching module (RSM) is divided into time slots. Each time slot carries a voice signal, which is composed of 16 bit pulses. Eight of the bits contain the encoding of the voice signal, and eight bits are used for signaling. One of these eight signaling bits, the E signaling bit, confirms the connection between the originating and terminating processes. The possible bit values are continuity and discontinuity.

The channel data block audit (CHDB) ensures the integrity of the channel data block relation (RLCHDB). The audit performs validity checks on the head cells in the relation and on the rCHDB tuples.

2. DATA STRUCTURES

The data structures checked by the CHDB audit are the channel data block relation (RLCHDB), the unconcentrated buffer head cell relation (RLUBHC), the channel control block common area relation (RLCCBCOM), the circuit data block common area relation (RLCDBCOM) and the PIDB time slot block relation (RLPTSB). The following data structures are used for SM-2000 processors only:

ldsfs_ts = A field in the RLCHDB relation that represents the LDSF TSI time slot associated with the CHDB.
rLNK_USE = The TSI link type relation structure.

Figure 1 illustrates the layout of rCHDB and the linkages associated with it.

3. AUDIT CHECKS

To verify the linkages associated with the channel data blocks, the CHDB audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the DRLINK driver executes a semantic check function that performs the following checks based on the state of the channel.

- The audit checks the channel state (inproc) field against the channel data block linkages. Idle channel data blocks should be owned by the channel data block idle list head cell or else the channel data blocks will be unavailable for call processing. An idle channel data block should have zero linkages except for those channel blocks dedicated to trunks. If an idle channel block has linkages, other resources cannot be used for call processing. An idle channel data block used for a trunk application should have a valid linkage to the trunk circuit data block. Otherwise, the trunk circuit cannot be used by call processing.
- If the inproc value indicates a special idle state for a given trunk type, the audit verifies the following:
 - The rCHDB tuple is properly linked to the channel data block idle list.
 - The rCHDB tuple is marked busy in the channel allocation maps.
 - The rCHDB tuple is properly linked to a circuit data block (trunk circuits).

- If an idle rCHDB tuple is associated with an HSM T1 facility, the audit verifies that the E-bit discontinuity is sent to the RSM.
- Any channel data block (except RSM control time slots in HSMs) associated with an idle channel (whether dedicated to a trunk or available for connection to a line) should be receiving from the time slot 0 (idle code) to prevent the signal of another telephone call from being transmitted to more than one destination. The audit performs a hardware read to determine which time slot is connected to the channel.
- If the inproc value indicates that the channel data block is in a stable busy state, the audit verifies the following:
 - The rCHDB tuple is properly linked to an rCCBCOM tuple owned by a terminal process.
 - The rCHDB tuple is properly linked to an rUBHC tuple.
 - The rCHDB tuple is marked busy in the channel allocation maps.
 - The rCHDB tuple is properly linked to rCDBCOM tuple or rPTSB tuple.
- If a stable rCHDB tuple is associated with an HSM T1 facility, the audit verifies that E-bit changes are being scanned. The audit also verifies that scanning detects continuity unless the tuple is on a hit-timing list.
- Sometimes the inproc value indicates that the state of the channel is expected to change soon. This situation usually occurs during transient states such as trunk signaling, digit collection for lines and trunks, and flash and hit timing for lines and trunks. If the inproc value indicates this case, the audit verifies the following:
 - The rCHDB tuple is marked busy in the channel allocation maps.
 - The rCHDB tuple is not in a transient state for more than five minutes.
- A few channels are used for special system purposes, such as audible ringing tone. If a transmission test facility (TTF) is used, three channels are dedicated to it. In this case, the inproc value indicates that the PC subsystem should ignore supervision of the channels, and the audit verifies the following:
 - The rCHDB tuples are properly linked to a special list.
 - The rCHDB tuples are marked busy in the channel allocation maps.
- The audit checks the four_wire bit in the rCHDB tuple. The four_wire bit in the rCHDB should only be set if the trunk using this time slot is 4-WIRE trunk or a precedence and preemption trunk.
- To save the call on hold during the selective initialization, the CHDB audit checks the path state in the PHDB. If the path is open and the inproc state is stable, then the call on hold will be saved.
- For TSIU to PSU nailed up channel in the idle state, verify that the CHDB tuple is properly linked to idle list (CCBCOM), linked to the logical time slot circuit (CDBCOM) and not linked to PTSB (since an idle channel has not been connected to the time slot in the PIDB).
- For TSIU to PSU nailed up channel in the stable state, verify that the CHDB tuple is properly linked to an active list (CCBCOM), linked to the logical time slot circuit (CDBCOM) and also linked to PTSB (since the channel has been connected to the time slot in the PIDB).
- The following checks are made only in the SM-2000 environment. For DNU related circuits, the audit checks that there is an associated rPLI_LNK tuple. It also checks whether the scanning of the F bits should be allowed or not.

4. ERROR HANDLING

If any errors are found, the channel data block is reinitialized and placed on the idle list, unless the channel is dedicated to a trunk. Stuck calls are terminated.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BSYPSU

Error Code:	BSYPSU
Error Description:	The inproc attribute indicates that the rCHDB tuple is being used for a busy TSIU to PSU nailed up channel. However the linkages are not correct for this busy state.
Possible Error Effect:	The rCHDB tuple could be incorrectly set up for a TSIU to PSU nailed up channel. This will cause the nailed up channel listening to a different time slot. When the time slot is accessed by the call, assert may occur.
Corrective Action Taken:	The audit recovers the rCHDB to the idle state and attempt to preserve the linkage to rCDBCOM if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of rCHDB tuple.
<i>Bad Data =</i>	Valid values are: 1 = Not linked to rCCBCOM active list. 2 = Not linked to rCDBCOM. 3 = Not linked to rPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump =</i>	The rCHDB tuple.

Table 2 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	The rCHDB tuple associated with a system tone failed a point-to-point-back check against the rCDBCOM tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Link to the rCDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump 1 =</i>	rCHDB tuple.
<i>Dump 2 =</i>	rCDBCOM tuple.

Table 3 Error Code: CHDB_TONE_FIELD

Error Code:	CHDB_TONE_FIELD
Error Description:	A CHDB's <i>inproc</i> field was not in the PCTONES state but its tone field was found to contain something but PCTNULL. Only PCTONES-type CHDBs should have a non-NULL value in the tone field.
Possible Error Effect:	An illegal CHDB tuple was lying around. Indicative of a process not properly cleaning up after itself.

Corrective Action Taken:	The offending tuple's tone field was set to PCTNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Bad tone field value.
<i>Good Data =</i>	PCTNULL value.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump =</i>	rCHDB tuple.

Table 4 Error Code: CRAM_E_UPPERBITS

Error Code:	CRAM_E_UPPERBITS
Error Description:	The value in the TSIU3-specific bits of control RAM E (CRAM-E) is invalid for the current set of options enabled for this TSIU model 3.
Possible Error Effect:	Unpredictable effects can occur when an option is enabled and the associated (and needed) hardware is not equipped.
Corrective Action Taken:	The correct value of the options covered in this audit has been written into CRAM-E.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the faulty CHDB tuple.
<i>Bad Data =</i>	A composite of the errors found during the check of the TSIU3-specific bits in CRAM-E. Any bit=1 is in error.
<i>Good Data =</i>	The value of the global TSIU3 option indicator, Gltsiopt. This register uses negative logic. 0 = Option enabled. 1 = Option disabled.
<i>Logical Index =</i>	The key of the faulty CHDB tuple.
<i>Dump 1 =</i>	The faulty CHDB tuple.
<i>Dump 2 =</i>	The entire structure of type tsCHNCONFIG. See file hdr/smim/SMts_emctl.n for definition of CRAM-E. See file hdr/smim/SMts_cmd3.h for definition of Gltsiopt.

Table 5 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	The input processing code marked in the rCHDB tuple (rCHDB.inproc) was marked to a state of PCASI_FLSH, but the advanced service interface feature bit (rCHDB.proxy_use) was not set. The PCASI_FLSH state is used for advanced service interface trunk signaling. For this state, the <i>proxy_use</i> field in the rCHDB tuple should always be set (marked to DBYES).
Possible Error Effect:	Call failures from loss of channel data, and failures during port conditioning activities.
Corrective Action Taken:	The rCHDB tuple, and all associated channel data are recovered to their idle states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	The state of the <i>proxy_use</i> bit (DBNO).
<i>Good Data =</i>	The correct state of the <i>proxy_use</i> bit (DBYES).
<i>Logical Index =</i>	The key of the rCHDB tuple (rCHDB.chnum).
<i>Dump =</i>	The contents of the rCHDB tuple.

Table 6 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rCHDB tuple that is associated with a system tone is not linked to the DBCADGEN head cell.

Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICHDB tuple will be recovered and linked back onto its idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Owner of the rICHDB tuple.
<i>Good Data</i> =	Value of DBCADGEN head cell.
<i>Logical Index</i> =	Index of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 7 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit could not read the rUNIT_CKT tuple for the signaling terminal equipment (STE) circuit, which is a fourth level parent of the digital signal level 0 (DS0) circuit for a DNU-S facility or the virtual tributary (VT1.5) circuit which is the second level circuit for a DNU-S facility. Check the <i>Error Address</i> field as to determine which circuit could not be read. The rICHDB tuple under audit represents this DS0 circuit and the DS0 is the first (0th) DS0 on the facility. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The rICHDB tuple in error is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = Could not read the VT1.5 circuit. 2 = Could not read the STE circuit.
<i>Bad Data</i> =	The circuit index of the VT1.5 or STE circuit that audit could not read.
<i>Good Data</i> =	If audit could not read the VT1.5 circuit, then the VT1.5 number else the number of the STE circuit(0-5).
<i>Logical Index</i> =	Key of the rICHDB tuple, <i>chnum</i> .
<i>Dump 1</i> =	The rICHDB tuple in error.
<i>Dump 2</i> =	The rICDBCOM tuple this rICHDB is linked to.

Table 8 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	A stable HSM rICHDB which should be scanning for E-bit discontinuity from the periphery is not.
Possible Error Effect:	The call may be stuck and never tear itself down since it will not see E-bit discontinuity.
Corrective Action Taken:	The rICHDB is initialized. As part of initialization, E-bit discontinuity is sent towards the periphery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 9 Error Code: E_BITSEND

Error Code:	E_BITSEND
Error Description:	An idle HSM rICHDB associated with a T1 facility is not sending E-bit discontinuity toward the periphery.
Possible Error Effect:	Calls originating or terminating in an RSM will fail when they attempt to use this channel.
Corrective Action Taken:	The rICHDB is initialized. As part of initialization, E-bit discontinuity is sent towards the periphery.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 10 Error Code: IDLE_CHDB

Error Code:	IDLE_CHDB
Error Description:	An idle rCHDB's associated channel is not set to the idle state in the TSI hardware.
Possible Error Effect:	The rCHDB's associated channel is introducing noise into the time slot.
Corrective Action Taken:	The audit initializes the rCHDB and places it on the idle list. As part of initialization, the audit forces the channel to receive from time slot 0 (idle code) and set all related TSI hardware to the idle state.
Manual Action Required:	Request the TSDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of rCHDB.
<i>Bad Data</i> =	Valid values are: 0 = Assignable channel. 1 = Nailed-up channel.
<i>Good Data</i> =	Time slot 0.
<i>Logical Index</i> =	Key of the rCHDB and channel (they have the same key).
<i>Dump</i> =	rCHDB pc_chn_config structure.

Table 11 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	An rCHDB whose state field indicates that it is idle does not have as its owner the rCHDB idle list head cell.
Possible Error Effect:	The rCHDB is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit places the rCHDB on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Actual rCHDB owner's physical key.
<i>Good Data</i> =	Expected rCHDB owner's physical key.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 12 Error Code: IDLPSU

Error Code:	IDLPSU
Error Description:	The inproc attribute indicates that the rCHDB tuple is being used for an idle TSIU to PSU nailed up channel. However the linkages are not correct for this idle state.
Possible Error Effect:	The rCHDB tuple could be incorrectly set up for a TSIU to PSU nailed up channel. This error will prevent the nailed up channel from being used by call processing or maintenance.
Corrective Action Taken:	The audit recovers the rCHDB to the idle state and attempts to preserve the linkage to rCDBCOM if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Valid values are: 1 = Not linked to rCCBCOM idle list. 2 = Not linked to rCDBCOM.

	3 = Linked to rIPTSB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	The rCHDB tuple.

Table 13 Error Code: INV

Error Code:	INV
Error Description:	The <i>ccstrk</i> field indicates whether or not this channel is associated with a common channel signal (CCS) trunk. The <i>ccstrk</i> field (rCHDB.ccstrk) has been found to be incorrect.
Possible Error Effect:	Trunk offhook reports may be incorrect. The trunk may be incorrectly queued in the wrong unconcentrated buffer.
Corrective Action Taken:	The audit sets the bit to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	<i>ccstrk</i> field of the rCHDB tuple.
<i>Good Data</i> =	<i>lp_type</i> of the trunk.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	The rCHDB tuple.

Table 14 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	The <i>dcstrk</i> field indicates whether or not this channel is associated with a digital AMPS (D-AMPS) trunk. The <i>dcstrk</i> field (rCHDB.dcstrk) has been found to be incorrect.
Possible Error Effect:	Trunk offhook reports may be incorrect. The trunk may be incorrectly queued in the wrong unconcentrated buffer.
Corrective Action Taken:	The audit sets the bit to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	<i>dcstrk</i> field of the rCHDB tuple.
<i>Good Data</i> =	<i>lp_type</i> of the trunk.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	The rCHDB tuple.

Table 15 Error Code: LINKS

Error Code:	LINKS
Error Description:	The rCHDB tuple is linked to rCDBCOM tuple through its <i>chubm.st_link</i> field. If this field is not null, that is not DALNKNUL, then an rCDBCOM tuple is attached to it. This error comes out when audit tried to access this rCDBCOM tuple and failed. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The rCHDB tuple in error is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>cdccm.st_link</i> field of rCDBCOM tuple which shows the linkage to rCHDB tuple.
<i>Bad Data</i> =	The <i>chubm.st_link</i> field minus DBSFCDDBCOM which indicates the logical key of the rCDBCOM tuple the rCHDB tuple is attached to and audit failed to access.
<i>Good Data</i> =	Key of the rCDBCOM tuple, <i>name</i> , that audit could not access.
<i>Logical Index</i> =	Key of the rCHDB tuple, <i>chnum</i> .
<i>Dump</i> =	The rCHDB tuple in error.

Table 16 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	This error code indicates an inconsistency between two or more of the following

	data structures: PCtsicn_ch[] = Time slot interchange (TSI) constant channel array. This array is indexed by rICHDB.cnst and contains either a channel number or PCNULLCHAN. PCtsicon[] = TSI constant bit map array. The i'th bit indicates whether the channel in the i'th entry of PCtsicn_ch[] is busy (bit=0) or idle (bit=1). PCtsiap_ch[] = (not for SM-2000) TSI access port channel array. This array is indexed by rICHDB.acspt and contains either a channel number or PCNULLCHAN. PCtsiap[] = (not for SM-2000) TSI access port bit map array. The i'th bit indicates whether the channel in the i'th entry of PCtsiap_ch[] is busy (bit=0) or idle (bit=1). An rICHDB tuple is an index into PCtsicn_ch[]. Its access port field (rICHDB.acspt) is an index into PCtsiap_ch[]. The index number in this description corresponds to the "Logical Index" value. Based on the "Logical Index", one of the following items will apply: 1 = The index into the PCtsicn_ch[] containing the channel number is not equal to the <i>cnst</i> field in corresponding rICHDB tuple (rICHDB.cnst). 2 = The index into the PCtsiap_ch[] containing the channel number is not equal to the <i>acspt</i> field in corresponding rICHDB tuple (rICHDB.acspt). 3 = The channel number in PCtsicn_ch[rICHDB.cnst] is corrupted. 4 = The channel number in PCtsiap_ch[rICHDB.acspt] is corrupted.
Possible Error Effect:	May not be able to use Diagnostics.
Corrective Action Taken:	Corresponds to Error Description as follows: 1,3 = The audit sets the channel number in PCtsicn_ch[rICHDB.cnst] to PCNULLCHAN and sets the corresponding status bit in PCtsicon[] to idle. 2,4 = The audit sets the channel number in PCtsiap_ch[rICHDB.acspt] to PCNULLCHAN and sets the corresponding status bit in PCtsiap[] to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Corresponds to Error Description as follows: 1,3 = Address of PCtsicn_ch[] element. 2,4 = Address of PCtsiap_ch[] element.
<i>Bad Data</i> =	Corresponds to Error Description as follows: 1,3 = Channel number stored in PCtsicn_ch[]. 2,4 = Channel number stored in PCtsiap_ch[].
<i>Good Data</i> =	Index into PC channel array.
<i>Logical Index</i> =	1, 2, 3, or 4 as stated in the Error Description.

<i>Dump 1 =</i>	Corresponds to Error Description as follows: 1,3 = Element of PC bit array, PCtsicon[]. 2,4 = Element of PC bit array, PCtsiapt[].
<i>Dump 2 =</i>	Corresponds to Error Description as follows: 1,3 = Element of PC constant channel, PCtsicn_ch[]. 2,4 = Element of PC access port channel array, PCtsiapt[].
<i>Dump 3 =</i>	The rICHDB tuple.

Table 17 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	This error code indicates an inconsistency between two or more of the following data structures: PCtsicn_ch[] = Time slot interchange (TSI) constant channel array. Always, PCtsicn_ch[] = PCNULLCHAN. PCtsicon[] = TSI constant bit map array. Its 0'th bit of the first entry is marked busy (bit=0). PCtsiap_ch[] = (not for SM-2000) TSI access port channel array. Always, PCtsiap_ch[] = PCNULLCHAN. PCtsiapt[] = (not for SM-2000) TSI access port bit map array. Its 0'th bit of the first entry is marked busy (bit = 0). The index number in this description corresponds to the "Logical Index" value. Based on "Logical Index", one of the following items will apply: 1 = Either PCtsicn_ch[] is not equal to PCNULLCHAN, or the corresponding status bit in PCtsicon[] (the 0'th bit) is marked idle. 2 = Either PCtsiap_ch[] is not equal to PCNULLCHAN, or the corresponding status bit in PCtsiapt[] (the 0'th bit) is marked idle.
Possible Error Effect:	May not be able to use Diagnostics.
Corrective Action Taken:	Corresponds to Error Description as follows: 1 = The audit sets PCtsicn_ch[] = PCNULLCHAN and the corresponding bit in constant bit map array (PCtsicon[]) to busy. 2 = The audit sets PCtsiap_ch[] = PCNULLCHAN and the corresponding bit in access port bit map array (PCtsiapt[]) to busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to Error Description as follows: 1 = Address of PCtsicn_ch[]. 2 = Address of PCtsiap_ch[].
<i>Bad Data =</i>	Corresponds to Error Description as follows:

	1 = Channel number stored in PCtsicn_ch[]. 2 = Channel number stored in PCtsiap_ch[].
<i>Good Data</i> =	Corresponds to Error Description as follows: 1 = Status bit for this channel number (PCtsicon[]). 2 = Status bit for this channel number (PCtsiap[]).
<i>Logical Index</i> =	1 or 2 as stated in the Error Description.
<i>Dump 1</i> =	PCtsicon[].
<i>Dump 2</i> =	PCtsiap[].

Table 18 Error Code: NOSUPV

Error Code:	NOSUPV
Error Description:	The channel has supervision incorrectly turned off.
Possible Error Effect:	The associated port cannot originate or disconnect.
Corrective Action Taken:	The rICHDB tuple is reinitialized and any associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 19 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit could not read the corresponding rIPLI_LNK tuple for a rICHDB tuple related to a DNU_S circuit. The audit will delete the rICHDB tuple and report a manual action error because the entry in PCLNK2USE exists which indicates the static rLNK_USE also exists. The PCLNK2USE is an array that contains all the links and the associated usage. It is built off of data in both the RLLNK_USE and rIPLI_LNK tuples. Therefore, it must have existed at one point. The <i>tsi_link</i> field in the dump is the key of the missing rIPLI_LNK tuple. It should be noted that this error will only be seen in the SM-2000 environment.
Possible Error Effect:	DNU_S facilities may be degraded since some static data is missing.
Corrective Action Taken:	The rICHDB tuple is deleted.
Manual Action Required:	The missing rIPLI_LNK tuple should be inserted.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	The value of the <i>tsi_link</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 20 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The index number in this description corresponds to the good data value. So, based on good data, one of the following items will apply: 0 = The rICHDB tuple's constant channel number field (rICHDB.cnst) is not set to the upper bound (PCNUMCN). 1 = (not for SM-2000) The rICHDB tuple's access port field (rICHDB.acspt) is not set to the upper bound (PCNUMAP).
Possible Error Effect:	May not be able to run diagnostics that require the use of a access or constant

	TSI port.
Corrective Action Taken:	The constant channel number field (rCHDB.cnst) is set to the upper bound (PCNUMCN) and the access port field (rCHDB.acspt) is set to the upper bound (PCNUMAP).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Corresponds to Error Description as follows: 0 = rCHDB tuple's <i>cnst</i> field. 1 = rCHDB tuple's <i>acspt</i> field.
<i>Good Data =</i>	0 or 1 as stated in the Error Description.
<i>Logical Index =</i>	Key of the rCHDB tuple (rCHDB.chnum).
<i>Dump =</i>	rCHDB tuple.

Table 21 Error Code: PC_QUAR

Error Code:	PC_QUAR
Error Description:	The input processing code (rCHDB.inproc) indicates that peripheral control encountered signaling, timing, or signal processing errors associated with this rCHDB tuple. When peripheral control encounters one of these types of errors the input processing code is set to peripheral control quarantine (rCHDB.inproc == PCQUARAN) indicating that the rCHDB tuple should be recovered and reinitialized by the audit.
Possible Error Effect:	The state of the rCHDB tuple is unknown and this may cause line and trunks from originating.
Corrective Action Taken:	The rCHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Input processing code (rCHDB.inproc = PCQUARAN)
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump =</i>	rCHDB tuple.

Table 22 Error Code: RSMCHAN

Error Code:	RSMCHAN
Error Description:	Invalid DB return when accessing RLRSCHAN relation, possible ODD problem. This error will continue to print out until fixed by office personnel.
Possible Error Effect:	Relation cannot be accessed.
Corrective Action Taken:	None.
Manual Action Required:	If error repeats, investigate the ODD data.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLRSCHAN relation.
<i>Dump =</i>	rRSMCHAN buffer.

Table 23 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	A bad linkage has been found in a stable rCHDB. One of the following errors has occurred: • The rCHDB is linked to a head cell (owner's logical key<0). • The rCHDB is linked to a dummy tuple (owner's logical key=0). • The linkage is not being used (linkage physical key in the rCHDB=0).

	All of the links in the rICHDB should be active in a stable state.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICHDB. This action terminates the active call that uses the rICHDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Linkage identification. 0 = rIUBHC slave linkage. 1 = rICBCOM sink linkage. 2 = rICDBCOM linkage. 3 = rICBCOM source linkage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump 1 =</i>	Linkage identification.
<i>Dump 2 =</i>	rICHDB tuple. The last 4 bytes of the CHDB dump indicate to which CDBCOM tuple is linked. The key of CDBCOM tuple represents the actual circuit or trunk name. This key can be used to access FC_TRUNK or CKTDATA to obtain the detailed information about that circuit or trunk.

Table 24 Error Code: STATE

Error Code:	STATE
Error Description:	Illegal state in the SP. The SP is not scanning for off-hooks for this channel. Either the SP ignore RAM is wrong or the rep.ag field in the rICHDB is wrong.
Possible Error Effect:	This channel cannot be used for phone calls.
Corrective Action Taken:	This rICHDB will be put on the idle list at which time scanning will be turned on for off-hooks.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICHDB.
<i>Bad Data =</i>	SP ignore RAM.
<i>Good Data =</i>	Expected value.
<i>Logical Index =</i>	rICHDB channel number.
<i>Dump =</i>	rICHDB tuple.

Table 25 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rICHDB tuple has an invalid state indicator in the inproc field.
Possible Error Effect:	The state of the rICHDB is unknown.
Corrective Action Taken:	The rICHDB tuple is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Bad state variable.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	rICHDB tuple.

Table 26 Error Code: STATUS

Error Code:	STATUS
Error Description:	The <i>four_wire</i> bit of a rICHDB tuple has incorrectly been turned on or off. The <i>four_wire</i> bit should be set according to the service class in rFC_PORTTYP

	tuple.
Possible Error Effect:	The rCHDB is incorrectly set up for <i>four_wire</i> access. This will impact the origination and termination from/to 4-wire trunk.
Corrective Action Taken:	The CHDB audit sets the bit to a correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	The <i>four_wire</i> field of the rCHDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 27 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than five minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rCHDB tuple is idled and the call is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 28 Error Code: TONE_CHDB

Error Code:	TONE_CHDB
Error Description:	The rCHDB tuple that is associated with a system tone failed a point-to-point-back check with the rTONECH tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Tone field of the rCHDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump 1</i> =	rCHDB tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 29 Error Code: TRUNK_LINK

Error Code:	TRUNK_LINK
Error Description:	An idle rCHDB tuple, used for a trunk application, has an invalid rCDBCOM tuple linkage.
Possible Error Effect:	The trunk circuit is unusable.
Corrective Action Taken:	The audit initializes the rCHDB and places it on the idle list. When the CDBCOM audit runs, the audit links the rCHDB to the proper rCDBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Linkage ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump 1</i> =	Linkage ID.
<i>Dump 2</i> =	rCHDB tuple.

Table 30 Error Code: TSOWNED

Error Code:	TSOWNED
Error Description:	An rCHDB tuple's local digital services function (LDSF) tone generator time slot does not match the time slot number held for it in write-protected data (rCHDB.lsf ts != PCdsc3cts[unit].pcdsc3ts[chan]).
Possible Error Effect:	The system tone associated with the rCHDB tuple's channel is not usable.

Corrective Action Taken:	The rCHDB tuple is reinitialized. Any associated processes are purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The associated system tone's ldsucInn (rITONECH.ldsucInn). This data is used to derive the unit and channel numbers used to index into PCdsc3ckts[][.pcdsc3ts[]].
<i>Bad Data =</i>	The rCHDB tuple's LDSF time slot (rCHDB.lsf ts).
<i>Good Data =</i>	The time slot from PCdsc3ckts[] (PCdsc3ckts[unit].pcdsc3ts[chann]).
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump 1 =</i>	The rCHDB tuple.
<i>Dump 2 =</i>	The associated rITONECH tuple.

Table 31 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The <i>rep_f</i> field of the rCHDB tuple, which shows whether the F-bit scanning should be allowed or not, did not match the same value calculated by the audit for a DNU-S facility. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The <i>rep_f</i> field of rCHDB tuple is corrected with the value calculated by audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The unit type, <i>utype</i> field of rCDBCOM tuple linked to rCHDB tuple. This should be SMDNU S.
<i>Bad Data =</i>	The <i>rep_f</i> field of the rCHDB tuple.
<i>Good Data =</i>	The value of the <i>rep_f</i> field calculated by audit.
<i>Logical Index =</i>	Key of the rCHDB tuple, <i>chnum</i> .
<i>Dump 1 =</i>	The rCHDB tuple in error.
<i>Dump 2 =</i>	The rCDBCOM tuple linked to rCHDB tuple.
<i>Dump 3 =</i>	The rSMEST tuple of the STE circuit, if accessed by audit.

Table 32 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rCHDB has nonzero linkage in one of its linkage blocks.
Possible Error Effect:	Some resource is linked to a rCHDB on the idle link list and may be unusable by call processing.
Corrective Action Taken:	The audit initializes the rCHDB and places it on the idle list. As part of initialization, it zeros the linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCHDB tuple.
<i>Bad Data =</i>	Bad linkage contents.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Key of the rCHDB tuple.
<i>Dump =</i>	rCHDB tuple.

CHDB-C

Software Release: 5E13 and later

1. CHANNEL DATA BLOCK (CHDB) AUDIT

The channel data block (CHDB) is an abstraction of the peripheral side channel of the network. The channel data block is used to allocate channels to lines for call processing. A line is assigned a channel when the customer goes off-hook. Trunk circuits have dedicated channels. If there is an error in the channel data block data associated with an active call, it will interfere with successful call processing. An error in the channel data block data associated with an idle trunk channel may result in lost supervision.

Time available on the link from the switching module (SM) to a remote switching module (RSM) is divided into time slots. Each time slot carries a voice signal, which is composed of 16 bit pulses. Eight of the bits contain the encoding of the voice signal, and eight bits are used for signaling. One of these eight signaling bits, the E signaling bit, confirms the connection between the originating and terminating processes. The possible bit values are continuity and discontinuity.

The channel data block audit (CHDB) ensures the integrity of the channel data block relation (RLCHDB). The audit performs validity checks on the head cells in the relation and on the rCHDB tuples.

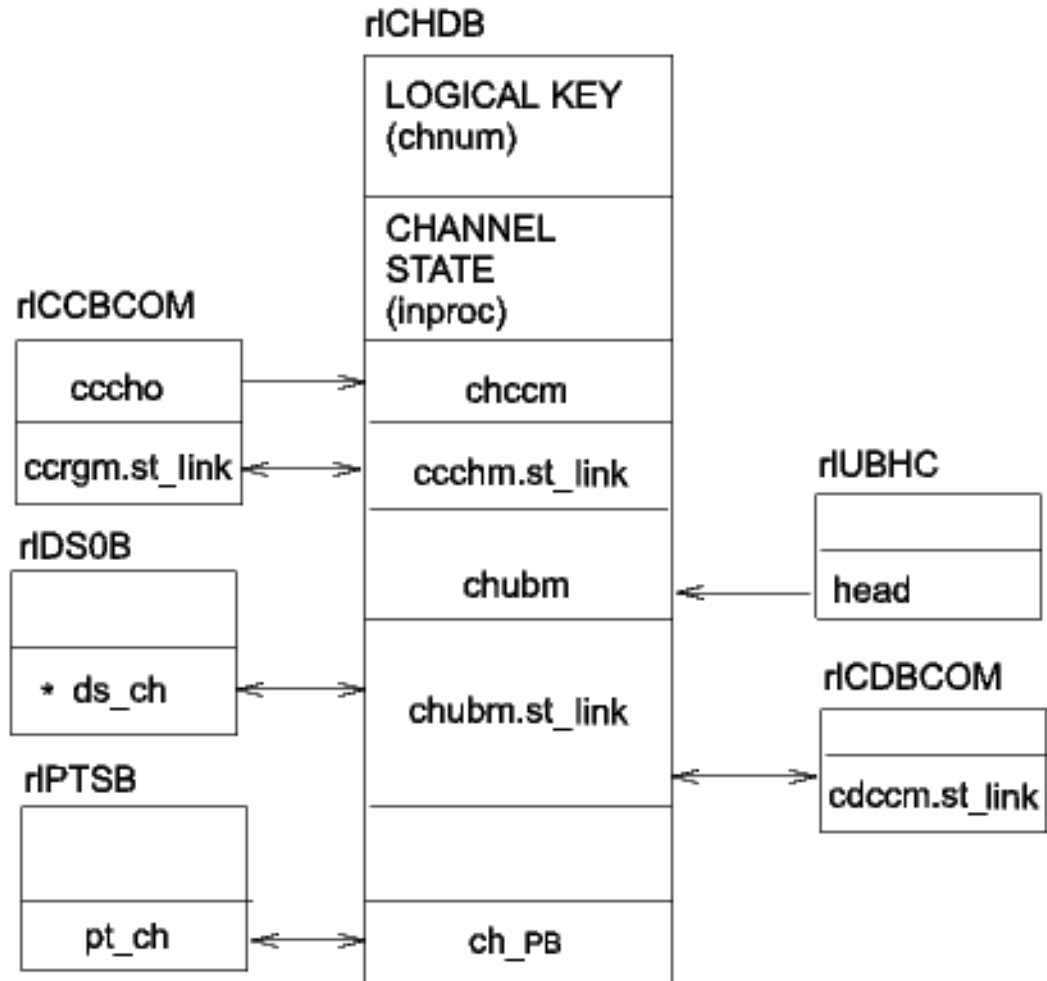
2. DATA STRUCTURES

The data structures checked by the CHDB audit are the channel data block relation (RLCHDB), the unconcentrated buffer head cell relation (RLUBHC), the channel control block common area relation (RLCCBCOM), the circuit data block common area relation (RLCDBCOM) and the PIDB time slot block relation (RLPTSB). The following data structures are used for SM-2000 processors only:

ldsf_ts = A field in the rCHDB relation that represents the LDSF TSI time slot associated with the CHDB.

rLNK_USE = The TSI link type relation structure.

Figure 1 illustrates the layout of rCHDB and the linkages associated with it.



SM

$\longleftrightarrow$ = One-to-One Linkage
 $\longrightarrow$ = Owner-Member Linkage

* = This linkage is used for IDCU

Figure 1 (CHDB-1) rCHDB Linkages

3. AUDIT CHECKS

To verify the linkages associated with the channel data blocks, the CHDB audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the DRLINK driver executes a semantic check function that performs the following checks based on the state of the channel.

- The audit checks the channel state (inproc) field against the channel data block linkages. Idle channel data

blocks should be owned by the channel data block idle list head cell or else the channel data blocks will be unavailable for call processing. An idle channel data block should have zero linkages except for those channel blocks dedicated to trunks. If an idle channel block has linkages, other resources cannot be used for call processing. An idle channel data block used for a trunk application should have a valid linkage to the trunk circuit data block. Otherwise, the trunk circuit cannot be used by call processing.

- If the inproc value indicates a special idle state for a given trunk type, the audit verifies the following:
 - The rICHDB tuple is properly linked to the channel data block idle list.
 - The rICHDB tuple is marked busy in the channel allocation maps.
 - The rICHDB tuple is properly linked to a circuit data block (trunk circuits).
- If an idle rICHDB tuple is associated with an HSM T1 facility, the audit verifies that the E-bit discontinuity is sent to the RSM.
- Any channel data block (except RSM control time slots in HSMs) associated with an idle channel (whether dedicated to a trunk or available for connection to a line) should be receiving from the time slot 0 (idle code) to prevent the signal of another telephone call from being transmitted to more than one destination. The audit performs a hardware read to determine which time slot is connected to the channel.
- If the inproc value indicates that the channel data block is in a stable busy state, the audit verifies the following:
 - The rICHDB tuple is properly linked to an rCCBCOM tuple owned by a terminal process.
 - The rICHDB tuple is properly linked to an rUBHC tuple.
 - The rICHDB tuple is marked busy in the channel allocation maps.
 - The rICHDB tuple is properly linked to rCDBCOM tuple or rPTSB tuple.
- If a stable rICHDB tuple is associated with an HSM T1 facility, the audit verifies that E-bit changes are being scanned. The audit also verifies that scanning detects continuity unless the tuple is on a hit-timing list.
- Sometimes the inproc value indicates that the state of the channel is expected to change soon. This situation usually occurs during transient states such as trunk signaling, digit collection for lines and trunks, and flash and hit timing for lines and trunks. If the inproc value indicates this case, the audit verifies the following:
 - The rICHDB tuple is marked busy in the channel allocation maps.
 - The rICHDB tuple is not in a transient state for more than five minutes.
- A few channels are used for special system purposes, such as audible ringing tone. If a transmission test facility (TTF) is used, three channels are dedicated to it. In this case, the inproc value indicates that the PC subsystem should ignore supervision of the channels, and the audit verifies the following:
 - The rICHDB tuples are properly linked to a special list.
 - The rICHDB tuples are marked busy in the channel allocation maps.
- The audit checks the four_wire bit in the rICHDB tuple. The four_wire bit in the rICHDB should only be set if the trunk using this time slot is 4-WIRE trunk or a precedence and preemption trunk.

- To save the call on hold during the selective initialization, the CHDB audit checks the path state in the PHDB. If the path is open and the inproc state is stable, then the call on hold will be saved.
- For TSIU to PSU nailed up channel in the idle state, verify that the CHDB tuple is properly linked to idle list (CCBCOM), linked to the logical time slot circuit (CDBCOM) and not linked to PTSB (since an idle channel has not been connected to the time slot in the PIDB).
- For TSIU to PSU nailed up channel in the stable state, verify that the CHDB tuple is properly linked to an active list (CCBCOM), linked to the logical time slot circuit (CDBCOM) and also linked to PTSB (since the channel has been connected to the time slot in the PIDB).
- The following checks are made only in the SM-2000 environment. For DNU related circuits, the audit checks that there is an associated rIPLI_LNK tuple. It also checks whether the scanning of the F bits should be allowed or not.

4. ERROR HANDLING

If any errors are found, the channel data block is reinitialized and placed on the idle list, unless the channel is dedicated to a trunk. Stuck calls are terminated.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BSYPSU

Error Code:	BSYPSU
Error Description:	The inproc attribute indicates that the rICHDB tuple is being used for a busy TSIU to PSU nailed up channel. However the linkages are not correct for this busy state.
Possible Error Effect:	The rICHDB tuple could be incorrectly set up for a TSIU to PSU nailed up channel. This will cause the nailed up channel listening to a different time slot. When the time slot is accessed by the call, assert may occur.
Corrective Action Taken:	The audit recovers the rICHDB to the idle state and attempt to preserve the linkage to rCDBCOM if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of rICHDB tuple.
<i>Bad Data =</i>	Valid values are: 1 = Not linked to rCCBCOM active list. 2 = Not linked to rCDBCOM. 3 = Not linked to rPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	The rICHDB tuple.

Table 2 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	The rICHDB tuple associated with a system tone failed a point-to-point-back check against the rCDBCOM tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICHDB tuple will be recovered to the idle list.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Link to the rICDBCOM tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump 1</i> =	rICHDB tuple.
<i>Dump 2</i> =	rICDBCOM tuple.

Table 3 Error Code: CHDB_TONE_FIELD

Error Code:	CHDB_TONE_FIELD
Error Description:	A CHDB's <i>inproc</i> field was not in the PCTONES state but its tone field was found to contain something but PCTNULL. Only PCTONES-type CHDBs should have a non-NULL value in the tone field.
Possible Error Effect:	An illegal CHDB tuple was lying around. Indicative of a process not properly cleaning up after itself.
Corrective Action Taken:	The offending tuple's tone field was set to PCTNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Bad tone field value.
<i>Good Data</i> =	PCTNULL value.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 4 Error Code: CRAM_E_UPPERBITS

Error Code:	CRAM_E_UPPERBITS
Error Description:	The value in the TSIU3-specific bits of control RAM E (CRAM-E) is invalid for the current set of options enabled for this TSIU model 3.
Possible Error Effect:	Unpredictable effects can occur when an option is enabled and the associated (and needed) hardware is not equipped.
Corrective Action Taken:	The correct value of the options covered in this audit has been written into CRAM-E.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the faulty CHDB tuple.
<i>Bad Data</i> =	A composite of the errors found during the check of the TSIU3-specific bits in CRAM-E. Any bit=1 is in error.
<i>Good Data</i> =	The value of the global TSIU3 option indicator, Gltsopt. This register uses negative logic. 0 = Option enabled. 1 = Option disabled.
<i>Logical Index</i> =	The key of the faulty CHDB tuple.
<i>Dump 1</i> =	The faulty CHDB tuple.
<i>Dump 2</i> =	The entire structure of type tsCHNCONFIG. See file <code>hdr/smim/SMts_emctl.n</code> for definition of CRAM-E. See file <code>hdr/smim/SMts_cmd3.h</code> for definition of Gltsopt.

Table 5 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	The input processing code marked in the rICHDB tuple (rICHDB.inproc) was marked to a state of PCASI_FLSH, but the advanced service interface feature bit (rICHDB.proxy_use) was not set. The PCASI_FLSH state is used for advanced service interface trunk signaling. For this state, the <i>proxy_use</i> field in the rICHDB tuple should always be set (marked to DBYES).
Possible Error Effect:	Call failures from loss of channel data, and failures during port conditioning

	activities.
Corrective Action Taken:	The rICHDB tuple, and all associated channel data are recovered to their idle states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	The state of the <i>proxy use</i> bit (DBNO).
<i>Good Data</i> =	The correct state of the <i>proxy use</i> bit (DBYES).
<i>Logical Index</i> =	The key of the rICHDB tuple (rICHDB.chnum).
<i>Dump</i> =	The contents of the rICHDB tuple.

Table 6 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICHDB tuple that is associated with a system tone is not linked to the DBCADGEN head cell.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICHDB tuple will be recovered and linked back onto its idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Owner of the rICHDB tuple.
<i>Good Data</i> =	Value of DBCADGEN head cell.
<i>Logical Index</i> =	Index of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 7 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit could not read the rUNIT_CKT tuple for the SONET Termination Equipment (STE) circuit, which is a fourth level parent of the digital signal level 0 (DS0) circuit for a DNU-S facility or the virtual tributary (VT1.5) circuit which is the second level circuit for a DNU-S facility. Check the Error address field as to determine which circuit could not be read. The rICHDB tuple under audit represents this DS0 circuit and the DS0 is the first (0th) DS0 on the facility. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The rICHDB tuple in error is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = Could not read the VT1.5 circuit. 2 = Could not read the STE circuit.
<i>Bad Data</i> =	The circuit index of the VT1.5 or STE circuit that audit could not read.
<i>Good Data</i> =	If audit could not read the VT1.5 circuit, then the VT1.5 number else the number of the STE circuit.
<i>Logical Index</i> =	Key of the rICHDB tuple, <i>chnum</i> .
<i>Dump 1</i> =	The rICHDB tuple in error.
<i>Dump 2</i> =	The rICDBCOM tuple this rICHDB is linked to.

Table 8 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	A stable HSM rICHDB which should be scanning for E-bit discontinuity from the periphery is not.
Possible Error Effect:	The call may be stuck and never tear itself down since it will not see E-bit discontinuity.
Corrective Action Taken:	The rICHDB is initialized. As part of initialization, E-bit discontinuity is sent towards the periphery.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 9 Error Code: E_BITSEND

Error Code:	E_BITSEND
Error Description:	An idle HSM rCHDB associated with a T1 facility is not sending E-bit discontinuity toward the periphery.
Possible Error Effect:	Calls originating or terminating in an RSM will fail when they attempt to use this channel.
Corrective Action Taken:	The rCHDB is initialized. As part of initialization, E-bit discontinuity is sent towards the periphery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 10 Error Code: IDLE_CHDB

Error Code:	IDLE_CHDB
Error Description:	An idle rCHDB's associated channel is not set to the idle state in the TSI hardware.
Possible Error Effect:	The rCHDB's associated channel is introducing noise into the time slot.
Corrective Action Taken:	The audit initializes the rCHDB and places it on the idle list. As part of initialization, the audit forces the channel to receive from time slot 0 (idle code) and set all related TSI hardware to the idle state.
Manual Action Required:	Request the TSDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of rCHDB.
<i>Bad Data</i> =	Valid values are: 0 = Assignable channel. 1 = Nailed-up channel.
<i>Good Data</i> =	Time slot 0.
<i>Logical Index</i> =	Key of the rCHDB and channel (they have the same key).
<i>Dump</i> =	rCHDB pc_chn_config structure.

Table 11 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	An rCHDB whose state field indicates that it is idle does not have as its owner the rCHDB idle list head cell.
Possible Error Effect:	The rCHDB is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit places the rCHDB on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Actual rCHDB owner's physical key.
<i>Good Data</i> =	Expected rCHDB owner's physical key.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 12 Error Code: IDLPSU

Error Code:	IDLPSU
Error Description:	The inproc attribute indicates that the rCHDB tuple is being used for an idle

	TSIU to PSU nailed up channel. However the linkages are not correct for this idle state.
Possible Error Effect:	The rICHDB tuple could be incorrectly set up for a TSIU to PSU nailed up channel. This error will prevent the nailed up channel from being used by call processing or maintenance.
Corrective Action Taken:	The audit recovers the rICHDB to the idle state and attempts to preserve the linkage to rCDBCOM if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	Valid values are: 1 = Not linked to rCCBCOM idle list. 2 = Not linked to rCDBCOM. 3 = Linked to rIPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	The rICHDB tuple.

Table 13 Error Code: INV

Error Code:	INV
Error Description:	The <i>ccstrk</i> field indicates whether or not this channel is associated with a common channel signal (CCS) trunk. The <i>ccstrk</i> field (rICHDB.ccstrk) has been found to be incorrect.
Possible Error Effect:	Trunk offhook reports may be incorrect. The trunk may be incorrectly queued in the wrong unconcentrated buffer.
Corrective Action Taken:	The audit sets the bit to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	<i>ccstrk</i> field of the rICHDB tuple.
<i>Good Data =</i>	<i>lp_type</i> of the trunk.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	The rICHDB tuple.

Table 14 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	The <i>dcstrk</i> field indicates whether or not this channel is associated with a digital AMPS (D-AMPS) trunk. The <i>dcstrk</i> field (rICHDB.dcstrk) has been found to be incorrect.
Possible Error Effect:	Trunk offhook reports may be incorrect. The trunk may be incorrectly queued in the wrong unconcentrated buffer.
Corrective Action Taken:	The audit sets the bit to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	<i>dcstrk</i> field of the rICHDB tuple.
<i>Good Data =</i>	<i>lp_type</i> of the trunk.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	The rICHDB tuple.

Table 15 Error Code: LINKS

Error Code:	LINKS
Error Description:	The rICHDB tuple is linked to rCDBCOM tuple through its <i>chubm.st_link</i> field. If this field is not null, that is not DALNKNUL, then an rCDBCOM tuple is attached to it. This error comes out when audit tried to access this rCDBCOM tuple and failed. This error will occur in an SM-2000 environment only.

Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The rCHDB tuple in error is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The <i>cdccm.st_link</i> field of rCDBCOM tuple which shows the linkage to rCHDB tuple.
<i>Bad Data =</i>	The <i>chubm.st_link</i> field minus DBSFCDBCOM which indicates the logical key of the rCDBCOM tuple the rCHDB tuple is attached to and audit failed to access.
<i>Good Data =</i>	Key of the rCDBCOM tuple, <i>name</i> , that audit could not access.
<i>Logical Index =</i>	Key of the rCHDB tuple, <i>chnum</i> .
<i>Dump =</i>	The rCHDB tuple in error.

Table 16 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	This error code indicates an inconsistency between two or more of the following data structures: PCtsicn_ch[] = Time slot interchange (TSI) constant channel array. This array is indexed by rCHDB.cnst and contains either a channel number or PCNULLCHAN. PCtsicon[] = TSI Constant Bit Map array. The i'th bit indicates whether the channel in the i'th entry of PCtsicn_ch[] is busy (bit=0) or idle (bit=1). PCtsiap_ch[] = (not for SM-2000) TSI access port channel array. This array is indexed by rCHDB.acspt and contains either a channel number or PCNULLCHAN. PCtsiap[] = (not for SM-2000) TSI access port bit map array. The i'th bit indicates whether the channel in the i'th entry of PCtsiap_ch[] is busy (bit=0) or idle (bit=1). An rCHDB tuple = The tuple's constant channel number (rCHDB.cnst) is an index into PCtsicn_ch[]. Its access port field (rCHDB.acspt) is an index into PCtsiap_ch[]. The index number in this description corresponds to the "Logical Index" value. Based on the "Logical Index", one of the following items will apply: 1 = The index into the PCtsicn_ch[] containing the channel number is not equal to the <i>cnst</i> field in corresponding rCHDB tuple (rCHDB.cnst). 2 = The index into the PCtsiap_ch[] containing the channel number is not equal to the <i>acspt</i> field in corresponding rCHDB tuple (rCHDB.acspt). 3 = The channel number in PCtsicn_ch[rCHDB.cnst] is corrupted. 4 = The channel number in PCtsiap_ch[rCHDB.acspt] is corrupted.
Possible Error Effect:	May not be able to use Diagnostics.
Corrective Action Taken:	Corresponds to Error Description as follows: 1,3 = The audit sets the channel number in PCtsicn_ch[rCHDB.cnst] to PCNULLCHAN and sets the corresponding status bit in PCtsicon[] to idle. 2,4 = The audit sets the channel number in PCtsiap_ch[rCHDB.acspt] to PCNULLCHAN and sets the corresponding status bit in PCtsiap[] to idle.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to Error Description as follows: 1,3 = Address of PCtsicn_ch[] element. 2,4 = Address of PCtsiap_ch[] element.
<i>Bad Data =</i>	Corresponds to Error Description as follows: 1,3 = Channel number stored in PCtsicn_ch[]. 2,4 = Channel number stored in PCtsiap_ch[].
<i>Good Data =</i>	Index into PC channel array.
<i>Logical Index =</i>	1, 2, 3, or 4 as stated in the Error Description.
<i>Dump 1 =</i>	Corresponds to Error Description as follows: 1,3 = Element of PC bit array, PCtsicon[]. 2,4 = Element of PC bit array, PCtsiapt[].
<i>Dump 2 =</i>	Corresponds to Error Description as follows: 1,3 = Element of PC constant channel, PCtsicn_ch[]. 2,4 = Element of PC access port channel array, PCtsiapt[].
<i>Dump 3 =</i>	The rICHDB tuple.

Table 17 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	This error code indicates an inconsistency between two or more of the following data structures: PCtsicn_ch[] = Time slot interchange (TSI) constant channel array. Always, PCtsicn_ch[] = PCNULLCHAN. PCtsicon[] = TSI constant bit map array. Its 0'th bit of the first entry is marked busy (bit=0). PCtsiap_ch[] = (not for SM-2000) TSI access port channel array. Always, PCtsiap_ch[] = PCNULLCHAN. PCtsiapt[] = (not for SM-2000) TSI access port bit map array. Its 0'th bit of the first entry is marked busy (bit = 0). The index number in this description corresponds to the "Logical Index" value. Based on "Logical Index", one of the following items will apply: 1 = Either PCtsicn_ch[] is not equal to PCNULLCHAN, or the corresponding status bit in PCtsicon[] (the 0'th bit) is marked idle. 2 = Either PCtsiap_ch[] is not equal to PCNULLCHAN, or the corresponding status bit in PCtsiapt[] (the 0'th bit) is marked idle.
Possible Error Effect:	May not be able to use Diagnostics.
Corrective Action Taken:	Corresponds to Error Description as follows:

	1 = The audit sets PCtsicn_ch[] = PCNULLCHAN and the corresponding bit in constant bit map array (PCtsicon[]) to busy. 2 = The audit sets PCtsiap_ch[] = PCNULLCHAN and the corresponding bit in access port bit map array (PCtsiapt[]) to busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to Error Description as follows: 1 = Address of PCtsicn_ch[]. 2 = Address of PCtsiap_ch[].
<i>Bad Data =</i>	Corresponds to Error Description as follows: 1 = Channel number stored in PCtsicn_ch[]. 2 = Channel number stored in PCtsiap_ch[].
<i>Good Data =</i>	Corresponds to Error Description as follows: 1 = Status bit for this channel number (PCtsicon[]). 2 = Status bit for this channel number (PCtsiapt[]).
<i>Logical Index =</i>	1 or 2 as stated in the Error Description.
<i>Dump 1 =</i>	PCtsicon[].
<i>Dump 2 =</i>	PCtsiapt[].

Table 18 Error Code: NOSUPV

Error Code:	NOSUPV
Error Description:	The channel has supervision incorrectly turned off.
Possible Error Effect:	The associated port cannot originate or disconnect.
Corrective Action Taken:	The rICHDB tuple is reinitialized and any associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICHDB tuple.
<i>Dump =</i>	rICHDB tuple.

Table 19 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit could not read the corresponding rIPLI_LNK tuple for a rICHDB tuple related to a DNU_S circuit. The audit will delete the rICHDB tuple and report a manual action error because the entry in PCLNK2USE exists which indicates the static rILNK_USE also exists. The PCLNK2USE is an array that contains all the links and the associated usage. It is built off of data in both the RLLNK_USE and rIPLI_LNK tuples. Therefore, it must have existed at one point. The <i>tsi_link</i> field in the dump is the key of the missing rIPLI_LNK tuple. It should be noted that this error will only be seen in the SM-2000 environment.
Possible Error Effect:	DNU_S facilities may be degraded since some static data is missing.
Corrective Action Taken:	The rICHDB tuple is deleted.
Manual Action Required:	The missing rIPLI_LNK tuple should be inserted.
Dump Description:	
<i>Error Address =</i>	Address of the rICHDB tuple.
<i>Bad Data =</i>	The value of the <i>tsi_link</i> field.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 20 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The index number in this description corresponds to the good data value. So, based on good data, one of the following items will apply: 0 = The rCHDB tuple's constant channel number field (rCHDB.cnst) is not set to the upper bound (PCNUMCN). 1 = (not for SM-2000) The rCHDB tuple's access port field (rCHDB.acspt) is not set to the upper bound (PCNUMAP).
Possible Error Effect:	May not be able to run diagnostics that require the use of a access or constant TSI port.
Corrective Action Taken:	The constant channel number field (rCHDB.cnst) is set to the upper bound (PCNUMCN) and the access port field (rCHDB.acspt) is set to the upper bound (PCNUMAP).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Corresponds to Error Description as follows: 0 = rCHDB tuple's <i>cnst</i> field. 1 = rCHDB tuple's <i>acspt</i> field.
<i>Good Data</i> =	0 or 1 as stated in the Error Description.
<i>Logical Index</i> =	Key of the rCHDB tuple (rCHDB.chnum).
<i>Dump</i> =	rCHDB tuple.

Table 21 Error Code: PC_QUAR

Error Code:	PC_QUAR
Error Description:	The input processing code (rCHDB.inproc) indicates that Peripheral Control encountered signaling, timing, or signal processing errors associated with this rCHDB tuple. When peripheral control encounters one of these types of errors the input processing code is set to Peripheral Control quarantine (rCHDB.inproc == PCQUARAN) indicating that the rCHDB tuple should be recovered and reinitialized by the audit.
Possible Error Effect:	The state of the rCHDB tuple is unknown and this may cause line and trunks from originating.
Corrective Action Taken:	The rCHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Input processing doce (rCHDB.inproc = PCQUARAN)
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 22 Error Code: RSMCHAN

Error Code:	RSMCHAN
Error Description:	Invalid DB return when accessing RLRSMCHAN relation, possible ODD problem. This error will continue to print out until fixed by office personnel.
Possible Error Effect:	Relation cannot be accessed.
Corrective Action Taken:	None.
Manual Action Required:	If error repeats, investigate the ODD data.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.

<i>Bad Data</i> =	Bad DB return.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key used to access the RLRSMBCHAN relation.
<i>Dump</i> =	rlRSMCHAN buffer.

Table 23 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	A bad linkage has been found in a stable rICHDB. One of the following errors has occurred: • The rICHDB is linked to a head cell (owner's logical key<0). • The rICHDB is linked to a dummy tuple (owner's logical key=0). • The linkage is not being used (linkage physical key in the rICHDB=0). All of the links in the rICHDB should be active in a stable state.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit idles the rICHDB. This action terminates the active call that uses the rICHDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Bad linkage value.
<i>Good Data</i> =	Linkage identification.
	0 = rUBHC slave linkage. 1 = rCCBCOM sink linkage. 2 = rCDBCOM linkage. 3 = rCCBCOM source linkage.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	The rICHDB tuple.

Table 24 Error Code: STATE

Error Code:	STATE
Error Description:	Illegal state in the SP. The SP is not scanning for off-hooks for this channel. Either the SP ignore RAM is wrong or the rep.ag field in the rICHDB is wrong.
Possible Error Effect:	This channel cannot be used for phone calls.
Corrective Action Taken:	This rICHDB will be put on the idle list at which time scanning will be turned on for off-hooks.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Bitmap calculated by audit.
<i>Bad Data</i> =	SP ignore RAM.
<i>Good Data</i> =	Expected value.
<i>Logical Index</i> =	rICHDB channel number.
<i>Dump</i> =	rICHDB tuple.

Table 25 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rICHDB tuple has an invalid state indicator in the inproc field.
Possible Error Effect:	The state of the rICHDB is unknown.
Corrective Action Taken:	The rICHDB tuple is idled.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Bad state variable.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 26 Error Code: STATUS

Error Code:	STATUS
Error Description:	The <i>four_wire</i> bit of a rCHDB tuple has incorrectly been turned on or off. The <i>four_wire</i> bit should be set according to the service class in rFC_PORTTYP tuple.
Possible Error Effect:	The rCHDB is incorrectly set up for <i>four_wire</i> access. This will impact the origination and termination from/to 4-wire trunk.
Corrective Action Taken:	The CHDB audit sets the bit to a correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	The <i>four_wire</i> field of the rCHDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 27 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than five minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rCHDB tuple is idled and the call is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump</i> =	rCHDB tuple.

Table 28 Error Code: TONE_CHDB

Error Code:	TONE_CHDB
Error Description:	The rCHDB tuple that is associated with a system tone failed a point-to-point-back check with the rTONECH tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCHDB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCHDB tuple.
<i>Bad Data</i> =	Tone field of the rCHDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rCHDB tuple.
<i>Dump 1</i> =	rCHDB tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 29 Error Code: TRUNK_LINK

Error Code:	TRUNK_LINK
Error Description:	An idle rCHDB tuple, used for a trunk application, has an invalid rCDBCOM tuple linkage.
Possible Error Effect:	The trunk circuit is unusable.
Corrective Action Taken:	The audit initializes the rCHDB and places it on the idle list. When the CDBCOM audit runs, the audit links the rCHDB to the proper rCDBCOM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Bad linkage value.
<i>Bad Data</i> =	Linkage ID.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

Table 30 Error Code: TSOWNED

Error Code:	TSOWNED
Error Description:	An rICHDB tuple's local digital services function (LDSF) tone generator time slot does not match the time slot number held for it in write-protected data (rICHDB.lsf ts != PCdsc3ckts[unit].pcdsc3ts[chan]).
Possible Error Effect:	The system tone associated with the rICHDB tuple's channel is not usable.
Corrective Action Taken:	The rICHDB tuple is reinitialized. Any associated processes are purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The associated system tone's ldsuclnn (rITONECH.ldsuclnn). This data is used to derive the unit and channel numbers used to index into PCdsc3ckts[], pcdsc3ts[].
<i>Bad Data</i> =	The rICHDB tuple's LDSF time slot (rICHDB.lsf ts).
<i>Good Data</i> =	The time slot from PCdsc3ckts[] (PCdsc3ckts[unit].pcdsc3ts[chann]).
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump 1</i> =	The rICHDB tuple.
<i>Dump 2</i> =	The associated rITONECH tuple.

Table 31 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The <i>rep_f</i> field of the rICHDB tuple, which shows whether the F-bit scanning should be allowed or not, did not match the same value calculated by the audit for a DNU-S facility. This error will occur in an SM-2000 environment only.
Possible Error Effect:	The F-bit scanning could be falsely turned on or off.
Corrective Action Taken:	The <i>rep_f</i> field of rICHDB tuple is corrected with the value calculated by audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The unit type, <i>utype</i> field of rCDBCOM tuple linked to rICHDB tuple. This should be SMDNU S.
<i>Bad Data</i> =	The <i>rep_f</i> field of the rICHDB tuple.
<i>Good Data</i> =	The value of the <i>rep_f</i> field calculated by audit.
<i>Logical Index</i> =	Key of the rICHDB tuple, <i>chnum</i> .
<i>Dump 1</i> =	The rICHDB tuple in error.
<i>Dump 2</i> =	The rCDBCOM tuple linked to rICHDB tuple.
<i>Dump 3</i> =	The rSMEST tuple of the STE circuit, if accessed by audit.

Table 32 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rICHDB has nonzero linkage in one of its linkage blocks.
Possible Error Effect:	Some resource is linked to a rICHDB on the idle link list and may be unusable by call processing.
Corrective Action Taken:	The audit initializes the rICHDB and places it on the idle list. As part of initialization, it zeros the linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHDB tuple.
<i>Bad Data</i> =	Bad linkage contents.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of the rICHDB tuple.
<i>Dump</i> =	rICHDB tuple.

CINDEX7

Software Release: 5E10 and later

1. CIC INDEX LIST (CINDEX7) AUDIT

The CINDEX7 audit is responsible for verifying the linkage data and other semantic data which is stored in the RLCINDEX7 relation. This audit runs in both the switching module (SM) and the administrative module (AM). Refer to Figure 1 for a pictorial view of the linkages related to the rLCINDEX7 tuple.

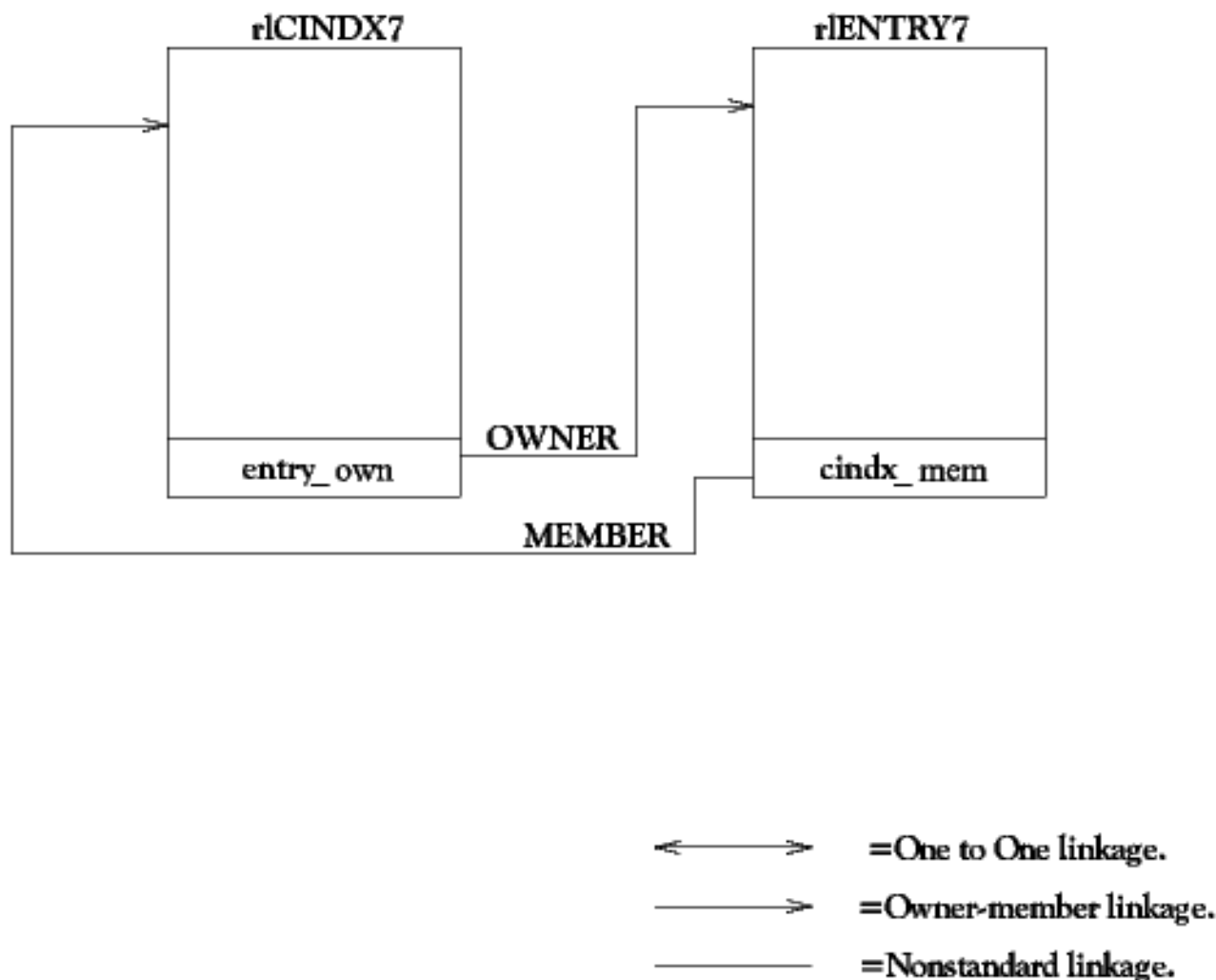


Figure 1 (CINDEX7-1) RLCINDEX7 Linkages

2. DATA STRUCTURES

The RLCINDEX7 relation is a dynamic relation that contains a (sorted) fragment of the entire PCI list. By maintaining the sorted key information about the head and tail entries in the fragment, it is possible to determine if the CIC being searched for may reside in any of the MCAH entries on that fragment of the PCI list. This provides the indexed list of all MCAH table entries (rLENTY7 tuples) for a given PCI. All unused rLCINDEX7 tuples are linked to a special "IDLE" rPCILST7 tuple.

3. AUDIT CHECKS

The CINDX7 audit uses the DRINASEK audit driver to step through each of the rCINDX7 tuples and perform the following checks.

BADDATAR	= Check to make sure that the length of an IDLE rCINDX7 tuple is always zero.
COUNTS	= Verify that the length field in rCINDX7 tuple matches the actual number of ENTRY7's linked to this CINDX7.
HEAD	= Verify that the head_cic field in rCINDX7 tuple is correct with respect to the the CINDX7 and ENTRY7.
MEMBAD	= Check to verify that when rCINDX7 tuple has no rENTRY7 tuples linked to it then linkages to PCILST must be NULL.
NONNULL	= Verify that the head_cic , tail_cic , and length fields are zeroed out in rCINDX7 tuple when there are no rENTRY7 tuples linked to it.
TWOLINKS	= Verify that the linkages between the CINDX7 and ENTRY7 are correct.
UNEQUAL	= Verify that the tail_cic field in rCINDX7 tuple is correct with respect to the CINDX7 and ENTRY7.
ZERO_LINK	= Check to make sure that rCINDX7 tuples linked to the IDLE owner list do not have anything linked to them.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	The length field in the rCINDX7 tuple was not set to zero for a CINDX7 which is on the IDLE list.
Possible Error Effect:	Incorrect allocation of MCAH table entry data.
Corrective Action Taken:	Reset the length field to zero for this tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDX7 tuple.
<i>Bad Data</i> =	Value of length in rCINDX7 tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The key of the rCINDX7 tuple (cindx).
<i>Dump</i> =	Contents of the rCINDX7 tuple being audited.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The length field in the rCINDX7 tuple did not match the total number of ENTRY7's linked to it.
Possible Error Effect:	Incorrect allocation of MCAH table entry data.
Corrective Action Taken:	Reset the length field to reflect the current number of ENTRY7's linked to the CINDX7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDX7 tuple.
<i>Bad Data</i> =	Value of length in rCINDX7 tuple.

<i>Good Data</i> =	Calculated value of length using ENTRY7's.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump</i> =	Contents of the rCINDEX7 tuple being audited.

Table 3 Error Code: HEAD

Error Code:	HEAD
Error Description:	There are three cases where this error could occur. The first is that the head_cic field in the CINDEX7 tuple does not match the CIC value of the first ENTRY7 tuple linked to this CINDEX7. The second is that the head_cic field in this CINDEX7 is greater than the tail_cic in this CINDEX7. The third is that the head_cic of this CINDEX7 is less than the tail_cic of the previous CINDEX7.
Possible Error Effect:	Incorrect allocation of MCAH table entry data.
Corrective Action Taken:	For the first case, reset the head_cic field to the CIC of the first ENTRY7 linked to this CINDEX7. For the other two cases, the CINDEX7 tuple will be recovered to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDEX7 tuple.
<i>Bad Data</i> =	Value of head_cic for this rCINDEX7 tuple.
<i>Good Data</i> =	For the first case, this is the value of the cic field in the first ENTRY7 linked to this CINDEX7. For the second case, this is the tail_cic field value of this CINDEX7. For the third case, this is the tail_cic field of the previous CINDEX7.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump 1</i> =	Contents of the rCINDEX7 tuple being audited.
<i>Dump 2</i> =	For the first case, contents of the first rENTRY7 tuple linked to this CINDEX7. For the third case, contents of the previous CINDEX7 tuple.

Table 4 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	A CINDEX7 tuple has no ENTRY7's linked to it but there are other CINDEX7's linked to this one.
Possible Error Effect:	Incorrect allocation of MCAH table entry data.
Corrective Action Taken:	Recover this CINDEX7 tuple to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDEX7 tuple.
<i>Bad Data</i> =	Value of the next CINDEX7 linked to this one.
<i>Good Data</i> =	Value of the previous CINDEX7 linked to this one.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump</i> =	Contents of the rCINDEX7 tuple being audited.

Table 5 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The values for head_cic , tail_cic , and length should be set to a NULL value since this CINDEX7 is not on the IDLE list and there are no ENTRY7's linked to it. One of these fields was set to something other than the NULL value.
Possible Error Effect:	Incorrect values could be analyzed for MCAH entries.
Corrective Action Taken:	Set the head_cic and tail_cic to CCNULCIC and the length to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value of head_cic in rCINDEX7 tuple.
<i>Bad Data</i> =	Value of tail_cic in rCINDEX7 tuple.
<i>Good Data</i> =	Value of length in rCINDEX7 tuple.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump</i> =	Contents of the rCINDEX7 tuple being audited.

Table 6 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	The set member linkages between the CINDEX7 and the associated ENTRY7

	were found to be incorrect.
Possible Error Effect:	Errors in traversal of data.
Corrective Action Taken:	Related audits will run to clean up any other linkage problems which could exist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDEX7 tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump</i> =	Contents of the rCINDEX7 tuple being audited.

Table 7 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The tail_cic field in the rCINDEX7 tuple does not equal the CIC in the "tail" associated ENTRY7 or the tail_cic field in the CINDEX7 tuple is greater than the head_cic field in the next CINDEX7 tuple.
Possible Error Effect:	Incorrect allocation of MCAH table entry data.
Corrective Action Taken:	Depending upon the error, the audit will either set the tail_cic field to match that of the "tail" ENTRY7 linked to the CINDEX7 or recover the CINDEX7 tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDEX7 tuple.
<i>Bad Data</i> =	Value of tail_cic in rCINDEX7 tuple.
<i>Good Data</i> =	Either the CIC of the tail ENTRY7 linked off of CINDEX7 or the head_cic field of the next CINDEX7.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump 1</i> =	Contents of the rCINDEX7 tuple being audited.
<i>Dump 2</i> =	Either the contents of the "tail" rENTRY7 tuple linked to this CINDEX7 or the contents of the next rCINDEX7 tuple in link with this CINDEX7.

Table 8 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An IDLE CINDEX7 has a non-zero value in one of its linkage fields.
Possible Error Effect:	Resources are not made available to users.
Corrective Action Taken:	Recover this CINDEX7 tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rCINDEX7 tuple.
<i>Bad Data</i> =	Return value from zero linkage check.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rCINDEX7 tuple (cindx).
<i>Dump</i> =	Contents of the rCINDEX7 tuple being audited.

9. CK

CKTDATA-A

Software Release: 5E10 only

1. CIRCUIT DATA (CKTDATA) AUDIT

The circuit data audit (CKTDATA) verifies that the information within each of these structures is correct and that the data is consistent when compared with the other data structures. It also performs static-to-dynamic existence checks for all data to find lost dynamic tuples. The CKTDATA audit has seven phases. The first phase calls the port linkage area audit (PORTLA) to check for the existence of trunk ports. The second phase checks the dynamic circuit data. The third phase checks the maintenance status of circuits. The fourth phase checks the consistency among the RLPORTLA, RLCDBCOM, and RLSMEST relations. The fifth phase checks the static-to-dynamic existence of PIDB time slot block relation (RLPTSB) tuples created for integrated services line unit (ISLU) and packet switching unit (PSU) peripheral interface data buses (PIDBs). The sixth phase performs the static-to-dynamic existence checks between the static RLCKTDATA relation and the dynamic RLGSDb relation. The seventh phase performs integrity checks between the RLCKTDATA relation and the peripheral control channel allocation maps (PCchalloc[]).

This audit is part of the peripheral control audits subgroup. It runs at routine level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The CKTDATA audit uses the circuit data relation (RLCKTDATA), a static relation, to verify information in the following dynamic data structures in the SMs: the peripheral control channel allocation map (PCchalloc[]), the port linkage area relation (RLPORTLA), the circuit data block common area relation (RLCDBCOM), the group summary data block relation (RLGSDb), the automatic queue relation (RLAUTOQ), the switch maintenance cross reference relation (RLSMXRF) and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these data structures.

3. AUDIT CHECKS

The circuit data audit uses the static driver (DRSTATIC) to step through the write-protected RLCKTDATA and retrieve tuples. The driver first determines which circuits exist and what data is available about them. It then calls the appropriate phase of the audit to perform the following checks.

Audit Phase 1 = The PORTLA audit phase of this audit uses the RLPORTLA relation to check the trunk ports. Although both trunk and line ports have tuples in RLPORTLA, this phase of the CKTDATA audit checks only trunk ports. Each trunk port has a tuple in RLPORTLA that describes its status and associations. Refer to the PORTLA audit description for more information.

The CKTDATA audit calls the AUdoportla function to perform all of the internal checks on one portla. Two additional sets of service values are required, FC_TCI_KT and FC_TCI_TT.

The CKTDATA audit also includes the following functions as part of its vertical checks:

- AUgetpar() is called to calculate the parent circuit(s) to replace the OSfap() call in AUsmest(). This allows the second pass of the audit to call the same function to calculate parents.
- OSfap() call for vertical checks is moved into the function AUvertch(). This will allow the second pass of the audit to call the same function to perform vertical checks.
- Initialization of the vertical check data is part of the audit entry function AUcktdata(). A flag is initialized indicating no need for a second pass. If at some point during the audit it is determined that a second pass is needed, the flag will be set to true. Also a flag indicating

which pass the audit is executing in is initialized.

- When the DRSTATIC audit drivers has stepped through all of the CKTDATA tuples, the function AUdo2pass will be invoked from AUcktdata() to step through all vertical recovery candidates identified during the first pass of the audit.

Audit Phase 2

= The dynamic circuit data phase of the audit uses the rICDBCOM relation to perform the following checks:

- The audit checks all point-to/point-back linkages to rICDBCOM.
- The audit verifies that in-service circuits are on appropriate idle lists. In-service circuits not on idle lists cannot be used for call processing.
- The audit verifies that stable rICDBCOM tuples have all required linkages.
- The audit verifies that idle rICDBCOM tuples do not have any linkages except idle list linkages.

See also the circuit data block common area audit (CDBCOM).

Audit Phase 3

= The maintenance status phase of the audit uses the rSMEST relation to perform the following checks:

- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA or PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (rSMEST), the switch maintenance cross-reference relation (rSMXRF), or the automatic queue relation (rAUTOQ).
- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA or PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.

- The audit performs a child-to-parent (vertical) consistency check to ensure that subtending circuits are in the proper base state (ACT, OOS, and so forth) with respect to the parent.

NOTE: Only the ISTF peripheral unit is allowed to perform the vertical check.

- The audit performs a horizontal check for circuits that have a mate circuit to ensure consistency. This is for circuits that are configured ACT/STBY or ACT MAJOR/ACT MINOR.

NOTE: Only the following circuits are allowed to perform the horizontal check:

ISLU CC
ISLU CD
PSUCOM
RISLU RRCLK

- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (rSMEST), the switch maintenance cross-reference relation (rSMXRF), or the automatic queue relation (rIAUTOQ).
- At the end of all the maintenance related checks, do the carrier group alarm (CGA) states checks. The units on which CGA checks are done are:
 - DFI2.
 - Digital networking unit-synchronous optical network (SONET) (DNU-S).

See also the switch maintenance equipment status table audit (SMEST).

- Audit Phase 4** = The consistency check phase of the audit verifies that the maintenance status for a given circuit (in-service or out-of-service) is consistent throughout the rIORTLA, rICDBCOM, and rSMEST relations.
- Audit Phase 5** = After all consistency checks are performed on the circuit, the audit checks to see if the circuit has dynamic rIPTSB tuples associated with it. Only PIDB types for ISLU, RISLU, and PSU have rIPTSBs. For these circuit types, the audit verifies that there are no lost rIPTSB tuples.
- Audit Phase 6** = After all consistency checks are performed on the circuit, the audit then insures that dynamic rIGSDB tuples exist. The rIGSDB tuples will exist for grid (PCGRID), cost reduced grid (PCLCGRID), line group controller (PCLGC), SLC[®] 96 shelf (PCS96SHF), and SLC[®] 96 bank (PCS96BNK) circuits that are marked equipped in the rICKTDATA tuple.
- Audit Phase 7** = At the completion of the audit checks will be made to insure the integrity of the usage counts (usecnt), allocation counts (allocnt), and (R)ISLU allocation counts (islu_alloc) in the peripheral control channel allocation map (PCchalloc[]).

During the running of the audit for every circuit in the RLCKTDATA relation the channel assignment (chnasn) field is examined. If the channel assignment is associated with an

allocable channel (PCALLOCHAN) the usage and allocation counts are pegged in a local copy of the map (AUdbch_cnt[]). However, if this channel is an allocable channel and is associated with a RISLU/ISLU cut-through PIDB the RISLU/ISLU allocation count (islu_alloc) will be pegged in addition to the usage (usecnt) and allocation (allocnt) counts. If the channel assignment is associated with a nailed-up channel (PCINITCHAN) only the usage count is pegged in the local copy of the map. At the completion of the audit the local copy of the channel allocation map (AUdbch_cnt[]) is compared to the peripheral control channel allocation map (PCchalloc[]).

4. ERROR HANDLING

If any dynamic tuples are lost, the audit will report an error and reinsert them.

If an inconsistency was found between the local copy of the channel allocation map (AUdbch_cnt[]) and the peripheral control channel allocation map (PCchalloc[]), the peripheral control channel allocation map (PCchalloc[]) is updated to be consistent with the local copy of the map (AUdbch_cnt[]).

If any inconsistencies are found, the rPORTLA and rCDBCOM tuples are zeroed and placed on the appropriate idle lists.

Units allowed to use the MRA recovery queue will recover the rSMEST to a safe, consistent state and (if MRA restorable) mark the rSMEST tuple SMAUOOS. The units will interface with SMIM to load the circuit on the MRA queue, which will result in an MRA restoration of the circuit to get the hardware and all software back to a consistent state. If the rSMEST tuple for the circuit/parent/mate is already in the MRA recovery queue (marked SMAUOOS) due to previous audit recovery, the audit skips further recovery. Also, for circuits having rCDBCOM tuples, the CDBCOM is placed on the OOS list to keep call processing from accessing the circuit while on the MRA recovery queue.

Circuits that fail the vertical check will be recovered to the correct state. Circuits that fail the horizontal check will be recovered using the MRA queue.

If a linkage between an rCDBCOM and an rCHDB tuple should exist but does not, an rCHDB tuple and a channel are allocated and linked to that rCDBCOM.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The bit_access field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of incorrect entry in CMrcldata.
<i>Bad Data =</i>	CMrcldata.hdw[].bit_access.
<i>Good Data =</i>	Correct access type based on the unit type 8 for DLTU, 16 for DLTU2.
<i>Logical Index =</i>	RCL circuit name.
<i>Dump =</i>	CMrcldata entry.

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An inconsistency has been detected between the dynamic data rSMEST CGA alarm fields and the actual state of the DFI hardware.
Possible Error Effect:	Alarms raised on DFI circuitry are not being served.
Corrective Action Taken:	Dynamic data is updated to agree with the DFI hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Bit map containing SMEST alarm fields: 0x4 = Red alarm. 0x2 = Yellow alarm. 0x1 = AIS alarm.
<i>Good Data =</i>	Bit map read from DFI hardware alarm: 0x4 = Red alarm. 0x2 = Yellow alarm. 0x1 = AIS alarm.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST.

Table 3 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	An rCDBCOM tuple exists for a circuit that is unequipped.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rCDBCOM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCDBCOM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCDBCOM tuple.
<i>Dump =</i>	rCDBCOM tuple.

Table 4 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rSMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rSMEST tuple.

Table 5 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by

	recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 6 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	A circuit is marked as rSMEST.qual_2 = SMAUOOS but either is not on the maintenance request administrator's (MRA's) queue, or its parent is not on the queue.
Possible Error Effect:	The actual maintenance state of a circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	Circuit is SMAUOOS but not in queue (4).
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of the circuit.
<i>Dump 2</i> =	rSMEST of the parent circuit.

Table 7 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A IDCU DS1 facility and its rSMEST.qual_2 is SMFAF, but either an alarm bit is not set, or the circuit is not on the MRA recovery queue.
Possible Error Effect:	The maintenance state of the circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The unittype field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .unittype.
<i>Good Data</i> =	Unit type according to rICKTDATA.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An RLI tuple in the SMmccstat[] array was found to be OOS RMVD AUTO. This is not a legal state for the RLI.
Possible Error Effect:	The RLI will not be usable.
Corrective Action Taken:	The RLI will be recovered to the correct state.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual 1.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 10 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The circuit has growth or special growth status but bas_stat is not equal to SMOOS.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	The bas_stat field is corrected to SMOOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 11 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state is active restricted but the circuit is not a PCS96FAC or a PCDCLCC.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 12 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The global data identifying RCL (remote cluster link) information has been corrupted. The picb field is incorrect when compared to rICKTDATA for the RCL.
Possible Error Effect:	Intercluster communication may be impaired.
Corrective Action Taken:	The PICB number stored in the CMrcldata tuple will be set to the PICB derived from the rICKTDATA base address.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CMrcldata.hdw entry that is in error.
<i>Bad Data</i> =	The PICB value of the bad entry.
<i>Good Data</i> =	The PICB value derived from the base address in the rICKTDATA tuple.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	NA.

Table 13 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The circuit is operational and either in service or unavailable but the qualifier, qual 1, is not SMSNUL.
Possible Error Effect:	Circuit may be erroneously denied service if bad value is unequipped or some similar value.
Corrective Action Taken:	Qualifier field is corrected to SMSNUL.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Value of SMSNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 14 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The circuit is operational and either in service or unavailable, but the qualifier, qual_2, is not SMSTNUL or SMPROOS with bas_stat SMACT.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier field is corrected to SMSTNUL. If the circuit is the remote clock, qual_2 is set to the hardware value of the remote clock mode: SMSMRCNORM SMRCFREE SMRCFAST SMRCHOLD
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_2 field.
<i>Good Data</i> =	Value of SMSTNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 15 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process and does not have a parent circuit which could be busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 16 Error Code: CAMA_IS

Error Code:	CAMA_IS
Error Description:	A CAMA-ONI keying trunk is incorrectly in service when its associated talking trunk is out-of-service.
Possible Error Effect:	The keying trunk will be hunted but it will not be possible to get control of the talking trunk. Per call failures will result whenever this CAMA-ONI trunk pair is hunted.
Corrective Action Taken:	The keying trunk is recovered to an OOS,FE,CAMA status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rPORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rPORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rPORTLA tuple for the talking trunk.

Table 17 Error Code: CAMA_OOS

Error Code:	CAMA_OOS
Error Description:	A CAMA-ONI keying trunk is incorrectly out-of-service (supposedly due to the talking trunk being out-of-service) when the associated talking trunk is in service.
Possible Error Effect:	The keying trunk will not be hunted and so this CAMA-ONI trunk pair will never be used by call processing.
Corrective Action Taken:	The keying trunk is recovered to an in service status provided its circuit status is in service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the keying trunk rIORTLA tuple.
<i>Bad Data =</i>	bsc_state for the keying trunk.
<i>Good Data =</i>	bsc_state for the talking trunk.
<i>Logical Index =</i>	Port name of the keying trunk.
<i>Dump 1 =</i>	rIORTLA tuple for the keying trunk.
<i>Dump 2 =</i>	rIORTLA tuple for the talking trunk.

Table 18 Error Code: CCDUPF

Error Code:	CCDUPF
Error Description:	The array SMccdupfail[] duplex failure status does not match the service groups' rSMEST tuple status. The error could be either that the array has the line unit as being duplex failing while at least one of the service groups' rSMEST tuples are active or the array indicates that the line unit is not duplex failing while both service groups rSMEST tuples are OOS.
Possible Error Effect:	The line unit will not have the correct IS/OOS status.
Corrective Action Taken:	The array will be corrected to reflect the correct status as per rSMEST tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	The status of the line unit as it appears in the SMdupfail[] array.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The circuit name of the service group.
<i>Dump 1 =</i>	rSMEST tuple for service group zero.
<i>Dump 2 =</i>	rSMEST tuple for the service group one.

Table 19 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The service status [in-service (0), out-of-service (1), or don't care (2)] of the rCDBCOM tuple disagrees with that of the rIORTLA tuple corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rCDBCOM tuple is placed on the idle link list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCDBCOM tuple
<i>Bad Data =</i>	Service status of rCDBCOM tuple.
<i>Good Data =</i>	Expected status.
<i>Logical Index =</i>	Internal name (rCDBCOM key).
<i>Dump =</i>	rCDBCOM tuple.

Table 20 Error Code: CGAQN_BITY

Error Code:	CGAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for carrier group alarm (CGA), yet it was not found to be on the TM queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rSMEST.tm_que field is recovered to DBNO.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.tm_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 21 Error Code: CGAQY_BITN

Error Code:	CGAQY_BITN
Error Description:	The circuit was found to be queued for carrier group alarm, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The rSMEST.tm_que field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.tm_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 22 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	A cdbcom that should have a dedicated linkage to a chdb is not linked to a chdb.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and reinitializes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	cdbcom.

Table 23 Error Code: CKTCUT

Error Code:	CKTCUT
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to store the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data</i> =	Value found in that address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump</i> =	The whole array AUs96inhp[].

Table 24 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter cluster communication may be impaired.

Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of start of CMrcldata.hdw.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 25 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter-cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CMrcldata[].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 26 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	A discrepancy was found between the actual usage of rCDBCOM and rIPTSB tuples and the count of INITCHANs and ALLOCHANs stored in the channel allocation map (PCchalloc[]) maintained by peripheral control. The contents of the usage counts (usecnt), allocation counts (allocnt), and ISLU allocation counts (islu_alloc) between the peripheral control channel allocation map (PCchalloc[]) and the audit's copy of the channel allocation map (AUdbch_cnt[]) are found to be inconsistent.
Possible Error Effect:	Peripheral control may have problems allocating channels to calls. Also, peripheral control will have problems relocating idle code and audible ring channels during hardware growth.
Corrective Action Taken:	The channel allocation map maintained (PCchalloc[]) by peripheral control will be updated from the local copy of the channel allocation (AUdbch_cnt[]) map kept by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The contents of the global flag (AUpc_upcknt) which indicates if peripheral control has updated their copy of the channel allocation map.
<i>Logical Index =</i>	The PIDB timeslot (dbchan) or index into the channel allocation map that was found in error.
<i>Dump 1 =</i>	The contents of the channel allocation map (PCchalloc[]) maintained by peripheral control.
<i>Dump 2 =</i>	The contents of the channel allocation map (AUdbch_cnt[]) maintained by Audits.

Table 27 Error Code: CURLNK

Error Code:	CURLNK
Error Description:	Audit found the rCDBCOM tuple with no rCHDB tuple attached to it for a DS0 circuit name in a DNU-S. The link field for the rCHDB tuple was found to have the null value, DALNKNUL.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The type field of the rICKTDATA tuple.
<i>Bad Data</i> =	The link field of rICDBCOM tuple that was found to be null, cdccm.st_link .
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name (rISMEST.unit_id).
<i>Dump 1</i> =	The rICDBCOM tuple.
<i>Dump 2</i> =	The rISMEST tuple.

Table 28 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICDBCOM is associated with a system tone but is not on the DBCADGEN idle list.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICDBCOM tuple.
<i>Bad Data</i> =	The owner of the rICDBCOM tuple.
<i>Good Data</i> =	DBCADGEN head cell value.
<i>Logical Index</i> =	Logical key of the rICDBCOM tuple.
<i>Dump</i> =	rICDBCOM tuple.

Table 29 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The max_ddl_size field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .max_ddl_size.
<i>Good Data</i> =	18 or 30.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 30 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	Inconsistencies have been detected in SMbufuse[] and/or SMquse[] global data used by the maintenance request administrator (MRA).
Possible Error Effect:	Possible problems scheduling MRA jobs.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from SMbufaud().
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 31 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	An rIDS0B tuple is missing from the database.
Possible Error Effect:	The line using that rIDS0B tuple will not have any call processing capability.
Corrective Action Taken:	The missing rIDS0B tuple will be inserted in the database and linked to the correct list. The list is determined by the in-service or out-of-service state of the rISMEST's tuple for the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDS0B tuple.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The key for the rIDS0B tuple.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	NA.

Table 32 Error Code: DS1PT_LOST

Error Code:	DS1PT_LOST
Error Description:	The audit detected that one of the rIDS1_PT tuple for a given IDCU DS1 facility is missing from the ODD. The rIDS1_PT tuple has two keys (first key is the rIDS1_PT.ds1name and the second key is the rIDS1_PT.key_slot). There should exist two tuple with the same first key but with different second key (zero and 12). The audit detected that one tuple was missing. Both tuples or none should be populated.
Possible Error Effect:	The nailup call processing using that tuple will not work.
Corrective Action Taken:	None.
Manual Action Required:	The craft should determine if the existing rIDS1_PT tuple should be deleted or the missing tuple should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS1_PT tuple.
<i>Bad Data</i> =	The return from the database read of the first half of the rIDS1_PT tuple where rIDS1_PT.key_slot=0. If it is a zero (GLSUCCESS), it means that the tuple existed, else the tuple was missing.
<i>Good Data</i> =	The return from the database read of the second half of the rIDS1_PT tuple where rIDS1_PT.key_slot=12. If it is zero, it means that the tuple existed, else the tuple was missing.
<i>Logical Index</i> =	Circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the rIDS1_PT tuple.

Table 33 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SMIM macros to read circuit name from one circuit to other circuit failed to read static relations, either RLCKTDATA or RLUNIT_CKT. Audit was trying to get the DS0 circuit name from any of the STE, STS1, VT1.5 or DS1 circuits for a DNU-S facility. Since this is a CKTDATA audit, the more likely missing tuple could be rLUNIT_CKT tuple for the DS0 circuit.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until either the appropriate rCKTDATA or rLUNIT_CKT tuple is inserted.
Manual Action Required:	If rCKTDATA tuple is missing, insert it with the key value dumped in the logical key field of audit message. For missing rLUNIT_CKT tuple, there are three keys. The unit key is dumped in the error address field. The utype key is dumped in the good data field. Audit does not have the information about the third key, ckt_indx, as this error was passed on to audit by SMIM code and that code calculated this circuit index value. Please contact any SMIM developer for this value.
Dump Description:	
<i>Error Address</i> =	The unit field of the rCKTDATA tuple.
<i>Bad Data</i> =	The circuit name received by audit. This should be GLNULCKT.
<i>Good Data</i> =	The unit type from the utype field of rCKTDATA tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rCKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 34 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	The circuit is operational and out-of-service, but bas_stat has an invalid value.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the

	circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad bas_stat field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 35 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	A BOOTSTRAP circuit (PCBTSR) on an SMP20 (switching module processor using <i>Motorola</i> ® 68020 processor), which is operational, has an rSMEST tuple which indicates that the circuit is not active.
Possible Error Effect:	Offline pump may fail for this switching module.
Corrective Action Taken:	The rSMEST bas_stat is set to SMACT, qual_1 is set to SMSNUL, and qual_2 is set to SMSTNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	Value of bas_stat in rSMEST.
<i>Good Data =</i>	The smest state value of SMACT=1.
<i>Logical Index =</i>	The circuit name.
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 36 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DNU-S facility and the carrier group alarm state. When the facility is marked active(ACT), there should be no carrier group alarm present.
Possible Error Effect:	The facility in the question may not be used, thus possible lost resource.
Corrective Action Taken:	The alarm fields are cleared to indicate that no alarm is present.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The red_alm field of the rSMEST tuple.
<i>Bad Data =</i>	The ylw_alm field of the rSMEST tuple.
<i>Good Data =</i>	The bas_stat field of the rSMEST tuple.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump =</i>	The rSMEST tuple.

Table 37 Error Code: KEY

Error Code:	KEY
Error Description:	A rSMEST tuple contains a bad logical key.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit reinitializes the rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST tuple.
<i>Bad Data =</i>	Logical key of rSMEST.
<i>Good Data =</i>	Logical key of associated rCDBCOM tuple.
<i>Logical Index =</i>	Logical key of rSMEST.
<i>Dump =</i>	rSMEST.

Table 38 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	No success in reading the tuple in RLSMEST to see if DCLU is in service.
Possible Error Effect:	No SLC® 96 call processing through this S-DFI.

Corrective Action Taken:	Request DFISHELF audit indirectly through T1USG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to DFI tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Equipment status table of the unit.

Table 39 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The rISMEST tuple indicates that the circuit corresponds to a live process, but it is not maintenance busy.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	PID of process which claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rISMEST tuple.
<i>Dump 2</i> =	rISMEST tuple for parent/mate (OPTIONAL).

Table 40 Error Code: LOST_GSDB

Error Code:	LOST_GSDB
Error Description:	A rIGSDB tuple has been found missing from the database for either a grid circuit (PCLCGRID, PCGRID) SLC [®] 96 bank circuit (PCS96BNK), a SLC [®] 96 shelf circuit (PCS96SHF), or a line group controller circuit (PCLGC) that is marked equipped in the rICKTDATA tuple (eqstat != SMUNEQ).
Possible Error Effect:	All line ports on these circuits will not be able to originate a call.
Corrective Action Taken:	The rIGSDB tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIGSDB tuple found in error.
<i>Bad Data</i> =	The circuit type found in the rICKTDATA tuple (type).
<i>Good Data</i> =	The circuit name found in the rICKTDATA tuple (name).
<i>Logical Index</i> =	The logical key of the rIGSDB tuple found missing in the database.
<i>Dump</i> =	The rICKTDATA tuple associated with this rIGSDB tuple.

Table 41 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	A circuit lacks its corresponding rIORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rIORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 42 Error Code: LOST_PTSTB

Error Code:	LOST_PTSTB
Error Description:	A PIDB circuit lacks one of its 32 corresponding rIPTSTB tuples.
Possible Error Effect:	The time slot resource is lost to the system.
Corrective Action Taken:	The rIPTSTB tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rIPTSB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Key of rIPTSB.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 43 Error Code: LOST_SMEST

Error Code:	LOST_SMEST
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Type of circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 44 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	A circuit lacks its corresponding rCDBCOM tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rCDBCOM tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type found.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 45 Error Code: LOSTTUP (1)

Error Code:	LOSTTUP (1)
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 46 Error Code: LOSTTUP (2)

Error Code:	LOSTTUP (2)
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 47 Error Code: LOSTTUP (3)

Error Code:	LOSTTUP (3)
Error Description:	The ttf tuple associated with this rCKTDATA tuple does not exist but should.
Possible Error Effect:	The circuit may be erroneously used.

Corrective Action Taken:	The ttf tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCKTDATA tuple.
<i>Bad Data =</i>	Associated ttf relation.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rCKTDATA circuit name.
<i>Dump =</i>	NA.

Table 48 Error Code: LOSTTUP (4)

Error Code:	LOSTTUP (4)
Error Description:	No success in reading the tuple from RLS96BASE to see if SG is in service.
Possible Error Effect:	No SLC® 96 call processing through this service group.
Corrective Action Taken:	Request BASE audit indirectly through PATHMAP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to DFI tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Equipment status table of the unit.

Table 49 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DFI facility and the carrier group alarm state. When there is a carrier group alarm, the facility should be marked out-of-service, auto, or removed but is not.
Possible Error Effect:	DFI facilities with CGAs will not be removed from service properly.
Corrective Action Taken:	The facility is removed and restored if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple in error.
<i>Bad Data =</i>	Facility basic state (rSMEST.bas_stat).
<i>Good Data =</i>	Out-of-service (4)
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump =</i>	rSMEST

Table 50 Error Code: MATE

Error Code:	MATE
Error Description:	The rSMEST tuple indicates that the mate of this circuit is in an invalid state. This is applicable to ACT/STBY type of circuits.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The circuit/mate is restored to correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Mate's bas_stat field in rSMEST tuple.
<i>Good Data =</i>	Circuit's bas_state field in rSMEST tuple.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	Circuit's unit ID.
<i>Dump 2 =</i>	Mate's unit ID.

Table 51 Error Code: MISSING

Error Code:	MISSING
Error Description:	Audit failed to read a rCDBCOM tuple for a DS0 circuit name for a DNU-S facility.
Possible Error Effect:	Possible failure of resource for this DS0 circuit.
Corrective Action Taken:	None. The audit will report this error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The type field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rICKTDATA tuple, name .
<i>Good Data</i> =	The circuit name for which audit could not get rCDBCOM tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 52 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The dynamic rPICB tuple associated with the static rICKTDATA tuple for this PIDB circuit is missing from the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PICB audit is scheduled to insert the missing rPICB tuple into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA in error.
<i>Bad Data</i> =	Key value of rPICB.
<i>Good Data</i> =	PICB value recalculated from circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rICKTDATA.
<i>Dump 2</i> =	rPICB.

Table 53 Error Code: MMP_IJA

Error Code:	MMP_IJA
Error Description:	State is allow but Service Request mask says inhibited.
Possible Error Effect:	No SLC [®] 96 call processing through this S_DFI.
Corrective Action Taken:	The service request mask will be corrected by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	The bit position that is wrong, start counting from 0.
<i>Dump</i> =	The equipment status table of the unit.

Table 54 Error Code: MP_HDW_IA

Error Code:	MP_HDW_IA
Error Description:	A rSMEST tuple has a bas_stat value only allowed for module processor equipment, and it isn't part of the module processor complex.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit will reinitialize the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Bad bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST tuple.

Table 55 Error Code: MRAQN_BITY

Error Code:	MRAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for recovery, yet it was not found to be on the queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.mra_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 56 Error Code: MRAQY_BITN

Error Code:	MRAQY_BITN
Error Description:	The circuit was found to be queued for restoration, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Value of rSMEST.mra_que field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Circuit's rSMEST tuple.

Table 57 Error Code: MSCU_IIA

Error Code:	MSCU_IIA
Error Description:	Since mate service request source register is used, bit 15 of the own SRSR must be allowed.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	This bit 15 of the "own" SRSR is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the service request mask.
<i>Bad Data =</i>	The value of the service request mask found to be wrong.
<i>Good Data =</i>	The correct value of the service request mask.
<i>Logical Index =</i>	15.
<i>Dump =</i>	The equipment status table of the unit.

Table 58 Error Code: NO_CHDB

Error Code:	NO_CHDB
Error Description:	The rCHDB tuple accessed by the audit is not the same tuple audit tried to get. It failed the key check, that is, the key of the rCHDB tuple, chnum is not equal to the link field in the rCDBCOM tuple, cdccm.st_link minus the shift for the rCHDB tuple, DBSFCHDB. The missing rCHDB tuple represented a DS0 circuit in a DNU-S.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The type field of the rCKTDATA tuple.
<i>Bad Data =</i>	The key field of rCHDB tuple, chnum .
<i>Good Data =</i>	The link field of rCDBCOM tuple, cdccm.st_link minus the shift value for rCHDB tuple, DBSFCHDB.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rCHDB tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 59 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous head cell (DBIDLMISC).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Idle link list physical key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the cdbcom.

<i>Dump 1</i> =	CKTDATA tuple.
<i>Dump 2</i> =	CDBCOM tuple.

Table 60 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rLSMEST tuple and in the F bits of the signaling information of the peripheral control and timing (PCT) links time slots. Audit found inconsistency between these informations when it compared it. The inconsistency could be in the form that the F bits says no alarm is present, but the rLSMEST of the circuit says there is an alarm present or the types of alarm present are not the same in both places. For exact information as to what happened, see the explanation in error address field in this section.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The audit will call TM function to put the circuit on the CGA recovery queue where its alarm information will be correctly applied.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field represents the type of error that occurred. Depending upon its value different explanations are given here. 1 = The F bits says that there is no alarm present on this circuit but one or more of the alarm bits are set in the rLSMEST tuple. 2 = The F bits says AIS alarm is present but the rLSMEST says either AIS alarm is not set or one of the other alarms, red or yellow, is set. 3 = The F bits says red alarm is present but the rLSMEST says either red alarm is not set or one of the other alarms, AIS or yellow, is set. 4 = The F bits says yellow alarm is present but the rLSMEST says either yellow alarm is not set or one of the other alarms, AIS or red, is set. 5 The audit received a value from TM that it is not expecting to receive.
<i>Bad Data</i> =	The local value of circuit type derived from the F bits, ckdtype .
<i>Good Data</i> =	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index</i> =	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1</i> =	The rLSMEST tuple.
<i>Dump 2</i> =	The rCDBCOM tuple.

Table 61 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rLSMEST tuple is not transient but its progress flag is set.
Possible Error Effect:	Audit transient timing may operate improperly.
Corrective Action Taken:	The progress flag is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rLSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rLSMEST tuple.

Table 62 Error Code: NOTSPARE

Error Code:	NOTSPARE
Error Description:	An rISMEST tuple has a bas_stat value only allowed for module processor and remote clock equipment.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The tuple is reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Value of rISMEST.bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISMEST for parent/mate (OPTIONAL).

Table 63 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their RLSMEST tuple and in the F bits of the signaling information of the PCT links time slots. The audit found an inconsistency when it compared the CGA alarm bits in the RLSMEST tuple against the bas_stat and qual_2 fields in that tuple. These two fields indicate the base state of the facility represented by the RLSMEST tuple. The alarm bits in the RLSMEST tuple (ais_alm, red_alm, or ylw_alm) should only be set if the base state indicates either out of service family equipment (OOS/FE) or out of service facility failure (OOS/FAF). In any other case the alarm bits should not be set.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The alarm bits (ais_alm , red_alm , or ylw_alm) are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The local value of circuit type derived from the F bits, ckdtype .
<i>Good Data</i> =	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index</i> =	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1</i> =	The rISMEST tuple.
<i>Dump 2</i> =	The rICDBCOM tuple.

Table 64 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	No rISLUPIDB tuple exists for this PCPIDBISLU or PCPIDBRISLU circuit.
Possible Error Effect:	This resource is lost to the system.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value used to read rISLUPIDB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 65 Error Code: ODDLOSTTUP (1)

Error Code:	ODDLOSTTUP (1)
Error Description:	A tuple in the static relation RLDSEQUIP is missing.
Possible Error Effect:	Full use of DSL equipment is impossible.
Corrective Action Taken:	None.
Manual Action Required:	A tuple should be inserted into RLDSEQUIP as appropriate for the missing PCRDTTMC or PCRDTEOC circuit.

Dump Description:	
<i>Error Address</i> =	Address of rIDSLEQUIP buffer.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	RLCKTDATA.

Table 66 Error Code: ODDLOSTTUP (2)

Error Code:	ODDLOSTTUP (2)
Error Description:	The CKTDATA audit is attempting to read a nonexistent rIDCKT2GP tuple. The key to the tuple is derived from the rCKTDATA tuple circuit name and circuit index.
Possible Error Effect:	Call processing cannot use the circuit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCKTDATA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rCKTDATA tuple.

Table 67 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The circuit is operational and out-of-service but qual_1 and qual_2 do not have compatible values.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 68 Error Code: PARENT

Error Code:	PARENT
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process nor is its parent circuit busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	Parent's rSMEST.

Table 69 Error Code: PCD

Error Code:	PCD
Error Description:	The audit was not able to read the rVPIDB_F tuple for a corresponding either the STE circuit or the STS-1 circuit for a DNU-S. Refer to good data field to determine for which type of circuit, STE or STS-1, audit could not read the

	rVPIDB_F tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report this error until the rVPID_F tuple is inserted.
Manual Action Required:	There are three key field for the rVPIDB_F tuple, unitttype , unitnumber and pct_link . Use the value of utype and unit from the rCKTDATA dump of this error for the first two keys of the rVPIDB_F tuple, unitttype and unitnumber respectively. Use the value of key field for the third key of the rVPIDB_F tuple, pct_link and insert this rVPIDB_F tuple.
Dump Description:	
<i>Error Address</i> =	The unit field of the rCKTDATA tuple.
<i>Bad Data</i> =	Contents of the DS0 circuit name calculated by audit if any.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The pct_link key of the rVPIDB_F tuple as calculated by audit from the given data.
<i>Dump 1</i> =	The rCKTDATA tuple, either for STE or STS-1.
<i>Dump 2</i> =	The rVPIDB_F tuple as returned by database read.

Table 70 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the rCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 71 Error Code: PICBKEY

Error Code:	PICBKEY
Error Description:	The static relation RLislupidb is missing a tuple.
Possible Error Effect:	The tuple is not available for access.
Corrective Action Taken:	None.
Manual Action Required:	The tuple should be inserted into the static relation RLISLUPIDB.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value.
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 72 Error Code: PID

Error Code:	PID
Error Description:	The PID in the tp_pid field is not OSnullpid nor a live PID and should be ignored.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The PID is set to OSnullpid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.

<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 73 Error Code: QUAL

Error Code:	QUAL
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but qual_1 indicates the circuit is growth or unequipped and not on the automatic or manual request queues.
Possible Error Effect:	The circuit may be erroneously processed by MRA or lost to call processing.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 74 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The circuit is operational and out-of-service, but qual_2 has an invalid value.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_2 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	Cross reference entry.
<i>Dump 3 =</i>	Queue entry.
<i>Dump 4 =</i>	riSMEST tuple under audit.
<i>Dump 5 =</i>	Parent of the riSMEST tuple under audit.

Table 75 Error Code: QUAL_EQPG

Error Code:	QUAL_EQPG
Error Description:	The circuit is growth or special growth, but qual_1 is not equal to SMGR or SMSGR.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qual_1 field is corrected to SMGR or SMSGR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 76 Error Code: QUE

Error Code:	QUE
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but the circuit does not have an entry in the queue specified by qual_1

	(auto or man).
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	Identity of queue in which the circuit was found.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 77 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	A reconfiguration request of a remote clock circuit was not successful.
Possible Error Effect:	Remote clocking may be impaired.
Corrective Action Taken:	All remote clock circuits generating this error, except the parent circuit, are marked out-of-service or active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Return value of the configuration routine.
<i>Good Data =</i>	Action performed on the circuit.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST of parent circuit.
<i>Dump 2 =</i>	rSMEST of subtending or mate side to a remote clock side.

Table 78 Error Code: READ

Error Code:	READ
Error Description:	Remote clock hardware can not be read at this time.
Possible Error Effect:	Can not determine active RCLK.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Value of mate's rSMEST.bas_stat field.
<i>Good Data =</i>	Value of circuit's rSMEST.bas_stat field.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST of circuit.
<i>Dump 2 =</i>	rSMEST of mate circuit.

Table 79 Error Code: RT2DS1_LOST (1)

Error Code:	RT2DS1_LOST (1)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCUs facilities.
Corrective Action Taken:	None.
Manual Action Required:	A tuple in RLRT2DS1 should be inserted as appropriate for the circuit identified in the associated audit dump.
Dump Description:	
<i>Error Address =</i>	Address of rRT2DS1 buffer.
<i>Bad Data =</i>	Return value from DB (DBNO_MATCH).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rCKTDATA.

Table 80 Error Code: RT2DS1_LOST (2)

Error Code:	RT2DS1_LOST (2)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCUs facilities.

Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIRT2DS1 buffer.
<i>Bad Data =</i>	Shelf number of RT circuit.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rCKTDATA.

Table 81 Error Code: SCNBAD

Error Code:	SCNBAD
Error Description:	An inconsistency has been detected in the value of the IDCUCOM register HWMLSCAN; either it is set when it should not be or vice versa. If the IDCUCOM is active, the bit should be set. If the IDCUCOM is stby, the bit should not be set.
Possible Error Effect:	The scanning state is incorrect for the IDCUCOM identified.
Corrective Action Taken:	The LSCAN bit is reset to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Value of hardware register read.
<i>Good Data =</i>	HWMLSCAN.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST.

Table 82 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Audit tried to read a tuple of the static relation, RLFAC_PROV, as part of the CGA checks on DNU-S, but failed to read it.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until the appropriate rFAC_PROV tuple is inserted.
Manual Action Required:	Insert a rFAC_PROV tuple with the key equal to the circuit name dumped in either the logical key or error address fields. The same value is dumped in these fields from different data structures.
Dump Description:	
<i>Error Address =</i>	The name field of the rCKTDATA tuple.
<i>Bad Data =</i>	The key of the rFAC_PROV tuple, cktname , which audit could not access.
<i>Good Data =</i>	The circuit type from the type field of rCKTDATA tuple.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 83 Error Code: SMEST

Error Code:	SMEST
Error Description:	The service status [in-service (0), out-of-service (1)] of the rSMEST tuple disagrees with that of other tuples corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name (or resources, if the name corresponds to a family of equipment) to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rSMEST tuple is forced idle and re-audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST tuple.
<i>Bad Data =</i>	Service status (in/out) of rSMEST tuple.
<i>Good Data =</i>	Expected status.
<i>Logical Index =</i>	Internal name (rSMEST key).
<i>Dump =</i>	rSMEST tuple.

Table 84 Error Code: SSTBAD

Error Code:	SSTBAD
Error Description:	Register contents do not match the contents of the IDCUCOM rSMEST and the LSI(s) RLCKTDATA.
Possible Error Effect:	The hardware state of the IDCU is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	Constructed mask indicating correct service state.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 85 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rSMEST tuple indicates that this circuit is idle, but it has an entry in one of the queues or the xrf table.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Cross reference entry.
<i>Dump 2</i> =	Queue entry.
<i>Dump 3</i> =	rSMEST tuple under audit.
<i>Dump 4</i> =	Parent of the rSMEST tuple under audit.

Table 86 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkusage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	linkusage (from cdbcom).
<i>Good Data</i> =	linkusage (expected).
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	Circuit type CDBCOM.

Table 87 Error Code: STATUS

Error Code:	STATUS
Error Description:	The rSMEST tuple indicates that this circuit is neither busy nor idle, thus the stat_act field is invalid.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 88 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rSMEST tuple has been stuck for several minutes in a transient state.
Possible Error Effect:	A process using the rSMEST tuple is stuck with allocated resources and some

	audit consistency checks are prevented.
Corrective Action Taken:	The rSMEST tuple is recovered to a non-transient state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST tuple.

Table 89 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	One or both RCLK circuits has been busy for at least 5 seconds under control of the GLPNRCLK process.
Possible Error Effect:	The circuit may be lost for further MRA requests.
Corrective Action Taken:	The RCLK circuits are further checked to verify RCLK configuration status and are recovered to a safe/consistent state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	MRA activity value rSMEST.stat_act.
<i>Good Data =</i>	DBIDLE
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST for busy RCLK.
<i>Dump 2 =</i>	rSMEST for mate RCLK.

Table 90 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rCDBCOM and its associated rTONECH tuple failed a point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCDBCOM tuple.
<i>Bad Data =</i>	Tone field in the rCDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rCDBCOM tuple.
<i>Dump 1 =</i>	rCDBCOM tuple.
<i>Dump 2 =</i>	rTONECH tuple.

Table 91 Error Code: TYPE

Error Code:	TYPE
Error Description:	The type of circuit that the CKTDATA audit is attempting to audit is either not present in its private circuit mapping array (AUcktreimap[]), or it is marked as invalid in the map.
Possible Error Effect:	The audit will continue to print the error each time it runs but will otherwise ignore the circuit. This is an audit error.
Corrective Action Taken:	None.
Manual Action Required:	Report the error with a failure report. Correcting this problem may require manual boot of the SM to restore the RLCKTDATA relation.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Circuit type.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 92 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The circuit is unequipped but the qualifier qual_1 is not SMUN.

Possible Error Effect:	The circuit is accessible but has no hardware, which will result in interrupts when it is used.
Corrective Action Taken:	Qualifier field is corrected to SMUN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual 1 field.
<i>Good Data =</i>	Value of SMUN.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 93 Error Code: VERTICAL

Error Code:	VERTICAL
Error Description:	The maintenance state of a circuit (rSMEST.bas_stat) is vertically inconsistent with the maintenance state of its "parent" circuit.
Possible Error Effect:	The circuit may not be available for use.
Corrective Action Taken:	The child's rSMEST tuple is made consistent with the parent's.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple for the child circuit.
<i>Bad Data =</i>	Child's rSMEST.bas_stat.
<i>Good Data =</i>	Number of parent circuits.
<i>Logical Index =</i>	Child circuit name.
<i>Dump 1 =</i>	The child's rSMEST tuple.
<i>Dump 2 =</i>	All the parent circuit rSMESTs (may be several).

Table 94 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rCDBCOM address.
<i>Bad Data =</i>	Actual idle list physical key.
<i>Good Data =</i>	Physical key of correct idle list.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	rCDBCOM tuple.

Table 95 Error Code: XRFPID

Error Code:	XRFPID
Error Description:	The rSMEST tuple indicates that this circuit has a live PID and the circuit is in xrf or one of the queues, but the PID doesn't match that in the queue.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	PID of process that claims association.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	xrf search results.
<i>Dump 2 =</i>	manq, autoq search results.
<i>Dump 3 =</i>	rSMEST.
<i>Dump 4 =</i>	Dump of parent's rSMEST tuple.

CKTDATA-B

Software Release: 5E11 only

1. CIRCUIT DATA (CKTDATA) AUDIT

The circuit data audit (CKTDATA) verifies that the information within each of these structures is correct and that the data is consistent when compared with the other data structures. It also performs static-to-dynamic existence checks for all data to find lost dynamic tuples. The CKTDATA audit has seven phases. The first phase calls the port linkage area audit (PORTLA) to check for the existence of trunk ports. The second phase checks the dynamic circuit data. The third phase checks the maintenance status of circuits. The fourth phase checks the consistency among the RLPORTLA, RLCDBCOM, and RLSMEST relations. The fifth phase checks the static-to-dynamic existence of PIDB time slot block relation (RLPTSDB) tuples created for integrated services line unit (ISLU) and packet switching unit (PSU) peripheral interface data buses (PIDBs). The sixth phase performs the static-to-dynamic existence checks between the static RLCKTDATA relation and the dynamic RLGSDDB relation. The seventh phase performs integrity checks between the RLCKTDATA relation and the peripheral control channel allocation maps (PCchalloc[]).

This audit is part of the peripheral control audits subgroup. It runs at routine level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The CKTDATA audit uses the circuit data relation (RLCKTDATA), a static relation, to verify information in the following dynamic data structures in the SMs: the peripheral control channel allocation map (PCchalloc[]), the port linkage area relation (RLPORTLA), the circuit data block common area relation (RLCDBCOM), the group summary data block relation (RLGSDDB), the automatic queue relation (RLAUTOQ), the switch maintenance cross reference relation (RLSMXRF) and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these data structures.

3. AUDIT CHECKS

The circuit data audit uses the static driver (DRSTATIC) to step through the write-protected RLCKTDATA and retrieve tuples. The driver first determines which circuits exist and what data is available about them. It then calls the appropriate phase of the audit to perform the following checks.

Audit Phase 1 = The PORTLA audit phase of this audit uses the rIPORTLA relation to check the trunk ports. Although both trunk and line ports have tuples in rIPORTLA, this phase of the CKTDATA audit checks only trunk ports. Each trunk port has a tuple in rIPORTLA that describes its status and associations. Refer to the PORTLA audit description for more information.

The CKTDATA audit calls the AUdoportla function to perform all of the internal checks on one portla. Two additional sets of service values are required, FC_TCI_KT and FC_TCI_TT.

The CKTDATA audit also includes the following functions as part of its vertical checks:

- AUgetpar() is called to calculate the parent circuit(s) to replace the OSfap() call in AUsmest(). This allows the second pass of the audit to call the same function to calculate parents.
- The OSfap() call for vertical checks is moved into the function AUvertch(). This will allow the second pass of the audit to call the same function to perform vertical checks.
- Initialization of the vertical check data is part of the audit entry function AUcktdata(). A flag is initialized indicating no need for a second pass. If at some point during the audit it is determined that a second pass is needed, the flag will be set to true. Also a flag indicating

which pass the audit is executing in is initialized.

- When the DRSTATIC audit drivers has stepped through all of the CKTDATA tuples, the function AUdo2pass will be invoked from AUcktdata() to step through all vertical recovery candidates identified during the first pass of the audit.

Audit Phase 2

= The dynamic circuit data phase of the audit uses the rICDBCOM relation to perform the following checks:

- The audit checks all point-to/point-back linkages to rICDBCOM.
- The audit verifies that in-service circuits are on appropriate idle lists. In-service circuits not on idle lists cannot be used for call processing.
- The audit verifies that stable rICDBCOM tuples have all required linkages.
- The audit verifies that idle rICDBCOM tuples do not have any linkages except idle list linkages.

See also the circuit data block common area audit (CDBCOM).

Audit Phase 3

= The maintenance status phase of the audit uses the RLSMEST relation to perform the following checks:

- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rISMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (rISMEST), the switch maintenance cross-reference relation (rISMXRF), or the automatic queue relation (rIAUTOQ).
- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.

- The audit performs a child-to-parent (vertical) consistency check to ensure that subtending circuits are in the proper base state (ACT, OOS, etc.) with respect to the parent.

NOTE: Only the ISTF peripheral unit is allowed to perform the vertical check.

- The audit performs a horizontal check for circuits that have a mate circuit to ensure consistency. This is for circuits that are configured ACT/STBY or ACT MAJOR/ACT MINOR.

NOTE: Only the following circuits are allowed to perform the horizontal check:

ISLU CC
ISLU CD
PSUCOM
RISLU RRCLK

- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in RLSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (RLSMEST), the switch maintenance cross-reference relation (RLSMXRF), or the automatic queue relation (RLAUTOQ).
- At the end of all the maintenance related checks, do the carrier group alarm (CGA) states checks. The units on which CGA checks are done are:
 - DFI2
 - Digital networking unit-synchronous optical network (SONET) (DNU-S).

See also the switch maintenance equipment status table audit (SMEST).

- Audit Phase 4** = The consistency check phase of the audit verifies that the maintenance status for a given circuit (in-service or out-of-service) is consistent throughout the RLPORTLA, RLCDBCOM, and RLSMEST relations.
- Audit Phase 5** = After all consistency checks are performed on the circuit, the audit checks to see if the circuit has dynamic rIPTSB tuples associated with it. Only PIDB types for ISLU, RISLU, and PSU have rIPTSBs. For these circuit types, the audit verifies that there are no lost rIPTSB tuples.
- Audit Phase 6** = After all consistency checks are performed on the circuit, the audit then insures that dynamic rIGSDB tuples exist. The rIGSDB tuples will exist for grid (PCGRID), cost reduced grid (PCLCGRID), line group controller (PCLGC), SLC[®] 96 shelf (PCS96SHF), and SLC[®] 96 bank (PCS96BNK) circuits that are marked equipped in the rLCKTDATA tuple.
- Audit Phase 7** = At the completion of the audit checks will be made to insure the integrity of the usage counts (usecnt), allocation counts (allocnt), and RISLU/ISLU allocation counts (islu_alloc) in the Peripheral Control channel allocation map (PCchalloc[]).

During the running of the audit for every circuit in the RLCKTDATA relation the channel assignment (chnasn) field is examined. If the channel assignment is associated with an

allocable channel (PCALLOCHAN) the usage and allocation counts are pegged in a local copy of the map (AUdbch_cnt[]). However, if this channel is an allocable channel and is associated with a RISLU/ISLU cut-through PIDB the RISLU/ISLU allocation count (islu_alloc) will be pegged in addition to the usage (usecnt) and allocation (allocnt) counts. If the channel assignment is associated with a nailed-up channel (PCINITCHAN) only the usage count is pegged in the local copy of the map. At the completion of the audit the local copy of the channel allocation map (AUdbch_cnt[]) is compared to the peripheral control channel allocation map (PCchalloc[]).

4. ERROR HANDLING

If any dynamic tuples are lost, the audit will report an error and reinsert them.

If an inconsistency was found between the local copy of the channel allocation map (AUdbch_cnt[]) and the Peripheral Control channel allocation map (PCchalloc[]), the Peripheral Control channel allocation map (PCchalloc[]) is updated to be consistent with the local copy of the map (AUdbch_cnt[]).

If any inconsistencies are found, the rPORTLA and rCDBCOM tuples are zeroed and placed on the appropriate idle lists.

Units allowed to use the MRA recovery queue will recover the rSMEST to a safe, consistent state and (if MRA restorable) mark the rSMEST tuple SMAUOOS. The units will interface with SMIM to load the circuit on the MRA queue, which will result in an MRA restoration of the circuit to get the hardware and all software back to a consistent state. If the rSMEST tuple for the circuit/parent/mate is already in the MRA recovery queue (marked SMAUOOS) due to previous audit recovery, the audit skips further recovery. Also, for circuits having rCDBCOM tuples, the CDBCOM is placed on the OOS list to keep call processing from accessing the circuit while on the MRA recovery queue.

Circuits that fail the vertical check will be recovered to the correct state. Circuits that fail the horizontal check will be recovered via the MRA queue.

If a linkage between an rCDBCOM and an rICHDB tuple should exist but does not, an rICHDB tuple and a channel are allocated and linked to that rCDBCOM.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The bit_access field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of incorrect entry in CMrcldata.
<i>Bad Data =</i>	CMrcldata.hdw[] .bit_access.
<i>Good Data =</i>	Correct access type based on the unit type 8 for DLTU, 16 for DLTU2.
<i>Logical Index =</i>	RCL circuit name.
<i>Dump =</i>	CMrcldata entry.

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An inconsistency has been detected between the dynamic data rSMEST CGA

	alarm fields and the actual state of the DFI hardware.
Possible Error Effect:	Alarms raised on DFI circuitry are not being served.
Corrective Action Taken:	Dynamic data is updated to agree with the DFI hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rISMEST in error.
<i>Bad Data =</i>	Bit map containing SMEST alarm fields: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Good Data =</i>	Bit map read from DFI hardware alarm: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rISMEST.

Table 3 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	An rICDBCOM tuple exists for a circuit that is unequipped.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rICDBCOM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rICDBCOM tuple.
<i>Dump =</i>	rICDBCOM tuple.

Table 4 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rISMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rISMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rISMEST tuple.

Table 5 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 6 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	A circuit is marked as rSMEST.qual_2 = SMAUOOS but either is not on the maintenance request administrator's (MRA's) queue, or its parent is not on the queue.
Possible Error Effect:	The actual maintenance state of a circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	Circuit is SMAUOOS but not in queue (4).
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of the circuit.
<i>Dump 2</i> =	rSMEST of the parent circuit.

Table 7 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A IDCUS1 facility and its rSMEST.qual_2 is SMFAF, but either an alarm bit is not set, or the circuit is not on the MRA recovery queue.
Possible Error Effect:	The maintenance state of the circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The unittyp field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .unittyp.
<i>Good Data</i> =	Unit type according to rCKTDATA.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An RLI tuple in the SMmcastat[] array was found to be OOS RMVD AUTO. This is not a legal state for the RLI.
Possible Error Effect:	The RLI will not be usable.
Corrective Action Taken:	The RLI will be recovered to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_1.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 10 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The circuit has growth or special growth status but bas_stat is not equal to SMOOS.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	The bas_stat field is corrected to SMOOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 11 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state is active restricted but the circuit is not a PCS96FAC or a PCDCCLCC.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 12 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The global data identifying remote cluster link (RCL) information has been corrupted. The picb field is incorrect when compared to rICKTDATA for the RCL.
Possible Error Effect:	Intercluster communication may be impaired.
Corrective Action Taken:	The PICB number stored in the CMrcldata tuple will be set to the PICB derived from the rICKTDATA base address.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CMrcldata.hdw entry that is in error.
<i>Bad Data</i> =	The PICB value of the bad entry.
<i>Good Data</i> =	The PICB value derived from the base address in the rICKTDATA tuple.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	NA.

Table 13 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The circuit is operational and either in service or unavailable but the qualifier, qual_1, is not SMSNUL.
Possible Error Effect:	Circuit may be erroneously denied service if bad value is unequipped or some similar value.
Corrective Action Taken:	Qualifier field is corrected to SMSNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.

<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Value of SMSNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 14 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The circuit is operational and either in service or unavailable, but the qualifier, qual_2, is not SMSTNUL or SMPROOS with bas_stat SMACT.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier field is corrected to SMSTNUL. If the circuit is the remote clock, qual_2 is set to the hardware value of the remote clock mode: SMSMRCNORM, SMRCFREE, SMRCFAST, or SMRCHOLD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_2 field.
<i>Good Data</i> =	Value of SMSTNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 15 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process and does not have a parent circuit which could be busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 16 Error Code: CAMA_IS

Error Code:	CAMA_IS
Error Description:	A CAMA-ONI keying trunk is incorrectly in service when its associated talking trunk is out-of-service.
Possible Error Effect:	The keying trunk will be hunted but it will not be possible to get control of the talking trunk. Per call failures will result whenever this CAMA-ONI trunk pair is hunted.
Corrective Action Taken:	The keying trunk is recovered to an OOS,FE,CAMA status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rPORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rPORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rPORTLA tuple for the talking trunk.

Table 17 Error Code: CAMA_OOS

Error Code:	CAMA_OOS
Error Description:	A CAMA-ONI keying trunk is incorrectly out-of-service (supposedly due to the talking trunk being out-of-service) when the associated talking trunk is in service.
Possible Error Effect:	The keying trunk will not be hunted and so this CAMA-ONI trunk pair will never

	be used by call processing.
Corrective Action Taken:	The keying trunk is recovered to an in service status provided its circuit status is in service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the keying trunk rIORTLA tuple.
<i>Bad Data =</i>	bsc_state for the keying trunk.
<i>Good Data =</i>	bsc_state for the talking trunk.
<i>Logical Index =</i>	Port name of the keying trunk.
<i>Dump 1 =</i>	rIORTLA tuple for the keying trunk.
<i>Dump 2 =</i>	rIORTLA tuple for the talking trunk.

Table 18 Error Code: CCDUPFAIL

Error Code:	CCDUPFAIL
Error Description:	The array SMccdupfail[] duplex failure status does not match the service groups' rISMEST tuple status. The error could be either that the array has the line unit as being duplex failing while at least one of the service groups' rISMEST tuples are active or the array indicates that the line unit is not duplex failing while both service groups rISMEST tuples are OOS.
Possible Error Effect:	The line unit will not have the correct IS/OOS status.
Corrective Action Taken:	The array will be corrected to reflect the correct status as per rISMEST tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rISMEST tuple.
<i>Bad Data =</i>	The status of the line unit as it appears in the SMdupfail[] array.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The circuit name of the service group.
<i>Dump 1 =</i>	rISMEST tuple for service group zero.
<i>Dump 2 =</i>	rISMEST tuple for the service group one.

Table 19 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The service status [in-service (0), out-of-service (1), or don't care (2)] of the rICDBCOM tuple disagrees with that of the rIORTLA tuple corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rICDBCOM tuple is placed on the idle link list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICDBCOM tuple.
<i>Bad Data =</i>	Service status of rICDBCOM tuple.
<i>Good Data =</i>	Expected status.
<i>Logical Index =</i>	Internal name (rICDBCOM key).
<i>Dump =</i>	rICDBCOM tuple.

Table 20 Error Code: CGAQN_BITY

Error Code:	CGAQN_BITY
Error Description:	The rISMEST tuple indicates that the circuit is queued for carrier group alarm (CGA), yet it was not found to be on the TM queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rISMEST.tm_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rISMEST tuple.
<i>Bad Data =</i>	Value of rISMEST.tm_que field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Circuit's rISMEST tuple.

Table 21 Error Code: CGAQY_BITN

Error Code:	CGAQY_BITN
Error Description:	The circuit was found to be queued for carrier group alarm, yet the appropriate field in rISMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The rISMEST.tm_que field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Value of rISMEST.tm_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rISMEST tuple.

Table 22 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	A cdbcom that should have a dedicated linkage to a chdb is not linked to a chdb.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and reinitializes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	cdbcom.

Table 23 Error Code: CKTCUT

Error Code:	CKTCUT
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to store the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data</i> =	Value found in that address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump</i> =	The whole array AUs96inhp[].

Table 24 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of start of CMrcldata.hdw.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	Circuit name of the RCL.
<i>Logical Index</i> =	Circuit name of the RCL.
<i>Dump</i> =	NA.

Table 25 Error Code: CKTNAME

Error Code:	CKTNAME

Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rICKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter-cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CMrcldata[].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 26 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	A discrepancy was found between the actual usage of rICDBCOM and rIPTSB tuples and the count of INITCHANS and ALLOCHANS stored in the channel allocation map (PCchalloc[]) maintained by peripheral control. The contents of the usage counts (usecnt), allocation counts (allocnt), and ISLU allocation counts (islu_alloc) between the peripheral control channel allocation map (PCchalloc[]) and the audit's copy of the channel allocation map (AUdbch_cnt[]) are found to be inconsistent.
Possible Error Effect:	Peripheral control may have problems allocating channels to calls. Also, peripheral control will have problems relocating idle code and audible ring channels during hardware growth.
Corrective Action Taken:	The channel allocation map maintained (PCchalloc[]) by peripheral control will be updated from the local copy of the channel allocation (AUdbch_cnt[]) map kept by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The contents of the global flag (AUpc_upckt) which indicates if peripheral control has updated their copy of the channel allocation map.
<i>Logical Index =</i>	The PIDB timeslot (dbchan) or index into the channel allocation map that was found in error.
<i>Dump 1 =</i>	The contents of the channel allocation map (PCchalloc[]) maintained by peripheral control.
<i>Dump 2 =</i>	The contents of the channel allocation map (AUdbch_cnt[]) maintained by audits.

Table 27 Error Code: CURLNK

Error Code:	CURLNK
Error Description:	Audit found the rICDBCOM tuple with no rICHDB tuple attached to it for a DS0 circuit name in a DNU-S. The link field for the rICHDB tuple was found to have the null value, DALNKNUL.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The type field of the rICKTDATA tuple.
<i>Bad Data =</i>	The link field of rICDBCOM tuple that was found to be null, cdccm.st_link .
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rICDBCOM tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 28 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICDBCOM is associated with a system tone but is not on the DBCADGEN

	idle list.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICDBCOM tuple.
<i>Bad Data</i> =	The owner of the rICDBCOM tuple.
<i>Good Data</i> =	DBCADGEN head cell value.
<i>Logical Index</i> =	Logical key of the rICDBCOM tuple.
<i>Dump</i> =	rICDBCOM tuple.

Table 29 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The max_ddl_size field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .max_ddl_size.
<i>Good Data</i> =	18 or 30.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 30 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	Inconsistencies have been detected in SMbufuse[] and/or SMquse[] global data used by the maintenance request administrator (MRA).
Possible Error Effect:	Possible problems scheduling MRA jobs.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from SMbufaud().
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 31 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	An rIDS0B tuple is missing from the data base.
Possible Error Effect:	The line using that rIDS0B tuple will not have any call processing capability.
Corrective Action Taken:	The missing rIDS0B tuple will be inserted in the data base and linked to the correct list. The list is determined by the in-service or out-of-service state of the rSMEST's tuple for the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDS0B tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The key for the rIDS0B tuple.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	NA.

Table 32 Error Code: DS1PT_LOST

Error Code:	DS1PT_LOST
Error Description:	The audit detected that one of the rIDS1_PT tuple for a given IDCU DS1 facility is missing from the ODD. The rIDS1_PT tuple has two keys (first key is the rIDS1_PT.ds1name and the second key is the rIDS1_PT.key_slot). There should exist two tuple with the same first key but with different second key (zero

	and 12). The audit detected that one tuple was missing. Both tuples or none should be populated.
Possible Error Effect:	The nailup call processing using that tuple will not work.
Corrective Action Taken:	None.
Manual Action Required:	The office personnel should determine if the existing rIDS1_PT tuple should be deleted or the missing tuple should be inserted into the data base.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS1_PT tuple.
<i>Bad Data =</i>	The return from the database read of the first half of the rIDS1_PT tuple where rIDS1_PT.key_slot=0. If it is a zero (GLSUCCESS), it means that the tuple existed, else the tuple was missing.
<i>Good Data =</i>	The return from the database read of the second half of the rIDS1_PT tuple where rIDS1_PT.key_slot=12. If it is zero, it means that the tuple existed, else the tuple was missing.
<i>Logical Index =</i>	Circuit name of the DS1 facility.
<i>Dump =</i>	The contents of the rIDS1_PT tuple.

Table 33 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SMIM macros to read circuit name from one circuit to other circuit failed to read static relations, either RLCKTDATA or RLUNIT_CKT. Audit was trying to get the DS0 circuit name from any of the STE, STS1, VT1.5 or DS1 circuits for a DNU-S facility. Since this is a CKTDATA audit, the more likely missing tuple could be rLUNIT_CKT tuple for the DS0 circuit.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until either the appropriate rCKTDATA or rLUNIT_CKT tuple is inserted.
Manual Action Required:	If rCKTDATA tuple is missing, insert it with the key value dumped in the logical key field of audit message. For missing rLUNIT_CKT tuple, there are three keys. The unit key is dumped in the error address field. The utype key is dumped in the good data field. Audit does not have the information about the third key, ckt_indx, as this error was passed on to audit by SMIM code and that code calculated this circuit index value. Please contact any SMIM developer for this value.
Dump Description:	
<i>Error Address =</i>	The unit field of the rCKTDATA tuple.
<i>Bad Data =</i>	The circuit name received by audit. This should be GLNULCKT.
<i>Good Data =</i>	The unit type from the utype field of rCKTDATA tuple.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 34 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	The circuit is operational and out-of-service, but bas_stat has an invalid value.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad bas_stat field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 35 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	A BOOTSTRAP circuit (PCBTSR) on an SMP20 (switching module processor using <i>Motorola</i> ® 68020 processor), which is operational, has an rSMEST tuple which indicates that the circuit is not active.
Possible Error Effect:	Offline pump may fail for this switching module (SM).
Corrective Action Taken:	The rSMEST bas_stat is set to SMACT, qual_1 is set to SMSNUL, and qual_2 is set to SMSTNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	Value of bas_stat in rSMEST.
<i>Good Data</i> =	The smest state value of SMACT=1.
<i>Logical Index</i> =	The circuit name.
<i>Dump 1</i> =	The rCKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 36 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DNU-S facility and the carrier group alarm state. When the facility is marked active (ACT), there should be no carrier group alarm present.
Possible Error Effect:	The facility in the question may not be used, thus possible lost resource.
Corrective Action Taken:	The alarm fields are cleared to indicate that no alarm is present.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The red_alm field of the rSMEST tuple.
<i>Bad Data</i> =	The ylw_alm field of the rSMEST tuple.
<i>Good Data</i> =	The bas_stat field of the rSMEST tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 37 Error Code: KEY

Error Code:	KEY
Error Description:	A rSMEST tuple contains a bad logical key.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit reinitializes the rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Logical key of rSMEST.
<i>Good Data</i> =	Logical key of associated rCDBCOM tuple.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST.

Table 38 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	No success in reading the tuple in RLSMEST to see if DCLU is in service.
Possible Error Effect:	No SLC® 96 call processing through this S-DFI.
Corrective Action Taken:	Request DFISHELF audit indirectly through T1USG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to DFI tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Equipment status table of the unit.

Table 39 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The rSMEST tuple indicates that the circuit corresponds to a live process, but it

	is not maintenance busy.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process which claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 40 Error Code: LOST_GSDB

Error Code:	LOST_GSDB
Error Description:	A rGSDB tuple has been found missing from the database for either a grid circuit (PCLCGRID, PCGRID), a SLC [®] 96 bank circuit (PCS96BNK), a SLC [®] 96 shelf circuit (PCS96SHF), or a line group controller circuit (PCLGC) that is marked equipped in the rICKTDATA tuple (eqstat != SMUNEQ).
Possible Error Effect:	All line ports on these circuits will not be able to originate a call.
Corrective Action Taken:	The rGSDB tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSDB tuple found in error.
<i>Bad Data</i> =	The circuit type found in the rICKTDATA tuple (type).
<i>Good Data</i> =	The circuit name found in the rICKTDATA tuple (name).
<i>Logical Index</i> =	The logical key of the rGSDB tuple found missing in the database.
<i>Dump</i> =	The rICKTDATA tuple associated with this rGSDB tuple.

Table 41 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 42 Error Code: LOST_PTSB

Error Code:	LOST_PTSB
Error Description:	A PIDB circuit lacks one of its 32 corresponding rPTSb tuples.
Possible Error Effect:	The time slot resource is lost to the system.
Corrective Action Taken:	The rPTSb tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPTSb.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Key of rPTSb.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 43 Error Code: LOST_SMEST

Error Code:	LOST_SMEST
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Type of circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 44 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	A circuit lacks its corresponding rCDBCOM tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rCDBCOM tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type found.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 45 Error Code: LOSTTUP (1)

Error Code:	LOSTTUP (1)
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 46 Error Code: LOSTTUP (2)

Error Code:	LOSTTUP (2)
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 47 Error Code: LOSTTUP (3)

Error Code:	LOSTTUP (3)
Error Description:	The ttf tuple associated with this rCKTDATA tuple does not exist but should.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The ttf tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCKTDATA tuple.
<i>Bad Data</i> =	Associated ttf relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rCKTDATA circuit name.
<i>Dump</i> =	NA.

Table 48 Error Code: LOSTTUP (4)

Error Code:	LOSTTUP (4)
Error Description:	No success in reading the tuple from RLS96BASE to see if SG is in service.

Possible Error Effect:	No SLC [®] 96 call processing through this service group.
Corrective Action Taken:	Request BASE audit indirectly through PATHMAP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to DFI tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Equipment status table of the unit.

Table 49 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DFI facility and the carrier group alarm state. When there is a carrier group alarm, the facility should be marked out-of-service, auto, or removed but is not.
Possible Error Effect:	DFI facilities with CGAs will not be removed from service properly.
Corrective Action Taken:	The facility is removed and restored if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple in error.
<i>Bad Data</i> =	Facility basic state (rSMEST.bas_stat).
<i>Good Data</i> =	Out-of-service (4).
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	rSMEST.

Table 50 Error Code: MATE

Error Code:	MATE
Error Description:	The rSMEST tuple indicates that the mate of this circuit is in an invalid state. This is applicable to ACT/STBY type of circuits.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The circuit/mate is restored to correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Mate's bas_stat field in rSMEST tuple.
<i>Good Data</i> =	Circuit's bas_state field in rSMEST tuple.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Circuit's unit ID.
<i>Dump 2</i> =	Mate's unit ID.

Table 51 Error Code: MISSING

Error Code:	MISSING
Error Description:	Audit failed to read a rICDBCOM tuple for a DS0 circuit name for a DNU-S facility.
Possible Error Effect:	Possible failure of resource for this DS0 circuit.
Corrective Action Taken:	None. The audit will report this error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rICKTDATA tuple, name .
<i>Good Data</i> =	The circuit name for which audit could not get rICDBCOM tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 52 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The dynamic rPICB tuple associated with the static rICKTDATA tuple for this PIDB circuit is missing from the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PICB audit is scheduled to insert the missing rPICB tuple into the database.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA in error.
<i>Bad Data</i> =	Key value of rPICB.
<i>Good Data</i> =	PICB value recalculated from circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rICKTDATA.
<i>Dump 2</i> =	rPICB.

Table 53 Error Code: MMP_IIA

Error Code:	MMP_IIA
Error Description:	State is allow but service request mask says inhibited.
Possible Error Effect:	No SLC [®] 96 call processing through this S_DFI.
Corrective Action Taken:	The service request mask will be corrected by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	The bit position that is wrong, start counting from 0.
<i>Dump</i> =	The equipment status table of the unit.

Table 54 Error Code: MP_HDW_IA

Error Code:	MP_HDW_IA
Error Description:	A rSMEST tuple has a bas_stat value only allowed for module processor equipment, and it isn't part of the module processor complex.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit will reinitialize the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Bad bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST tuple.

Table 55 Error Code: MRAQN_BITY

Error Code:	MRAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for recovery, yet it was not found to be on the queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.mra_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 56 Error Code: MRAQY_BITN

Error Code:	MRAQY_BITN
Error Description:	The circuit was found to be queued for restoration, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.mra_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.

<i>Dump</i> =	Circuit's rSMEST tuple.

Table 57 Error Code: MSCU_IIA

Error Code:	MSCU_IIA
Error Description:	Since mate service request source register is used, bit 15 of the own SRSR must be allowed.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	This bit 15 of the "own" SRSR is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	15.
<i>Dump</i> =	The equipment status table of the unit.

Table 58 Error Code: NO_CHDB

Error Code:	NO_CHDB
Error Description:	The rCHDB tuple accessed by the audit is not the same tuple audit tried to get. It failed the key check, that is, the key of the rCHDB tuple, chnum is not equal to the link field in the rCDBCOM tuple, cdccm.st_link minus the shift for the rCHDB tuple, DBSFCHDB. The missing rCHDB tuple represented a DS0 circuit in a DNU-S.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type field of the rCKTDATA tuple.
<i>Bad Data</i> =	The key field of rCHDB tuple, chnum .
<i>Good Data</i> =	The link field of rCDBCOM tuple, cdccm.st_link minus the shift value for rCHDB tuple, DBSFCHDB.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rCHDB tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 59 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous head cell (DBIDLMISC).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Idle link list physical key.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump 1</i> =	CKTDATA tuple.
<i>Dump 2</i> =	CDBCOM tuple.

Table 60 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their RLSMEST tuple and in the F bits of the signaling information of the peripheral control and timing (PCT) links time slots. Audit found inconsistency between these informations when it compared it. The inconsistency could be in the form that the F bits says no alarm is present, but the rSMEST of the circuit says there is an alarm present or the types of alarm present are not the same in

	both places. For exact information as to what happened, see the explanation in error address field in this section.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The audit will call TM function to put the circuit on the CGA recovery queue where its alarm information will be correctly applied.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field represents the type of error that occurred. Depending upon its value different explanations are given here. 1 = The F bits says that there is no alarm present on this circuit but one or more of the alarm bits are set in the rSMEST tuple. 2 = The F bits says AIS alarm is present but the rSMEST says either AIS alarm is not set or one of the other alarms, red or yellow, is set. 3 = The F bits says red alarm is present but the rSMEST says either red alarm is not set or one of the other alarms, AIS or yellow, is set. 4 = The F bits says yellow alarm is present but the rSMEST says either yellow alarm is not set or one of the other alarms, AIS or red, is set. 5 = The audit received a value from TM that it is not expecting to receive.
<i>Bad Data</i> =	The local value of circuit type derived from the F bits, ckdtype .
<i>Good Data</i> =	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index</i> =	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1</i> =	The rSMEST tuple.
<i>Dump 2</i> =	The rCDBCOM tuple.

Table 61 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rSMEST tuple is not transient but its progress flag is set.
Possible Error Effect:	Audit transient timing may operate improperly.
Corrective Action Taken:	The progress flag is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST tuple.

Table 62 Error Code: NOTSPARE

Error Code:	NOTSPARE
Error Description:	An rSMEST tuple has a bas_stat value only allowed for module processor and remote clock equipment.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The tuple is reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST for parent/mate (OPTIONAL).

Table 63 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their RLSMEST tuple and in the F bits of the signaling information of the PCT links time slots. The audit found an inconsistency when it compared the CGA alarm bits in the RLSMEST tuple against the bas_stat and qual_2 fields in that tuple. These two fields indicate the base state of the facility represented by the RLSMEST tuple. The alarm bits in the RLSMEST tuple (ais_alm , red_alm , or ylw_alm) should only be set if the base state indicates either out of service family equipment (OOS/FE) or out of service facility failure (OOS/FAF). In any other case the alarm bits should not be set.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The alarm bits (ais_alm , red_alm , or ylw_alm) are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The local value of circuit type derived from the F bits, ckttype .
<i>Good Data =</i>	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index =</i>	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1 =</i>	The rLSMEST tuple.
<i>Dump 2 =</i>	The rICDBCOM tuple.

Table 64 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	No rISLUPIDB tuple exists for this PCPIDBISLU or PCPIDBRISLU circuit.
Possible Error Effect:	This resource is lost to the system.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PICB value used to read rISLUPIDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 65 Error Code: ODDLOSTTUP (1)

Error Code:	ODDLOSTTUP (1)
Error Description:	A tuple in the static relation rIDSLEQUIP is missing.
Possible Error Effect:	Full use of DSL equipment is impossible.
Corrective Action Taken:	None.
Manual Action Required:	A tuple should be inserted into rIDSLEQUIP as appropriate for the missing PCRDTTMC or PCRDEOC circuit.
Dump Description:	
<i>Error Address =</i>	Address of rIDSLEQUIP buffer.
<i>Bad Data =</i>	Return value from DB (DBNO MATCH).
<i>Good Data =</i>	Circuit name.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rICKTDATA.

Table 66 Error Code: ODDLOSTTUP (2)

Error Code:	ODDLOSTTUP (2)
Error Description:	The CKTDATA audit is attempting to read a nonexistent rIIDCKT2GP tuple. The key to the tuple is derived from the rICKTDATA tuple circuit name and circuit index.
Possible Error Effect:	Call Processing cannot use the circuit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rICKTDATA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA tuple.

Table 67 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The circuit is operational and out-of-service but qual_1 and qual_2 do not have compatible values.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 68 Error Code: PARENT

Error Code:	PARENT
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process nor is its parent circuit busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	Parent's rSMEST.

Table 69 Error Code: PCD

Error Code:	PCD
Error Description:	The audit was not able to read the rVPIDB_F tuple for a corresponding either the STE circuit or the STS-1 circuit for a DNU-S. Refer to the <i>Good Data</i> field to determine for which type of circuit, STE or STS-1, audit could not read the rVPIDB_F tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report this error until the rVPIDB_F tuple is inserted.
Manual Action Required:	There are three key field for the rVPIDB_F tuple, unittype , unitnumber and pct_link . Use the value of utype and unit from the rICKTDATA dump of this error for the first two keys of the rVPIDB_F tuple, unittype and unitnumber respectively. Use the value of key field for the third key of the rVPIDB_F tuple, pct_link and insert this rVPIDB_F tuple.
Dump Description:	
<i>Error Address</i> =	The unit field of the rICKTDATA tuple.
<i>Bad Data</i> =	Contents of the DS0 circuit name calculated by audit if any.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The pct_link key of the rVPIDB_F tuple as calculated by audit from the given data.
<i>Dump 1</i> =	The rICKTDATA tuple, either for STE or STS-1.
<i>Dump 2</i> =	The rVPIDB_F tuple as returned by database read.

Table 70 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the rICKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the data base.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the data base.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rPICB tuple.
<i>Dump =</i>	NA.

Table 71 Error Code: PICBKEY

Error Code:	PICBKEY
Error Description:	The static relation rISLUPIDB is missing a tuple.
Possible Error Effect:	The tuple is not available for access.
Corrective Action Taken:	None.
Manual Action Required:	The tuple should be inserted into the static relation rISLUPIDB.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PICB value.
<i>Good Data =</i>	Circuit name.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 72 Error Code: PID

Error Code:	PID
Error Description:	The PID in the tp_pid field is not OSnullpid nor a live PID and thus is garbage.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The PID is set to OSnullpid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	PID of process that claims association.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 73 Error Code: QUAL

Error Code:	QUAL
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but qual_1 indicates the circuit is growth or unequipped and not on the automatic or manual request queues.
Possible Error Effect:	The circuit may be erroneously processed by MRA or lost to call processing.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.

<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 74 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The circuit is operational and out-of-service, but qual_2 has an invalid value.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_2 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	Cross reference entry.
<i>Dump 3 =</i>	Queue entry.
<i>Dump 4 =</i>	rSMEST tuple under audit.
<i>Dump 5 =</i>	Parent of the rSMEST tuple under audit.

Table 75 Error Code: QUAL_EQPG

Error Code:	QUAL_EQPG
Error Description:	The circuit is growth or special growth, but qual_1 is not equal to SMGR or SMSGR.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qual_1 field is corrected to SMGR or SMSGR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 76 Error Code: QUE

Error Code:	QUE
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but the circuit does not have an entry in the queue specified by qual_1 (auto or man).
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	Identity of queue in which the circuit was found.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 77 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	A reconfiguration request of a remote clock circuit was not successful.
Possible Error Effect:	Remote clocking may be impaired.

Corrective Action Taken:	All remote clock circuits generating this error, except the parent circuit, are marked out-of-service or active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Return value of the configuration routine.
<i>Good Data</i> =	Action performed on the circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rISMEST of parent circuit.
<i>Dump 2</i> =	rISMEST of subtending or mate side to a remote clock side.

Table 78 Error Code: READ

Error Code:	READ
Error Description:	Remote clock hardware can not be read at this time.
Possible Error Effect:	Can not determine active RCLK.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Value of mate's rISMEST.bas_stat field.
<i>Good Data</i> =	Value of circuit's rISMEST.bas_stat field.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rISMEST of circuit.
<i>Dump 2</i> =	rISMEST of mate circuit.

Table 79 Error Code: RT2DS1_LOST (1)

Error Code:	RT2DS1_LOST (1)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCU facilities.
Corrective Action Taken:	None.
Manual Action Required:	A tuple in RLRT2DS1 should be inserted as appropriate for the circuit identified in the associated audit dump.
Dump Description:	
<i>Error Address</i> =	Address of rIRT2DS1 buffer.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rCKTDATA.

Table 80 Error Code: RT2DS1_LOST (2)

Error Code:	RT2DS1_LOST (2)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCU facilities.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRT2DS1 buffer.
<i>Bad Data</i> =	Shelf number of RT circuit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rCKTDATA.

Table 81 Error Code: SCNBAD

Error Code:	SCNBAD
Error Description:	An inconsistency has been detected in the value of the IDCU register HWMLSCAN; either it is set when it should not be or vice versa. If the IDCUCOM is active, the bit should be set. If the IDCUCOM is stby, the bit should not be set.
Possible Error Effect:	The scanning state is incorrect for the IDCUCOM identified.
Corrective Action Taken:	The LSCAN bit is reset to the correct value.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	HWMLSCAN.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 82 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Audit tried to read a tuple of the static relation, RLFAC_PROV, as part of the CGA checks on DNU-S, but failed to read it.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until the appropriate rFAC_PROV tuple is inserted.
Manual Action Required:	Insert a rFAC_PROV tuple with the key equal to the circuit name dumped in either the logical key or error address fields. The same value is dumped in these fields from different data structures.
Dump Description:	
<i>Error Address</i> =	The name field of the rCKTDATA tuple.
<i>Bad Data</i> =	The key of the rFAC_PROV tuple, cktname , which audit could not access.
<i>Good Data</i> =	The circuit type from the type field of rCKTDATA tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rCKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 83 Error Code: SMEST

Error Code:	SMEST
Error Description:	The service status [in-service (0), out-of-service (1)] of the rSMEST tuple disagrees with that of other tuples corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name (or resources, if the name corresponds to a family of equipment) to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rSMEST tuple is forced idle and re-audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Service status (in/out) of rSMEST tuple.
<i>Good Data</i> =	Expected status.
<i>Logical Index</i> =	Internal name (rSMEST key).
<i>Dump</i> =	rSMEST tuple.

Table 84 Error Code: SSTBAD

Error Code:	SSTBAD
Error Description:	Register contents do not match the contents of the IDCUCOM rSMEST and the LSI(s) rCKTDATA.
Possible Error Effect:	The hardware state of the IDCU is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	Constructed mask indicating correct service state.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 85 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rSMEST tuple indicates that this circuit is idle, but it has an entry in one of the queues or the xrf table.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	None.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Cross reference entry.
<i>Dump 2</i> =	Queue entry.
<i>Dump 3</i> =	rSMEST tuple under audit.
<i>Dump 4</i> =	Parent of the rSMEST tuple under audit.

Table 86 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkusage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	linkusage (from cdbcom).
<i>Good Data</i> =	linkusage (expected).
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	Circuit type CDBCOM.

Table 87 Error Code: STATUS

Error Code:	STATUS
Error Description:	The rSMEST tuple indicates that this circuit is neither busy nor idle, thus the stat_act field is invalid.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 88 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rSMEST tuple has been stuck for several minutes in a transient state.
Possible Error Effect:	A process using the rSMEST tuple is stuck with allocated resources and some audit consistency checks are prevented.
Corrective Action Taken:	The rSMEST tuple is recovered to a non-transient state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST tuple.

Table 89 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	One or both RCLK circuits has been busy for at least 5 seconds under control of the GLPNRCLK process.
Possible Error Effect:	The circuit may be lost for further MRA requests.
Corrective Action Taken:	The RCLK circuits are further checked to verify RCLK configuration status and are recovered to a safe/consistent state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.

<i>Bad Data</i> =	MRA activity value rSMEST.stat_act.
<i>Good Data</i> =	DBIDLE
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST for busy RCLK.
<i>Dump 2</i> =	rSMEST for mate RCLK.

Table 90 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rCDBCOM and its associated rTONECH tuple failed a point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCDBCOM tuple.
<i>Bad Data</i> =	Tone field in the rCDBCOM tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rCDBCOM tuple.
<i>Dump 1</i> =	rCDBCOM tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 91 Error Code: TYPE

Error Code:	TYPE
Error Description:	The type of circuit that the CKTDATA audit is attempting to audit is either not present in its private circuit mapping array (AUcktreimap[]), or it is marked as invalid in the map.
Possible Error Effect:	The audit will continue to print the error each time it runs but will otherwise ignore the circuit. This is an audit error.
Corrective Action Taken:	None.
Manual Action Required:	Report the error with failure report (FR). Correcting this problem may require manual boot of the SM to restore the RLCKTDATA relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 92 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The circuit is unequipped but the qualifier qual_1 is not SMUN.
Possible Error Effect:	The circuit is accessible but has no hardware, which will result in interrupts when it is used.
Corrective Action Taken:	Qualifier field is corrected to SMUN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Value of SMUN.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 93 Error Code: VERTICAL

Error Code:	VERTICAL
Error Description:	The maintenance state of a circuit (rSMEST.bas_stat) is vertically inconsistent with the maintenance state of its "parent" circuit.
Possible Error Effect:	The circuit may not be available for use.
Corrective Action Taken:	The child's rSMEST tuple is made consistent with the parent's.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Pointer to rSMEST tuple for the child circuit.
<i>Bad Data</i> =	Child's rSMEST.bas_stat.
<i>Good Data</i> =	Number of parent circuits.
<i>Logical Index</i> =	Child circuit name.
<i>Dump 1</i> =	The child's rSMEST tuple.
<i>Dump 2</i> =	All the parent circuit rSMESTs (may be several).

Table 94 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Actual idle list physical key.
<i>Good Data</i> =	Physical key of correct idle list.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	CDBCOM tuple.

Table 95 Error Code: XRFPID

Error Code:	XRFPID
Error Description:	The rSMEST tuple indicates that this circuit has a live PID and the circuit is in xrf or one of the queues, but the PID doesn't match that in the queue.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	xrf search results.
<i>Dump 2</i> =	manq, autoq search results.
<i>Dump 3</i> =	rSMEST.
<i>Dump 4</i> =	Dump of parent's rSMEST tuple.

CKTDATA-C

Software Release: 5E12 only

1. CIRCUIT DATA (CKTDATA) AUDIT

The circuit data audit (CKTDATA) verifies that the information within each of these structures is correct and that the data is consistent when compared with the other data structures. It also performs static-to-dynamic existence checks for all data to find lost dynamic tuples. The CKTDATA audit has seven phases. The first phase calls the port linkage area audit (PORTLA) to check for the existence of trunk ports. The second phase checks the dynamic circuit data. The third phase checks the maintenance status of circuits. The fourth phase checks the consistency among the RLPORTLA, RLCDBCOM, and RLSMEST relations. The fifth phase checks the static-to-dynamic existence of PIDB time slot block relation (RLPTSB) tuples created for integrated services line unit (ISLU) and packet switching unit (PSU) peripheral interface data buses (PIDBs). The sixth phase performs the static-to-dynamic existence checks between the static RLCKTDATA relation and the dynamic RLGSDb relation. The seventh phase performs integrity checks between the RLCKTDATA relation and the peripheral control channel allocation maps (PCchalloc[]).

This audit is part of the peripheral control audits subgroup. It runs at routine level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The CKTDATA audit uses the circuit data relation (RLCKTDATA), a static relation, to verify information in the following dynamic data structures in the SMs: the peripheral control channel allocation map (PCchalloc[]), the port linkage area relation (RLPORTLA), the circuit data block common area relation (RLCDBCOM), the group summary data block relation (RLGSDb), the automatic queue relation (RLAUTOQ), the switch maintenance cross reference relation (RLSMXRF) and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these data structures.

3. AUDIT CHECKS

The circuit data audit uses the static driver (DRSTATIC) to step through the write-protected RLCKTDATA and retrieve tuples. The driver first determines which circuits exist and what data is available about them. It then calls the appropriate phase of the audit to perform the following checks.

Audit Phase 1 = The PORTLA audit phase of this audit uses the RLPORTLA relation to check the trunk ports. Although both trunk and line ports have tuples in RLPORTLA, this phase of the CKTDATA audit checks only trunk ports. Each trunk port has a tuple in RLPORTLA that describes its status and associations. Refer to the PORTLA audit description for more information.

The CKTDATA audit calls the AUdoportla function to perform all of the internal checks on one portla. Two additional sets of service values are required, FC_TCI_KT and FC_TCI_TT.

The CKTDATA audit also includes the following functions as part of its vertical checks:

- AUgetpar() is called to calculate the parent circuit(s) to replace the OSfap() call in AUsmest(). This allows the second pass of the audit to call the same function to calculate parents.
- The OSfap() call for vertical checks is moved into the function AUvertch(). This will allow the second pass of the audit to call the same function to perform vertical checks.
- Initialization of the vertical check data is part of the audit entry function AUcktdata(). A flag is initialized indicating no need for a second pass. If at some point during the audit it is determined that a second pass is needed, the flag will be set to true. Also a flag indicating

which pass the audit is executing in is initialized.

- When the DRSTATIC audit drivers has stepped through all of the CKTDATA tuples, the function AUdo2pass will be invoked from AUcktdata() to step through all vertical recovery candidates identified during the first pass of the audit.

Audit Phase 2

= The dynamic circuit data phase of the audit uses the RLCDBCOM relation to perform the following checks:

- The audit checks all point-to/point-back linkages to rLCDBCOM.
- The audit verifies that in-service circuits are on appropriate idle lists. In-service circuits not on idle lists cannot be used for call processing.
- The audit verifies that stable rLCDBCOM tuples have all required linkages.
- The audit verifies that idle rLCDBCOM tuples do not have any linkages except idle list linkages.

See also the circuit data block common area audit (CDBCOM).

Audit Phase 3

= The maintenance status phase of the audit uses the rSMEST relation to perform the following checks:

- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (rSMEST), the switch maintenance cross-reference relation (rSMXRF), or the automatic queue relation (rAUTOQ).
- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.

- The audit performs a child-to-parent (vertical) consistency check to ensure that subtending circuits are in the proper base state (ACT, OOS, etc.) with respect to the parent.

NOTE: Only the ISTF peripheral unit is allowed to perform the vertical check.

- The audit performs a horizontal check for circuits that have a mate circuit to ensure consistency. This is for circuits that are configured ACT/STBY or ACT MAJOR/ACT MINOR.

NOTE: Only the following circuits are allowed to perform the horizontal check:

ISLU CC
ISLU CD
PSUCOM
RISLU RRCLK

- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (RLSMEST), the switch maintenance cross-reference relation (RLSMXRF), or the automatic queue relation (RLAUTOQ).
- At the end of all the maintenance related checks, do the carrier group alarm (CGA) states checks. The units on which CGA checks are done are:
 - DFI2
 - digital networking unit-synchronous optical network (SONET) (DNU-S)

See also the switch maintenance equipment status table audit (SMEST).

- Audit Phase 4** = The consistency check phase of the audit verifies that the maintenance status for a given circuit (in-service or out-of-service) is consistent throughout the rIORTLA, rICDBCOM, and rSMEST relations.
- Audit Phase 5** = After all consistency checks are performed on the circuit, the audit checks to see if the circuit has dynamic rIPTSB tuples associated with it. Only PIDB types for ISLU, RISLU, and PSU have rIPTSBs. For these circuit types, the audit verifies that there are no lost rIPTSB tuples.
- Audit Phase 6** = After all consistency checks are performed on the circuit, the audit then insures that dynamic rIGSDB tuples exist. The rIGSDB tuples will exist for grid (PCGRID), cost reduced grid (PCLCGRID), line group controller (PCLGC), SLC[®] 96 shelf (PCS96SHF), and SLC[®] 96 bank (PCS96BNK) circuits that are marked equipped in the rLCKTDATA tuple.
- Audit Phase 7** = At the completion of the audit checks will be made to insure the integrity of the usage counts (usecnt), allocation counts (allocnt), and (R)ISLU allocation counts (islu_alloc) in the peripheral control channel allocation map (PCchalloc[]).

During the running of the audit for every circuit in the RLCKTDATA relation the channel assignment (chnasn) field is examined. If the channel assignment is associated with an

allocable channel (PCALLOCHAN) the usage and allocation counts are pegged in a local copy of the map (AUdbch_cnt[]). However, if this channel is an allocable channel and is associated with a RISLU/ISLU cut-through PIDB the RISLU/ISLU allocation count (islu_alloc) will be pegged in addition to the usage (usecnt) and allocation (allocnt) counts. If the channel assignment is associated with a nailed-up channel (PCINITCHAN) only the usage count is pegged in the local copy of the map. At the completion of the audit the local copy of the channel allocation map (AUdbch_cnt[]) is compared to the peripheral control channel allocation map (PCchalloc[]).

4. ERROR HANDLING

If any dynamic tuples are lost, the audit will report an error and reinsert them.

If an inconsistency was found between the local copy of the channel allocation map (AUdbch_cnt[]) and the peripheral control channel allocation map (PCchalloc[]), the peripheral control channel allocation map (PCchalloc[]) is updated to be consistent with the local copy of the map (AUdbch_cnt[]).

If any inconsistencies are found, the rPORTLA and rCDBCOM tuples are zeroed and placed on the appropriate idle lists.

Units allowed to use the MRA recovery queue will recover the rSMEST to a safe, consistent state and (if MRA restorable) mark the rSMEST tuple SMAUOOS. The units will interface with SMIM to load the circuit on the MRA queue, which will result in an MRA restoration of the circuit to get the hardware and all software back to a consistent state. If the rSMEST tuple for the circuit/parent/mate is already in the MRA recovery queue (marked SMAUOOS) due to previous audit recovery, the audit skips further recovery. Also, for circuits having rCDBCOM tuples, the CDBCOM is placed on the OOS list to keep call processing from accessing the circuit while on the MRA recovery queue.

Circuits that fail the vertical check will be recovered to the correct state. Circuits that fail the horizontal check will be recovered using the MRA queue.

If a linkage between an rCDBCOM and an rCHDB tuple should exist but does not, an rCHDB tuple and a channel are allocated and linked to that rCDBCOM.

The audit produces the following error code(s).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The bit_access field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .bit_access.
<i>Good Data</i> =	Correct access type based on the unit type 8 for DLTU, 16 for DLTU2.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An inconsistency has been detected between the dynamic data rSMEST CGA alarm fields and the actual state of the DFI hardware.
Possible Error Effect:	Alarms raised on DFI circuitry are not being served.
Corrective Action Taken:	Dynamic data is updated to agree with the DFI hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Bit map containing SMEST alarm fields: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Good Data =</i>	Bit map read from DFI hardware alarm: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST.

Table 3 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	An rCDBCOM tuple exists for a circuit that is unequipped.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rCDBCOM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCDBCOM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rCDBCOM tuple.
<i>Dump =</i>	rCDBCOM tuple.

Table 4 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rSMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rSMEST tuple.

Table 5 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by

	recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 6 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	A circuit is marked as rSMEST.qual_2 = SMAUOOS but either is not on the maintenance request administrator's (MRA's) queue, or its parent is not on the queue.
Possible Error Effect:	The actual maintenance state of a circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	Circuit is SMAUOOS but not in queue (4).
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of the circuit.
<i>Dump 2</i> =	rSMEST of the parent circuit.

Table 7 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A IDCUS1 facility and its rSMEST.qual_2 is SMFAF, but either an alarm bit is not set, or the circuit is not on the MRA recovery queue.
Possible Error Effect:	The maintenance state of the circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The unittype field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .unittype.
<i>Good Data</i> =	Unit type according to rCKTDATA.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An RLI tuple in the SMmccstat[] array was found to be OOS RMVD AUTO. This is not a legal state for the RLI.
Possible Error Effect:	The RLI will not be usable.
Corrective Action Taken:	The RLI will be recovered to the correct state.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Value of rISMEST.qual 1.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISMEST.

Table 10 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The circuit has growth or special growth status but bas_stat is not equal to SMOOS.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	The bas_stat field is corrected to SMOOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rISMEST tuple.
<i>Dump 2</i> =	rISMEST tuple for parent/mate (OPTIONAL).

Table 11 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state is active restricted but the circuit is not a PCS96FAC or a PCDCLCC.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rISMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rISMEST tuple.
<i>Dump 2</i> =	rISMEST tuple for parent/mate (OPTIONAL).

Table 12 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The global data identifying remote cluster link (RCL) information has been corrupted. The picb field is incorrect when compared to rICKTDATA for the RCL.
Possible Error Effect:	Intercluster communication may be impaired.
Corrective Action Taken:	The PICB number stored in the CMrcldata tuple will be set to the PICB derived from the rICKTDATA base address.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CMrcldata.hdw entry that is in error.
<i>Bad Data</i> =	The PICB value of the bad entry.
<i>Good Data</i> =	The PICB value derived from the base address in the rICKTDATA tuple.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	NA.

Table 13 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The circuit is operational and either in service or unavailable but the qualifier, qual 1, is not SMSNUL.
Possible Error Effect:	Circuit may be erroneously denied service if bad value is unequipped or some similar value.
Corrective Action Taken:	Qualifier field is corrected to SMSNUL.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Value of SMSNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 14 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The circuit is operational and either in service or unavailable, but the qualifier, qual_2, is not SMSTNUL or SMPROOS with bas_stat SMACT.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier field is corrected to SMSTNUL. If the circuit is the remote clock, qual_2 is set to the hardware value of the remote clock mode: SMSMRCNORM, SMRCFREE, SMRCFAST, or SMRCHOLD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_2 field.
<i>Good Data</i> =	Value of SMSTNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 15 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process and does not have a parent circuit which could be busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 16 Error Code: CAMA_IS

Error Code:	CAMA_IS
Error Description:	A CAMA-ONI keying trunk is incorrectly in service when its associated talking trunk is out-of-service.
Possible Error Effect:	The keying trunk will be hunted but it will not be possible to get control of the talking trunk. Per call failures will result whenever this CAMA-ONI trunk pair is hunted.
Corrective Action Taken:	The keying trunk is recovered to an OOS,FE,CAMA status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rPORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rPORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rPORTLA tuple for the talking trunk.

Table 17 Error Code: CAMA_OOS

Error Code:	CAMA_OOS
Error Description:	A CAMA-ONI keying trunk is incorrectly out-of-service (supposedly due to the

	talking trunk being out-of-service) when the associated talking trunk is in service.
Possible Error Effect:	The keying trunk will not be hunted and so this CAMA-ONI trunk pair will never be used by call processing.
Corrective Action Taken:	The keying trunk is recovered to an in service status provided its circuit status is in service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rIORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rIORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rIORTLA tuple for the talking trunk.

Table 18 Error Code: CCDUPFAIL

Error Code:	CCDUPFAIL
Error Description:	The array SMccdupfail[] duplex failure status does not match the service groups' rISMEST tuple status. The error could be either that the array has the line unit as being duplex failing while at least one of the service groups' rISMEST tuples are active or the array indicates that the line unit is not duplex failing while both service groups rISMEST tuples are OOS.
Possible Error Effect:	The line unit will not have the correct IS/OOS status.
Corrective Action Taken:	The array will be corrected to reflect the correct status as per rISMEST tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	The status of the line unit as it appears in the SMdupfail[] array.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name of the service group.
<i>Dump 1</i> =	rISMEST tuple for service group zero.
<i>Dump 2</i> =	rISMEST tuple for the service group one.

Table 19 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The service status [in-service (0), out-of-service (1), or don't care (2)] of the rICDBCOM tuple disagrees with that of the rIORTLA tuple corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rICDBCOM tuple is placed on the idle link list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICDBCOM tuple
<i>Bad Data</i> =	Service status of rICDBCOM tuple.
<i>Good Data</i> =	Expected status.
<i>Logical Index</i> =	Internal name (rICDBCOM key).
<i>Dump</i> =	rICDBCOM tuple.

Table 20 Error Code: CGAQN_BITY

Error Code:	CGAQN_BITY
Error Description:	The rISMEST tuple indicates that the circuit is queued for carrier group alarm (CGA), yet it was not found to be on the TM queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rISMEST.tm_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Value of rISMEST.tm_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rISMEST tuple.

Table 21 Error Code: CGAQY_BITN

Error Code:	CGAQY_BITN
Error Description:	The circuit was found to be queued for carrier group alarm, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The rSMEST.tm_que field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Value of rSMEST.tm_que field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Circuit's rSMEST tuple.

Table 22 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	A cdbcom that should have a dedicated linkage to a chdb is not linked to a chdb.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and reinitializes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Circuit type.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	cdbcom.

Table 23 Error Code: CKTCUT

Error Code:	CKTCUT
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data =</i>	Value found in that address.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump =</i>	The whole array AUs96inhp[].

Table 24 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of start of CMrcldata.hdw.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 25 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter-cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CMrcldata[].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 26 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	A discrepancy was found between the actual usage of rCDBCOM and rIPTSB tuples and the count of INITCHANs and ALLOCHANs stored in the channel allocation map (PCchalloc[]) maintained by peripheral control. The contents of the usage counts (usecnt), allocation counts (allocnt), and ISLU allocation counts (islu_alloc) between the peripheral control channel allocation map (PCchalloc[]) and the audit's copy of the channel allocation map (AUdbch_cnt[]) are found to be inconsistent.
Possible Error Effect:	Peripheral control may have problems allocating channels to calls. Also, peripheral control will have problems relocating idle code and audible ring channels during hardware growth.
Corrective Action Taken:	The channel allocation map maintained (PCchalloc[]) by peripheral control will be updated from the local copy of the channel allocation (AUdbch_cnt[]) map kept by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The contents of the global flag (AUpc_upckt) which indicates if peripheral control has updated their copy of the channel allocation map.
<i>Logical Index =</i>	The PIDB timeslot (dbchan) or index into the channel allocation map that was found in error.
<i>Dump 1 =</i>	The contents of the channel allocation map (PCchalloc[]) maintained by peripheral control.
<i>Dump 2 =</i>	The contents of the channel allocation map (AUdbch_cnt[]) maintained by audits.

Table 27 Error Code: CURLNK

Error Code:	CURLNK
Error Description:	Audit found the rCDBCOM tuple with no rCHDB tuple attached to it for a DS0 circuit name in a DNU-S. The link field for the rCHDB tuple was found to have the null value, DALNKNULL.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The type field of the rCKTDATA tuple.
<i>Bad Data =</i>	The link field of rCDBCOM tuple that was found to be null, cdccm.st_link .
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rCDBCOM tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 28 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICDBCOM is associated with a system tone but is not on the DBCADGEN idle list.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	The owner of the rICDBCOM tuple.
<i>Good Data =</i>	DBCADGEN head cell value.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump =</i>	rICDBCOM tuple.

Table 29 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The max_ddl_size field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of incorrect entry in CMrcldata.
<i>Bad Data =</i>	CMrcldata.hdw[] .max_ddl_size.
<i>Good Data =</i>	18 or 30.
<i>Logical Index =</i>	RCL circuit name.
<i>Dump =</i>	CMrcldata entry.

Table 30 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	Inconsistencies have been detected in SMbufuse[] and/or SMquse[] global data used by the maintenance request administrator (MRA).
Possible Error Effect:	Possible problems scheduling MRA jobs.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return from SMbufaud().
<i>Good Data =</i>	0.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 31 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The rSMEST tuple indicates that the circuit is busy (rSMEST.tp_pid) for maintenance, but it is not associated with a valid rSMXRF tuple (rSMEST.smxrf_entry).
Possible Error Effect:	Possible that stable calls could be lost when the circuit is recovered or if the terminal process is purged.
Corrective Action Taken:	The circuit under audit is recovered which includes recovering the circuit's rSMEST tuple, purging the terminal process if necessary, and running the related SMXRF audit to recover the rSMXRF tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the dynamic rSMEST tuple under audit.
<i>Bad Data =</i>	The key of the rSMXRF tuple (rSMXRF.entry).
<i>Good Data =</i>	The key of the rSMXRF tuple according to the rSMEST tuple (rSMEST.smxrf_entry).
<i>Logical Index =</i>	The key of the rSMEST tuple (rSMEST.unit_id).
<i>Dump 1 =</i>	The dynamic rSMEST tuple under audit.

<i>Dump 2 =</i>	The dynamic rSMXRF tuple.

Table 32 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	An rIDS0B tuple is missing from the database.
Possible Error Effect:	The line using that rIDS0B tuple will not have any call processing capability.
Corrective Action Taken:	The missing rIDS0B tuple will be inserted in the database and linked to the correct list. The list is determined by the in-service or out-of-service state of the rSMEST's tuple for the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDS0B tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The key for the rIDS0B tuple.
<i>Logical Index =</i>	The circuit name of the DS1 facility.
<i>Dump =</i>	NA.

Table 33 Error Code: DS1PT_LOST

Error Code:	DS1PT_LOST
Error Description:	The audit detected that one of the rIDS1_PT tuple for a given IDCUS DS1 facility is missing from the ODD. The rIDS1_PT tuple has two keys (first key is the rIDS1_PT.ds1name and the second key is the rIDS1_PT.key_slot). There should exist two tuple with the same first key but with different second key (zero and 12). The audit detected that one tuple was missing. Both tuples or none should be populated.
Possible Error Effect:	The nailup call processing using that tuple will not work.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should determine if the existing rIDS1_PT tuple should be deleted or the missing tuple should be inserted into the database.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS1_PT tuple.
<i>Bad Data =</i>	The return from the database read of the first half of the rIDS1_PT tuple where rIDS1_PT.key_slot=0. If it is a zero (GLSUCCESS), it means that the tuple existed, else the tuple was missing.
<i>Good Data =</i>	The return from the database read of the second half of the rIDS1_PT tuple where rIDS1_PT.key_slot=12. If it is zero, it means that the tuple existed, else the tuple was missing.
<i>Logical Index =</i>	Circuit name of the DS1 facility.
<i>Dump =</i>	The contents of the rIDS1_PT tuple.

Table 34 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SMIM macros to read circuit name from one circuit to other circuit failed to read static relations, either RLCKTDATA or RLUNIT_CKT. Audit was trying to get the DS0 circuit name from any of the STE, STS1, VT1.5 or DS1 circuits for a DNU-S facility. Since this is a CKTDATA audit, the more likely missing tuple could be rLUNIT_CKT tuple for the DS0 circuit.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until either the appropriate rCKTDATA or rLUNIT_CKT tuple is inserted.
Manual Action Required:	If rCKTDATA tuple is missing, insert it with the key value dumped in the logical key field of audit message. For missing rLUNIT_CKT tuple, there are three keys. The unit key is dumped in the error address field. The utype key is dumped in the good data field. Audit does not have the information about the third key, ckt_indx, as this error was passed on to audit by SMIM code and that code calculated this circuit index value. Please contact any SMIM developer for this value.

Dump Description:	
<i>Error Address</i> =	The unit field of the rICKTDATA tuple.
<i>Bad Data</i> =	The circuit name received by audit. This should be GLNULCKT.
<i>Good Data</i> =	The unit type from the utype field of rICKTDATA tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rICKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 35 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	The circuit is operational and out-of-service, but bas_stat has an invalid value.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 36 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	A BOOTSTRAP circuit (PCBTSR) on an SMP20 (switching module processor using <i>Motorola</i> ® 68020 processor), which is operational, has an rSMEST tuple which indicates that the circuit is not active.
Possible Error Effect:	Offline pump may fail for this switching module (SM).
Corrective Action Taken:	The rSMEST bas_stat is set to SMACT, qual_1 is set to SMSNUL, and qual_2 is set to SMSTNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	Value of bas_stat in rSMEST.
<i>Good Data</i> =	The smest state value of SMACT=1.
<i>Logical Index</i> =	The circuit name.
<i>Dump 1</i> =	The rICKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 37 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DNU-S facility and the carrier group alarm state. When the facility is marked active (ACT), there should be no carrier group alarm present.
Possible Error Effect:	The facility in the question may not be used, thus possible lost resource.
Corrective Action Taken:	The alarm fields are cleared to indicate that no alarm is present.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The red_alm field of the rSMEST tuple.
<i>Bad Data</i> =	The ylw_alm field of the rSMEST tuple.
<i>Good Data</i> =	The bas_stat field of the rSMEST tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 38 Error Code: KEY

Error Code:	KEY
Error Description:	A rSMEST tuple contains a bad logical key.
Possible Error Effect:	Use of this tuple will propagate errors.

Corrective Action Taken:	The audit reinitializes the rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Logical key of rSMEST.
<i>Good Data</i> =	Logical key of associated rCDBCOM tuple.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST.

Table 39 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	No success in reading the tuple in RLSMEST to see if DCLU is in service.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	Request DFISHELF audit indirectly through T1USG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to DFI tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Equipment status table of the unit.

Table 40 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The rSMEST tuple indicates that the circuit corresponds to a live process, but it is not maintenance busy.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process which claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 41 Error Code: LOST_GSDB

Error Code:	LOST_GSDB
Error Description:	A rIGSDB tuple has been found missing from the database for either a grid circuit (PCLCGRID, PCGRID), a SLC [®] 96 bank circuit (PCS96BNK), a SLC [®] 96 shelf circuit (PCS96SHF), or a line group controller circuit (PCLGC) that is marked equipped in the rICKTDATA tuple (eqstat != SMUNEQ).
Possible Error Effect:	All line ports on these circuits will not be able to originate a call.
Corrective Action Taken:	The rIGSDB tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIGSDB tuple found in error.
<i>Bad Data</i> =	The circuit type found in the rICKTDATA tuple (type).
<i>Good Data</i> =	The circuit name found in the rICKTDATA tuple (name).
<i>Logical Index</i> =	The logical key of the rIGSDB tuple found missing in the database.
<i>Dump</i> =	The rICKTDATA tuple associated with this rIGSDB tuple.

Table 42 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 43 Error Code: LOST_PTSB

Error Code:	LOST_PTSB
Error Description:	A PIDB circuit lacks one of its 32 corresponding rIPTSB tuples.
Possible Error Effect:	The time slot resource is lost to the system.
Corrective Action Taken:	The rIPTSB tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPTSB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Key of rIPTSB.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 44 Error Code: LOST_SMEST

Error Code:	LOST_SMEST
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Type of circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 45 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	A circuit lacks its corresponding rCDBCOM tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rCDBCOM tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type found.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 46 Error Code: LOSTTUP (1)

Error Code:	LOSTTUP (1)
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 47 Error Code: LOSTTUP (2)

Error Code:	LOSTTUP (2)
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.

Corrective Action Taken:	The rIORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 48 Error Code: LOSTTUP (3)

Error Code:	LOSTTUP (3)
Error Description:	The ttf tuple associated with this rICKTDATA tuple does not exist but should.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The ttf tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICKTDATA tuple.
<i>Bad Data =</i>	Associated ttf relation.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rICKTDATA circuit name.
<i>Dump =</i>	NA.

Table 49 Error Code: LOSTTUP (4)

Error Code:	LOSTTUP (4)
Error Description:	No success in reading the tuple from RLS96BASE to see if SG is in service.
Possible Error Effect:	No SLC [®] 96 call processing through this service group.
Corrective Action Taken:	Request BASE audit indirectly through PATHMAP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to DFI tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Equipment status table of the unit.

Table 50 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DFI facility and the carrier group alarm state. When there is a carrier group alarm, the facility should be marked out-of-service, auto, or removed but is not.
Possible Error Effect:	DFI facilities with CGAs will not be removed from service properly.
Corrective Action Taken:	The facility is removed and restored if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple in error.
<i>Bad Data =</i>	Facility basic state (rSMEST.bas_stat).
<i>Good Data =</i>	Out-of-service (4).
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump =</i>	rSMEST

Table 51 Error Code: MATE

Error Code:	MATE
Error Description:	The rSMEST tuple indicates that the mate of this circuit is in an invalid state. This is applicable to ACT/STBY type of circuits.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The circuit/mate is restored to correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Mate's bas_stat field in rSMEST tuple.
<i>Good Data =</i>	Circuit's bas_state field in rSMEST tuple.

<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Circuit's unit ID.
<i>Dump 2</i> =	Mate's unit ID.

Table 52 Error Code: MISSING

Error Code:	MISSING
Error Description:	Audit failed to read a rICDBCOM tuple for a DS0 circuit name for a DNU-S facility.
Possible Error Effect:	Possible failure of resource for this DS0 circuit.
Corrective Action Taken:	None. The audit will report this error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rICKTDATA tuple, name .
<i>Good Data</i> =	The circuit name for which audit could not get rICDBCOM tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 53 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The dynamic rPICB tuple associated with the static rICKTDATA tuple for this PIDB circuit is missing from the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PICB audit is scheduled to insert the missing rPICB tuple into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA in error.
<i>Bad Data</i> =	Key value of rPICB.
<i>Good Data</i> =	PICB value recalculated from circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rICKTDATA.
<i>Dump 2</i> =	rPICB.

Table 54 Error Code: MMP_IIA

Error Code:	MMP_IIA
Error Description:	State is allow but service request mask says inhibited.
Possible Error Effect:	No SLC [®] 96 call processing through this S_DFI.
Corrective Action Taken:	The service request mask will be corrected by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	The bit position that is wrong, start counting from 0.
<i>Dump</i> =	The equipment status table of the unit.

Table 55 Error Code: MP_HDW_IA

Error Code:	MP_HDW_IA
Error Description:	A rSMEST tuple has a bas_stat value only allowed for module processor equipment, and it isn't part of the module processor complex.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit will reinitialize the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Bad bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST tuple.

Table 56 Error Code: MRAQN_BITY

Error Code:	MRAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for recovery, yet it was not found to be on the queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.mra_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 57 Error Code: MRAQY_BITN

Error Code:	MRAQY_BITN
Error Description:	The circuit was found to be queued for restoration, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.mra_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 58 Error Code: MSCU_IIA

Error Code:	MSCU_IIA
Error Description:	Since mate service request source register is used, bit 15 of the own SRSR must be allowed.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	This bit 15 of the "own" SRSR is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	15.
<i>Dump</i> =	The equipment status table of the unit.

Table 59 Error Code: NO_CHDB

Error Code:	NO_CHDB
Error Description:	The rCHDB tuple accessed by the audit is not the same tuple audit tried to get. It failed the key check, that is, the key of the rCHDB tuple, chnum is not equal to the link field in the rCDBCOM tuple, cdccm.st_link minus the shift for the rCHDB tuple, DBSFCHDB. The missing rCHDB tuple represented a DS0 circuit in a DNU-S.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type field of the rCKTDATA tuple.
<i>Bad Data</i> =	The key field of rCHDB tuple, chnum .
<i>Good Data</i> =	The link field of rCDBCOM tuple, cdccm.st_link minus the shift value for rCHDB tuple, DBSFCHDB.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rCHDB tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 60 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous head cell (DBIDLMISC).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Idle link list physical key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump 1 =</i>	CKTDATA tuple.
<i>Dump 2 =</i>	CDBCOM tuple.

Table 61 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rSMEST tuple and in the F bits of the signaling information of the peripheral control and timing (PCT) links time slots. Audit found inconsistency between these informations when it compared it. The inconsistency could be in the form that the F bits says no alarm is present, but the rSMEST of the circuit says there is an alarm present or the types of alarm present are not the same in both places. For exact information as to what happened, see the explanation in error address field in this section.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The audit will call TM function to put the circuit on the CGA recovery queue where its alarm information will be correctly applied.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field represents the type of error that occurred. Depending upon its value different explanations are given here. 1 = The F bits says that there is no alarm present on this circuit but one or more of the alarm bits are set in the rSMEST tuple. 2 = The F bits says AIS alarm is present but the rSMEST says either AIS alarm is not set or one of the other alarms, red or yellow, is set. 3 = The F bits says red alarm is present but the rSMEST says either red alarm is not set or one of the other alarms, AIS or yellow, is set. 4 = The F bits says yellow alarm is present but the rSMEST says either yellow alarm is not set or one of the other alarms, AIS or red, is set. 5 = The audit received a value from TM that it is not expecting to receive.
<i>Bad Data =</i>	The local value of circuit type derived from the F bits, ckdtype .
<i>Good Data =</i>	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index =</i>	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The rCDBCOM tuple.

Table 62 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rSMEST tuple is not transient but its progress flag is set.

Possible Error Effect:	Audit transient timing may operate improperly.
Corrective Action Taken:	The progress flag is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST tuple.

Table 63 Error Code: NOTSPARE

Error Code:	NOTSPARE
Error Description:	An rSMEST tuple has a bas_stat value only allowed for module processor and remote clock equipment.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The tuple is reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Value of rSMEST.bas_stat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST for parent/mate (OPTIONAL).

Table 64 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rSMEST tuple and in the F bits of the signaling information of the PCT links time slots. The audit found an inconsistency when it compared the CGA alarm bits in the rSMEST tuple against the bas_stat and qual_2 fields in that tuple. These two fields indicate the base state of the facility represented by the rSMEST tuple. The alarm bits in the rSMEST tuple (ais_alm , red_alm , or ylw_alm) should only be set if the base state indicates either out of service family equipment (OOS/FE) or out of service facility failure (OOS/FAF). In any other case the alarm bits should not be set.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The alarm bits (ais_alm , red_alm , or ylw_alm) are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The local value of circuit type derived from the F bits, cktttype .
<i>Good Data =</i>	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index =</i>	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The rCDBCOM tuple.

Table 65 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	No rISLUPIDB tuple exists for this PCPIDBISLU or PCPIDBRISLU circuit.
Possible Error Effect:	This resource is lost to the system.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PICB value used to read rISLUPIDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 66 Error Code: ODDLOSTTUP (1)

Error Code:	ODDLOSTTUP (1)
Error Description:	A tuple in the static relation RLDSLEQUIP is missing.
Possible Error Effect:	Full use of DSL equipment is impossible.
Corrective Action Taken:	None.
Manual Action Required:	A tuple should be inserted into rIDSLEQUIP as appropriate for the missing PCRDTTMC or PCRDTTEOC circuit.
Dump Description:	
<i>Error Address</i> =	Address of rIDSLEQUIP buffer.
<i>Bad Data</i> =	Return value from DB (DBNO MATCH).
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA.

Table 67 Error Code: ODDLOSTTUP (2)

Error Code:	ODDLOSTTUP (2)
Error Description:	The CKTDATA audit is attempting to read a nonexistent rIDCKT2GP tuple. The key to the tuple is derived from the rICKTDATA tuple circuit name and circuit index.
Possible Error Effect:	Call processing cannot use the circuit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA tuple.

Table 68 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The circuit is operational and out-of-service but qual_1 and qual_2 do not have compatible values.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 69 Error Code: PARENT

Error Code:	PARENT
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process nor is its parent circuit busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	Parent's rSMEST.

Table 70 Error Code: PCD

Error Code:	PCD
Error Description:	The audit was not able to read the rVPIDB_F tuple for a corresponding either the STE circuit or the STS-1 circuit for a DNU-S. Refer to <i>Good Data</i> field to determine for which type of circuit, STE or STS-1, audit could not read the rVPIDB_F tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report this error until the rVPID_F tuple is inserted.
Manual Action Required:	There are three key field for the rVPIDB_F tuple, unitttype , unitnumber and pct_link . Use the value of utype and unit from the rCKTDATA dump of this error for the first two keys of the rVPIDB_F tuple, unitttype and unitnumber respectively. Use the value of key field for the third key of the rVPIDB_F tuple, pct_link and insert this rVPIDB_F tuple.
Dump Description:	
<i>Error Address</i> =	The unit field of the rCKTDATA tuple.
<i>Bad Data</i> =	Contents of the DS0 circuit name calculated by audit if any.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The pct_link key of the rVPIDB_F tuple as calculated by audit from the given data.
<i>Dump 1</i> =	The rCKTDATA tuple, either for STE or STS-1.
<i>Dump 2</i> =	The rVPIDB_F tuple as returned by database read.

Table 71 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the rCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 72 Error Code: PICBKEY

Error Code:	PICBKEY
Error Description:	The static relation RLISLUPIDB is missing a tuple.
Possible Error Effect:	The tuple is not available for access.
Corrective Action Taken:	None.
Manual Action Required:	The tuple should be inserted into the static relation RLISLUPIDB.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value.
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 73 Error Code: PID

Error Code:	PID
Error Description:	The PID in the tp_pid field is not OSnullpid nor a live PID and thus is garbage.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The PID is set to OSnullpid.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 74 Error Code: QUAL

Error Code:	QUAL
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but qual_1 indicates the circuit is growth or unequipped and not on the automatic or manual request queues.
Possible Error Effect:	The circuit may be erroneously processed by MRA or lost to call processing.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 75 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The circuit is operational and out-of-service, but qual_2 has an invalid value.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_2 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	Cross reference entry.
<i>Dump 3</i> =	Queue entry.
<i>Dump 4</i> =	rSMEST tuple under audit.
<i>Dump 5</i> =	Parent of the rSMEST tuple under audit.

Table 76 Error Code: QUAL_EQPG

Error Code:	QUAL_EQPG
Error Description:	The circuit is growth or special growth, but qual_1 is not equal to SMGR or SMSGR.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qual_1 field is corrected to SMGR or SMSGR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 77 Error Code: QUE

Error Code:	QUE
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but the circuit does not have an entry in the queue specified by qual_1 (auto or man).
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested otherwise; qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Identity of queue in which the circuit was found.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 78 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	A reconfiguration request of a remote clock circuit was not successful.
Possible Error Effect:	Remote clocking may be impaired.
Corrective Action Taken:	All remote clock circuits generating this error, except the parent circuit, are marked out-of-service or active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Return value of the configuration routine.
<i>Good Data</i> =	Action performed on the circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of parent circuit.
<i>Dump 2</i> =	rSMEST of subtending or mate side to a remote clock side.

Table 79 Error Code: READ

Error Code:	READ
Error Description:	Remote clock hardware can not be read at this time.
Possible Error Effect:	Can not determine active RCLK.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of mate's rSMEST.bas_stat field.
<i>Good Data</i> =	Value of circuit's rSMEST.bas_stat field.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of circuit.
<i>Dump 2</i> =	rSMEST of mate circuit.

Table 80 Error Code: RT2DS1_LOST (1)

Error Code:	RT2DS1_LOST (1)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCUs facilities.
Corrective Action Taken:	None.
Manual Action Required:	A tuple in RLRT2DS1 should be inserted as appropriate for the circuit identified in the associated audit dump.
Dump Description:	
<i>Error Address</i> =	Address of RLRT2DS1 buffer.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rCKTDATA.

Table 81 Error Code: RT2DS1_LOST (2)

Error Code:	RT2DS1_LOST (2)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCU facilities.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of RLRT2DS1 buffer.
<i>Bad Data</i> =	Shelf number of RT circuit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA.

Table 82 Error Code: SCNBAD

Error Code:	SCNBAD
Error Description:	An inconsistency has been detected in the value of the IDCU register HWMLSCAN: either it is set when it should not be or vice versa. If the IDCUCOM is active, the bit should be set. If the IDCUCOM is stby, the bit should not be set.
Possible Error Effect:	The scanning state is incorrect for the IDCUCOM identified.
Corrective Action Taken:	The LSCAN bit is reset to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	HWMLSCAN.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 83 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Audit tried to read a tuple of the static relation, RLFAC_PROV, as part of the CGA checks on DNU-S, but failed to read it.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until the appropriate rFAC_PROV tuple is inserted.
Manual Action Required:	Insert a rFAC_PROV tuple with the key equal to the circuit name dumped in either the logical key or error address fields. The same value is dumped in these fields from different data structures.
Dump Description:	
<i>Error Address</i> =	The name field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rFAC_PROV tuple, cktname , which audit could not access.
<i>Good Data</i> =	The circuit type from the type field of rICKTDATA tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rICKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 84 Error Code: SMEST

Error Code:	SMEST
Error Description:	The service status [in-service (0), out-of-service (1)] of the rSMEST tuple disagrees with that of other tuples corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name (or resources, if the name corresponds to a family of equipment) to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rSMEST tuple is forced idle and re-audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Service status (in/out) of rSMEST tuple.
<i>Good Data</i> =	Expected status.

<i>Logical Index</i> =	Internal name (rSMEST key).
<i>Dump</i> =	rSMEST tuple.

Table 85 Error Code: SSTBAD

Error Code:	SSTBAD
Error Description:	Register contents do not match the contents of the IDCUCOM rSMEST and the LSI(s) rCKTDATA.
Possible Error Effect:	The hardware state of the IDCU is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	Constructed mask indicating correct service state.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 86 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rSMEST tuple indicates that this circuit is idle, but it has an entry in one of the queues or the xrf table.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Cross reference entry.
<i>Dump 2</i> =	Queue entry.
<i>Dump 3</i> =	rSMEST tuple under audit.
<i>Dump 4</i> =	Parent of the rSMEST tuple under audit.

Table 87 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkusage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	linkusage (from cdbcom).
<i>Good Data</i> =	linkusage (expected).
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	Circuit type CDBCOM.

Table 88 Error Code: STATUS

Error Code:	STATUS
Error Description:	The rSMEST tuple indicates that this circuit is neither busy nor idle, thus the stat_act field is invalid.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 89 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rSMEST tuple has been stuck for several minutes in a transient state.
Possible Error Effect:	A process using the rSMEST tuple is stuck with allocated resources and some audit consistency checks are prevented.
Corrective Action Taken:	The rSMEST tuple is recovered to a non-transient state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST tuple.

Table 90 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	One or both RCLK circuits has been busy for at least 5 seconds under control of the GLPNRCLK process.
Possible Error Effect:	The circuit may be lost for further MRA requests.
Corrective Action Taken:	The RCLK circuits are further checked to verify RCLK configuration status and are recovered to a safe/consistent state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	MRA activity value rSMEST.stat_act.
<i>Good Data =</i>	DBIDLE
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST for busy RCLK.
<i>Dump 2 =</i>	rSMEST for mate RCLK.

Table 91 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rCDBCOM and its associated rTONECH tuple failed a point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCDBCOM tuple.
<i>Bad Data =</i>	Tone field in the rCDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rCDBCOM tuple.
<i>Dump 1 =</i>	rCDBCOM tuple.
<i>Dump 2 =</i>	rTONECH tuple.

Table 92 Error Code: TYPE

Error Code:	TYPE
Error Description:	The type of circuit that the CKTDATA audit is attempting to audit is either not present in its private circuit mapping array (AUcktreImap[]), or it is marked as invalid in the map.
Possible Error Effect:	The audit will continue to print the error each time it runs but will otherwise ignore the circuit. This is an audit error.
Corrective Action Taken:	None.
Manual Action Required:	Report the error with failure report (FR). Correcting this problem may require manual boot of the SM to restore the RLCKTDATA relation.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Circuit type.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 93 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The circuit is unequipped but the qualifier qual_1 is not SMUN.
Possible Error Effect:	The circuit is accessible but has no hardware, which will result in interrupts when it is used.
Corrective Action Taken:	Qualifier field is corrected to SMUN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Value of SMUN.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 94 Error Code: VERTICAL

Error Code:	VERTICAL
Error Description:	The maintenance state of a circuit (rSMEST.bas_stat) is vertically inconsistent with the maintenance state of its "parent" circuit.
Possible Error Effect:	The circuit may not be available for use.
Corrective Action Taken:	The child's rSMEST tuple is made consistent with the parent's.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple for the child circuit.
<i>Bad Data</i> =	Child's rSMEST.bas_stat.
<i>Good Data</i> =	Number of parent circuits.
<i>Logical Index</i> =	Child circuit name.
<i>Dump 1</i> =	The child's rSMEST tuple.
<i>Dump 2</i> =	All the parent circuit rSMESTs (may be several).

Table 95 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Actual idle list physical key.
<i>Good Data</i> =	Physical key of correct idle list.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	CDBCOM tuple.

Table 96 Error Code: XRFPID

Error Code:	XRFPID
Error Description:	The rSMEST tuple indicates that this circuit has a live PID and the circuit is in xrf or one of the queues, but the PID doesn't match that in the queue.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	xrf search results.
<i>Dump 2</i> =	manq, autoq search results.
<i>Dump 3</i> =	rSMEST.
<i>Dump 4</i> =	Dump of parent's rSMEST tuple.

CKTDATA-D

Software Release: 5E13 only

1. CIRCUIT DATA (CKTDATA) AUDIT

The circuit data audit (CKTDATA) verifies that the information within each of these structures is correct and that the data is consistent when compared with the other data structures. It also performs static-to-dynamic existence checks for all data to find lost dynamic tuples.

The CKTDATA audit has seven phases. The first phase calls the port linkage area audit (PORTLA) to check for the existence of trunk ports. The second phase checks the dynamic circuit data. The third phase checks the maintenance status of circuits. The fourth phase checks the consistency among the RLPORTLA, RLCDBCOM, and RLSMEST relations. The fifth phase checks the static-to-dynamic existence of PIDB time slot block relation (RLPTSB) tuples created for integrated services line unit (ISLU) and packet switching unit (PSU) peripheral interface data buses (PIDBs). The sixth phase performs the static-to-dynamic existence checks between the static RLCKTDATA relation and the dynamic RLGSDb relation. The seventh phase performs integrity checks between the RLCKTDATA relation and the peripheral control channel allocation maps (PCchalloc[]) for classic SMs. Additional checks are made for ring generator bus data (RLRGBUS), DS0 data blocks (RLDS0B), and Maintenance Request Administrator global buffers.

This audit is part of the peripheral control audits subgroup. It runs at routine level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The CKTDATA audit uses the circuit data relation (RLCKTDATA), a static relation, to verify information in the following dynamic data structures in the SMs: the peripheral control channel allocation map (PCchalloc[]), the port linkage area relation (RLPORTLA), the circuit data block common area relation (RLCDBCOM), the group summary data block relation (RLGSDb), the automatic queue relation (RLAUTOQ), the switch maintenance cross reference relation (RLSMXRF) and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these data structures.

3. AUDIT CHECKS

The circuit data audit uses the static driver (DRSTATIC) to step through the write-protected RLCKTDATA and retrieve tuples. The driver first determines which circuits exist and what data is available about them. It then calls the appropriate phase of the audit to perform the following checks.

Audit Phase 1 = The PORTLA audit phase of this audit uses the RLPORTLA relation to check the trunk ports. Although both trunk and line ports have tuples in RLPORTLA, this phase of the CKTDATA audit checks only trunk ports. Each trunk port has a tuple in RLPORTLA that describes its status and associations. Refer to the PORTLA audit description for more information.

The CKTDATA audit calls the AUdoportla function to perform all of the internal checks on one portla.

Audit Phase 2 = The dynamic circuit data phase of the audit uses the RLCDBCOM relation to perform the following checks:

- The audit checks all point-to/point-back linkages to rLCDBCOM.
- The audit verifies that in-service circuits are on appropriate idle lists. In-service circuits not on idle lists cannot be used for call processing.
- The audit verifies that stable rLCDBCOM tuples have all required linkages.

- The audit verifies that idle rCDBCOM tuples do not have any linkages except idle list linkages.

See also the circuit data block common area audit (CDBCOM).

Audit Phase 3

= The maintenance status phase of the audit uses the rSMEST relation to perform the following checks:

- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (rSMEST), the switch maintenance cross-reference relation (rSMXRF), or the automatic queue relation (rIAUTOQ).
- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The audit performs a child-to-parent (vertical) consistency check to ensure that subtending circuits are in the proper base state (ACT, OOS, etc.) with respect to the parent.
- The audit performs a horizontal check for circuits that have a mate circuit to ensure consistency. This is for circuits that are configured ACT/STBY or ACT MAJOR/ACT MINOR.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (RLSMEST), the switch maintenance cross-reference relation (RLSMXRF), or the automatic queue relation (RLAUTOQ).

- At the end of all the maintenance related checks, do the carrier group alarm (CGA) states checks.

See also the switch maintenance equipment status table audit (SMEST).

- Audit Phase 4** = The consistency check phase of the audit verifies that the maintenance status for a given circuit (in-service or out-of-service) is consistent throughout the rIORTLA, rICDBCOM, and rSMEST relations.
- Audit Phase 5** = After all consistency checks are performed on the circuit, the audit checks to see if the circuit has dynamic rIPTSB tuples associated with it. Only PIDB types for ISLU, AIU, RISLU, PSU and some transport applications have rIPTSBs. For these circuit types, the audit verifies that there are no lost rIPTSB tuples.
- Audit Phase 6** = After all consistency checks are performed on the circuit, the audit then insures that dynamic rIGSDB tuples exist. The rIGSDB tuples will exist for grid (PCGRID), cost reduced grid (PCLCGRID), line group controller (PCLGC), SLC[®] 96 shelf (PCS96SHF), and SLC[®] 96 bank (PCS96BNK) circuits, remote terminal (PCRDT303) and some AIU line pack types that are marked equipped in the rCKTDATA tuple.
- Audit Phase 7** = At the completion of the audit checks will be made to insure the integrity of the usage counts (usecnt), allocation counts (allocnt), and ISLU/RISLU allocation counts (islu_alloc) in the peripheral control channel allocation map (PCchalloc[]).

During the running of the audit for every circuit in the RLCKTDATA relation the channel assignment (chnasn) field is examined. If the channel assignment is associated with an allocable channel (PCALLOCHAN) the usage and allocation counts are pegged in a local copy of the map (AUdbch_cnt[]). However, if this channel is an allocable channel and is associated with a RISLU/ISLU cut-through PIDB the RISLU/ISLU allocation count (islu_alloc) will be pegged in addition to the usage (usecnt) and allocation (allocnt) counts. If the channel assignment is associated with a nailed-up channel (PCINITCHAN) only the usage count is pegged in the local copy of the map. At the completion of the audit the local copy of the channel allocation map (AUdbch_cnt[]) is compared to the peripheral control channel allocation map (PCchalloc[]).

4. ERROR HANDLING

If any dynamic tuples are lost, the audit will report an error and reinsert them.

If an inconsistency was found between the local copy of the channel allocation map (AUdbch_cnt[]) and the peripheral control channel allocation map (PCchalloc[]), the peripheral control channel allocation map (PCchalloc[]) is updated to be consistent with the local copy of the map (AUdbch_cnt[]).

If any inconsistencies are found, the rIORTLA and rICDBCOM tuples are zeroed and placed on the appropriate idle lists.

Units allowed to use the MRA recovery queue will recover the rSMEST to a safe, consistent state and (if MRA restorable) mark the rSMEST tuple SMAUOOS. The units will interface with SMIM to load the circuit on the MRA queue, which will result in an MRA restoration of the circuit to get the hardware and all software back to a consistent state. If the rSMEST tuple for the circuit/parent/mate is already in the MRA recovery queue (marked SMAUOOS) due to previous audit recovery, the audit skips further recovery. Also, for circuits having rICDBCOM tuples, the CDBCOM is placed on the OOS list to keep call processing from accessing the circuit while on the MRA recovery queue.

Circuits that fail the vertical check will be recovered to the correct state. Circuits that fail the horizontal check will be

recovered using the MRA queue.

If a linkage between an rICDBCOM and an rICHDB tuple should exist but does not, an rICHDB tuple and a channel are allocated and linked to that rICDBCOM.

The audit produces the following error code(s).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The bit_access field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of incorrect entry in CMrcldata.
<i>Bad Data =</i>	CMrcldata.hdw[]_bit_access.
<i>Good Data =</i>	Correct access type based on the unit type 8 for DLTU, 16 for DLTU2.
<i>Logical Index =</i>	RCL circuit name.
<i>Dump =</i>	CMrcldata entry.

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An inconsistency has been detected between the dynamic data rISMEST CGA alarm fields and the actual state of the DFI hardware.
Possible Error Effect:	Alarms raised on DFI circuitry are not being served.
Corrective Action Taken:	Dynamic data is updated to agree with the DFI hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rISMEST in error.
<i>Bad Data =</i>	Bit map containing SMEST alarm fields: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Good Data =</i>	Bit map read from DFI hardware alarm: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rISMEST.

Table 3 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	An rICDBCOM tuple exists for a circuit that is unequipped.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rICDBCOM tuple is deleted.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rICDBCOM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICDBCOM tuple.
<i>Dump</i> =	rICDBCOM tuple.

Table 4 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rSMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rSMEST tuple.

Table 5 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 6 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	A circuit is marked as rSMEST.qual_2 = SMAUOOS but either is not on the maintenance request administrator's (MRA's) queue, or its parent is not on the queue.
Possible Error Effect:	The actual maintenance state of a circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	Circuit is SMAUOOS but not in queue (4).
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of the circuit.
<i>Dump 2</i> =	rSMEST of the parent circuit.

Table 7 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A IDC DS1 facility and its rSMEST.qual_2 is SMFAF, but either an alarm bit is not set, or the circuit is not on the MRA recovery queue.
Possible Error Effect:	The maintenance state of the circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.

<i>Bad Data</i> =	Value of rISMEST.qual 2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISMEST.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The unittyp field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .unittyp.
<i>Good Data</i> =	Unit type according to rICKTDATA.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An RLI tuple in the SMmccustat[] array was found to be OOS RMVD AUTO. This is not a legal state for the RLI.
Possible Error Effect:	The RLI will not be usable.
Corrective Action Taken:	The RLI will be recovered to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Value of rISMEST.qual 1.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISMEST.

Table 10 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The circuit has growth or special growth status but bas_stat is not equal to SMOOS.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	The bas_stat field is corrected to SMOOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rISMEST tuple.
<i>Dump 2</i> =	rISMEST tuple for parent/mate (OPTIONAL).

Table 11 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state is active restricted but the circuit is not a PCS96FAC or a PCDCLCC.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rISMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 12 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The global data identifying remote cluster link (RCL) information has been corrupted. The picb field is incorrect when compared to rICKTDATA for the RCL.
Possible Error Effect:	Intercluster communication may be impaired.
Corrective Action Taken:	The PICB number stored in the CMrcldata tuple will be set to the PICB derived from the rICKTDATA base address.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CMrcldata.hdw entry that is in error.
<i>Bad Data</i> =	The PICB value of the bad entry.
<i>Good Data</i> =	The PICB value derived from the base address in the rICKTDATA tuple.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	NA.

Table 13 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The circuit is operational and either in service or unavailable but the qualifier, qual 1, is not SMSNUL.
Possible Error Effect:	Circuit may be erroneously denied service if bad value is unequipped or some similar value.
Corrective Action Taken:	Qualifier field is corrected to SMSNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual 1 field.
<i>Good Data</i> =	Value of SMSNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 14 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The circuit is operational and either in service or unavailable, but the qualifier, qual 2, is not SMSTNUL or SMPROOS with bas_stat SMACT.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier field is corrected to SMSTNUL. If the circuit is the remote clock, qual_2 is set to the hardware value of the remote clock mode: SMSMRCNORM, SMRCFREE, SMRCFAST, or SMRCHOLD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual 2 field.
<i>Good Data</i> =	Value of SMSTNUL.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 15 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process and does not have a parent circuit which could be busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 16 Error Code: CAMA_IS

Error Code:	CAMA_IS
Error Description:	A CAMA-ONI keying trunk is incorrectly in service when its associated talking trunk is out-of-service.
Possible Error Effect:	The keying trunk will be hunted but it will not be possible to get control of the talking trunk. Per call failures will result whenever this CAMA-ONI trunk pair is hunted.
Corrective Action Taken:	The keying trunk is recovered to an OOS,FE,CAMA status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rPORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rPORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rPORTLA tuple for the talking trunk.

Table 17 Error Code: CAMA_OOS

Error Code:	CAMA_OOS
Error Description:	A CAMA-ONI keying trunk is incorrectly out-of-service (supposedly due to the talking trunk being out-of-service) when the associated talking trunk is in service.
Possible Error Effect:	The keying trunk will not be hunted and so this CAMA-ONI trunk pair will never be used by call processing.
Corrective Action Taken:	The keying trunk is recovered to an in service status provided its circuit status is in service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rPORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rPORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rPORTLA tuple for the talking trunk.

Table 18 Error Code: CCDUPFAIL

Error Code:	CCDUPFAIL
Error Description:	The array SMccdupfail[] duplex failure status does not match the service groups' rSMEST tuple status. The error could be either that the array has the line unit as being duplex failing while at least one of the service groups' rSMEST tuples are active or the array indicates that the line unit is not duplex failing while both service groups' rSMEST tuples are OOS.
Possible Error Effect:	The line unit will not have the correct IS/OOS status.
Corrective Action Taken:	The array will be corrected to reflect the correct status as per rSMEST tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	The status of the line unit as it appears in the SMdupfail[] array.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name of the service group.
<i>Dump 1</i> =	rSMEST tuple for service group zero.
<i>Dump 2</i> =	rSMEST tuple for the service group one.

Table 19 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The service status [in-service (0), out-of-service (1), or don't care (2)] of the rCDBCOM tuple disagrees with that of the rPORTLA tuple corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rCDBCOM tuple is placed on the idle link list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCDBCOM tuple
<i>Bad Data</i> =	Service status of rCDBCOM tuple.
<i>Good Data</i> =	Expected status.
<i>Logical Index</i> =	Internal name (rCDBCOM key).
<i>Dump</i> =	rCDBCOM tuple.

Table 20 Error Code: CGAQN_BITY

Error Code:	CGAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for carrier group alarm (CGA), yet it was not found to be on the TM queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The <i>rSMEST.tm_que</i> field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of <i>rSMEST.tm_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 21 Error Code: CGAQY_BITN

Error Code:	CGAQY_BITN
Error Description:	The circuit was found to be queued for carrier group alarm, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The <i>rSMEST.tm_que</i> field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of <i>rSMEST.tm_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 22 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	A cdbcom that should have a dedicated linkage to a chdb is not linked to a chdb.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and reinitializes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	cdbcom.

Table 23 Error Code: CKTCUT

Error Code:	CKTCUT
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the

	corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data =</i>	Value found in that address.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump =</i>	The whole array AUs96inhp[].

Table 24 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of start of CMrcldata.hdw.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 25 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in rCKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter-cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CMrcldata[].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 26 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	A discrepancy was found between the actual usage of rCDBCOM and rIPTSB tuples and the count of INITCHANs and ALLOCHANs stored in the channel allocation map (PCchalloc[]) maintained by peripheral control. The contents of the usage counts (usecnt), allocation counts (allocnt), and ISLU allocation counts (islu_alloc) between the peripheral control channel allocation map (PCchalloc[]) and the audit's copy of the channel allocation map (AUdbch_cnt[]) are found to be inconsistent.
Possible Error Effect:	Peripheral control may have problems allocating channels to calls. Also, peripheral control will have problems relocating idle code and audible ring channels during hardware growth.
Corrective Action Taken:	The channel allocation map maintained (PCchalloc[]) by peripheral control will be updated from the local copy of the channel allocation (AUdbch_cnt[]) map kept by audits.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The contents of the global flag (AUpc_upckt) which indicates if peripheral control has updated their copy of the channel allocation map.
<i>Logical Index</i> =	The PIDB timeslot (dbchan) or index into the channel allocation map that was found in error.
<i>Dump 1</i> =	The contents of the channel allocation map (PCchalloc[]) maintained by peripheral control.
<i>Dump 2</i> =	The contents of the channel allocation map (AUdbch_cnt[]) maintained by audits.

Table 27 Error Code: CURLNK

Error Code:	CURLNK
Error Description:	Audit found the rCDBCOM tuple with no rCHDB tuple attached to it for a DS0 circuit name in a DNU-S. The link field for the rCHDB tuple was found to have the null value, DALNKNUL.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>type</i> field of the rCKTDATA tuple.
<i>Bad Data</i> =	The link field of rCDBCOM tuple that was found to be null, <i>cdccm.st_link</i> .
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rCDBCOM tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 28 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rCDBCOM is associated with a system tone but is not on the DBCADGEN idle list.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCDBCOM tuple.
<i>Bad Data</i> =	The owner of the rCDBCOM tuple.
<i>Good Data</i> =	DBCADGEN head cell value.
<i>Logical Index</i> =	Logical key of the rCDBCOM tuple.
<i>Dump</i> =	rCDBCOM tuple.

Table 29 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The max_ddl_size field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[].max_ddl_size.
<i>Good Data</i> =	18 or 30.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 30 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	Inconsistencies have been detected in SMbufuse[] and/or SMquse[] global data used by the maintenance request administrator (MRA).
Possible Error Effect:	Possible problems scheduling MRA jobs.

Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return from SMbufaud().
<i>Good Data =</i>	0.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 31 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The rSMEST tuple indicates that the circuit is busy (rSMEST.tp_pid) for maintenance, but it is not associated with a valid rSMXRF tuple (rSMEST.smxrf_entry).
Possible Error Effect:	Possible that stable calls could be lost when the circuit is recovered or if the terminal process is purged.
Corrective Action Taken:	The circuit under audit is recovered which includes recovering the circuit's rSMEST tuple, purging the terminal process if necessary, and running the related SMXRF audit to recover the rSMXRF tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the dynamic rSMEST tuple under audit.
<i>Bad Data =</i>	The key of the rSMXRF tuple (rSMXRF.entry).
<i>Good Data =</i>	The key of the rSMXRF tuple according to the rSMEST tuple (rSMEST.smxrf_entry).
<i>Logical Index =</i>	The key of the rSMEST tuple (rSMEST.unit_id).
<i>Dump 1 =</i>	The dynamic rSMEST tuple under audit.
<i>Dump 2 =</i>	The dynamic rSMXRF tuple.

Table 32 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	An rIDS0B tuple is missing from the database.
Possible Error Effect:	The line using that rIDS0B tuple will not have any call processing capability.
Corrective Action Taken:	The missing rIDS0B tuple will be inserted in the database and linked to the correct list. The list is determined by the in-service or out-of-service state of the rSMEST's tuple for the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDS0B tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The key for the rIDS0B tuple.
<i>Logical Index =</i>	The circuit name of the DS1 facility.
<i>Dump =</i>	NA.

Table 33 Error Code: DS1PT_LOST

Error Code:	DS1PT_LOST
Error Description:	The audit detected that one of the rIDS1_PT tuple for a given IDCU DS1 facility is missing from the ODD. The rIDS1_PT tuple has two keys (first key is the rIDS1_PT.ds1name and the second key is the rIDS1_PT.key_slot). There should exist two tuple with the same first key but with different second key (zero and 12). The audit detected that one tuple was missing. Both tuples or none should be populated.
Possible Error Effect:	The nailup call processing using that tuple will not work.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should determine if the existing rIDS1_PT tuple should be deleted or the missing tuple should be inserted into the database.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS1_PT tuple.
<i>Bad Data =</i>	The return from the database read of the first half of the rIDS1_PT tuple where rIDS1_PT.key_slot=0. If it is a zero (GLSUCCESS), it means that the tuple

	existed, else the tuple was missing.
<i>Good Data</i> =	The return from the database read of the second half of the rIDS1_PT tuple where rIDS1_PT.key_slot=12. If it is zero, it means that the tuple existed, else the tuple was missing.
<i>Logical Index</i> =	Circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the rIDS1_PT tuple.

Table 34 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SMIM macros to read circuit name from one circuit to other circuit failed to read static relations, either RLCKTDATA or RLUNIT_CKT. Audit was trying to get the DS0 circuit name from any of the STE, STS1, VT1.5 or DS1 circuits for a DNU-S facility. Since this is a CKTDATA audit, the more likely missing tuple could be rLUNIT_CKT tuple for the DS0 circuit.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until either the appropriate rLCKTDATA or rLUNIT_CKT tuple is inserted.
Manual Action Required:	If rLCKTDATA tuple is missing, insert it with the key value dumped in the logical key field of audit message. For missing rLUNIT_CKT tuple, there are three keys. The unit key is dumped in the error address field. The utype key is dumped in the good data field. Audit does not have the information about the third key, ckt_idx, as this error was passed on to audit by SMIM code and that code calculated this circuit index value. Please contact any SMIM developer for this value.
Dump Description:	
<i>Error Address</i> =	The <i>unit</i> field of the rLCKTDATA tuple.
<i>Bad Data</i> =	The circuit name received by audit. This should be GLNULCKT.
<i>Good Data</i> =	The unit type from the <i>utype</i> field of rLCKTDATA tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rLCKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 35 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	The circuit is operational and out-of-service, but bas_stat has an invalid value.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad bas_stat field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 36 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	A BOOTSTRAP circuit (PCBTSR) on an SMP20 (switching module processor using <i>Motorola</i> ® 68020 processor), which is operational, has an rSMEST tuple which indicates that the circuit is not active.
Possible Error Effect:	Offline pump may fail for this switching module (SM).
Corrective Action Taken:	The rSMEST bas_stat is set to SMACT, qual_1 is set to SMSNUL, and qual_2 is set to SMSTNUL.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	Value of <i>bas_stat</i> in rSMEST.
<i>Good Data</i> =	The <i>smest</i> state value of <i>SMACT</i> =1.
<i>Logical Index</i> =	The circuit name.
<i>Dump 1</i> =	The rCKTDATA tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 37 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DNU-S facility and the carrier group alarm state. When the facility is marked active (ACT), there should be no carrier group alarm present.
Possible Error Effect:	The facility in the question may not be used, thus possible lost resource.
Corrective Action Taken:	The alarm fields are cleared to indicate that no alarm is present.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>red_alm</i> field of the rSMEST tuple.
<i>Bad Data</i> =	The <i>ylw_alm</i> field of the rSMEST tuple.
<i>Good Data</i> =	The <i>bas_stat</i> field of the rSMEST tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 38 Error Code: KEY

Error Code:	KEY
Error Description:	A rSMEST tuple contains a bad logical key.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit reinitializes the rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Logical key of rSMEST.
<i>Good Data</i> =	Logical key of associated rCDBCOM tuple.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST.

Table 39 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	No success in reading the tuple in RLSMEST to see if DCLU is in service.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	Request DFISHELF audit indirectly through T1USG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to DFI tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Equipment status table of the unit.

Table 40 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The rSMEST tuple indicates that the circuit corresponds to a live process, but it is not maintenance busy.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process which claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.

<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 41 Error Code: LOST_GSDB

Error Code:	LOST_GSDB
Error Description:	A rIGSDB tuple has been found missing from the database for either a grid circuit (PCLCGRID, PCGRID), a SLC [®] 96 bank circuit (PCS96BNK), a SLC [®] 96 shelf circuit (PCS96SHF), or a line group controller circuit (PCLGC) that is marked equipped in the rICKTDATA tuple (eqstat != SMUNEQ).
Possible Error Effect:	All line ports on these circuits will not be able to originate a call.
Corrective Action Taken:	The rIGSDB tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIGSDB tuple found in error.
<i>Bad Data</i> =	The circuit type found in the rICKTDATA tuple (type).
<i>Good Data</i> =	The circuit name found in the rICKTDATA tuple (name).
<i>Logical Index</i> =	The logical key of the rIGSDB tuple found missing in the database.
<i>Dump</i> =	The rICKTDATA tuple associated with this rIGSDB tuple.

Table 42 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 43 Error Code: LOST_PTSB

Error Code:	LOST_PTSB
Error Description:	A PIDB circuit lacks one of its 32 corresponding rIPTSB tuples.
Possible Error Effect:	The time slot resource is lost to the system.
Corrective Action Taken:	The rIPTSB tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPTSB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Key of rIPTSB.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 44 Error Code: LOST_SMEST

Error Code:	LOST_SMEST
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Type of circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 45 Error Code: LOSTCDB

Error Code:	LOSTCDB

Error Description:	A circuit lacks its corresponding rCDBCOM tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rCDBCOM tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit type found.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 46 Error Code: LOSTTUP (1)

Error Code:	LOSTTUP (1)
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit type.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 47 Error Code: LOSTTUP (2)

Error Code:	LOSTTUP (2)
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 48 Error Code: LOSTTUP (3)

Error Code:	LOSTTUP (3)
Error Description:	The ttf tuple associated with this rCKTDATA tuple does not exist but should.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The ttf tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCKTDATA tuple.
<i>Bad Data =</i>	Associated ttf relation.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rCKTDATA circuit name.
<i>Dump =</i>	NA.

Table 49 Error Code: LOSTTUP (4)

Error Code:	LOSTTUP (4)
Error Description:	No success in reading the tuple from RLS96BASE to see if SG is in service.
Possible Error Effect:	No SLC [®] 96 call processing through this service group.
Corrective Action Taken:	Request BASE audit indirectly through PATHMAP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to DFI tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Equipment status table of the unit.

Table 50 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DFI facility and the carrier group alarm state. When there is a carrier group alarm, the facility should be marked out-of-service, auto, or removed but is not.
Possible Error Effect:	DFI facilities with CGAs will not be removed from service properly.
Corrective Action Taken:	The facility is removed and restored if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple in error.
<i>Bad Data =</i>	Facility basic state (rSMEST.bas_stat).
<i>Good Data =</i>	Out-of-service (4).
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump =</i>	rSMEST

Table 51 Error Code: MAINT_STATE2

Error Code:	MAINT_STATE2
Error Description:	An inconsistency has been detected between the dynamic data RLSMEST basic status (rSMEST.bas_stat) or second qualifier (rSMEST.qual_2) for a packet switching unit common (PSUCOM) circuit and the data delivery (DD) key (DDPSUFRC) representing the forced status of the PSUCOM circuit.
Possible Error Effect:	If corruption occurs when the PSUCOM circuit is supposed to be in the forced state, the switch may try to switch to the unavailable PSUCOM service group which can cause the switch to escalate to a level of initialization.
Corrective Action Taken:	The PSUCOM circuit is removed and restored using the MRA recovery queue. As part of the restoration process the PSUCOM circuit will be made consistent with the DDPSUFRC DD key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple in error.
<i>Bad Data =</i>	Contents of the DDPSUFRC DD key.
<i>Good Data =</i>	Service group of PSUCOM circuit found in error.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	rSMEST tuple associated with the PSUCOM circuit.
<i>Dump 2 =</i>	rSMEST tuple associated with the mate PSUCOM circuit.

Table 52 Error Code: MATE

Error Code:	MATE
Error Description:	The rSMEST tuple indicates that the mate of this circuit is in an invalid state. This is applicable to ACT/STBY type of circuits.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The circuit/mate is restored to correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Mate's bas_stat field in rSMEST tuple.
<i>Good Data =</i>	Circuit's bas_state field in rSMEST tuple.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	Circuit's unit ID.
<i>Dump 2 =</i>	Mate's unit ID.

Table 53 Error Code: MISSING

Error Code:	MISSING
Error Description:	Audit failed to read a rCDBCOM tuple for a DS0 circuit name for a DNU-S facility.
Possible Error Effect:	Possible failure of resource for this DS0 circuit.
Corrective Action Taken:	None. The audit will report this error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The <i>type</i> field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rICKTDATA tuple, <i>name</i> .
<i>Good Data</i> =	The circuit name for which audit could not get rCDBCOM tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 54 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The dynamic rPICB tuple associated with the static rICKTDATA tuple for this PIDB circuit is missing from the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PICB audit is scheduled to insert the missing rPICB tuple into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA in error.
<i>Bad Data</i> =	Key value of rPICB.
<i>Good Data</i> =	PICB value recalculated from circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rICKTDATA.
<i>Dump 2</i> =	rPICB.

Table 55 Error Code: MMP_IJA

Error Code:	MMP_IJA
Error Description:	State is allow but service request mask says inhibited.
Possible Error Effect:	No SLC [®] 96 call processing through this S_DFI.
Corrective Action Taken:	The service request mask will be corrected by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	The bit position that is wrong, start counting from 0.
<i>Dump</i> =	The equipment status table of the unit.

Table 56 Error Code: MP_HDW_IA

Error Code:	MP_HDW_IA
Error Description:	A rSMEST tuple has a bas_stat value only allowed for module processor equipment, and it isn't part of the module processor complex.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit will reinitialize the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Bad bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST tuple.

Table 57 Error Code: MRAQN_BITY

Error Code:	MRAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for recovery, yet it was not found to be on the queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of rSMEST.mra_que field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 58 Error Code: MRAQY_BITN

Error Code:	MRAQY_BITN
Error Description:	The circuit was found to be queued for restoration, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The <i>rSMEST.mra_que</i> field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of <i>rSMEST.mra_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 59 Error Code: MSCU_IIA

Error Code:	MSCU_IIA
Error Description:	Since mate service request source register is used, bit 15 of the own SRSR must be allowed.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	This bit 15 of the "own" SRSR is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	15.
<i>Dump</i> =	The equipment status table of the unit.

Table 60 Error Code: NO_CHDB

Error Code:	NO_CHDB
Error Description:	The rCHDB tuple accessed by the audit is not the same tuple audit tried to get. It failed the key check, that is, the key of the rCHDB tuple, <i>chnum</i> is not equal to the link field in the rCDBCOM tuple, <i>cdccm.st_link</i> minus the shift for the rCHDB tuple, DBSFCHDB. The missing rCHDB tuple represented a DS0 circuit in a DNU-S.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>type</i> field of the rCKTDATA tuple.
<i>Bad Data</i> =	The key field of rCHDB tuple, <i>chnum</i> .
<i>Good Data</i> =	The link field of rCDBCOM tuple, <i>cdccm.st_link</i> minus the shift value for rCHDB tuple, DBSFCHDB.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	The rCHDB tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 61 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous head cell (DBIDLMISC).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Idle link list physical key.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump 1</i> =	CKTDATA tuple.

<i>Dump 2 =</i>	CDBCOM tuple.

Table 62 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rSMEST tuple and in the F bits of the signaling information of the peripheral control and timing (PCT) links time slots. Audit found inconsistency between these informations when it compared it. The inconsistency could be in the form that the F bits says no alarm is present, but the rSMEST of the circuit says there is an alarm present or the types of alarm present are not the same in both places. For exact information as to what happened, see the explanation in error address field in this section.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The audit will call TM function to put the circuit on the CGA recovery queue where its alarm information will be correctly applied.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field represents the type of error that occurred. Depending upon its value different explanations are given here. 1 = The F bits says that there is no alarm present on this circuit but one or more of the alarm bits are set in the rSMEST tuple. 2 = The F bits says AIS alarm is present but the rSMEST says either AIS alarm is not set or one of the other alarms, red or yellow, is set. 3 = The F bits says red alarm is present but the rSMEST says either red alarm is not set or one of the other alarms, AIS or yellow, is set. 4 = The F bits says yellow alarm is present but the rSMEST says either yellow alarm is not set or one of the other alarms, AIS or red, is set. 5 = The audit received a value from TM that it is not expecting to receive.
<i>Bad Data =</i>	The local value of circuit type derived from the F bits, <i>cktype</i> .
<i>Good Data =</i>	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index =</i>	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The rCDBCOM tuple.

Table 63 Error Code: NOTBUSY (2)

Error Code:	NOTBUSY (2)
Error Description:	An rSMEST tuple is not transient but its progress flag is set.
Possible Error Effect:	Audit transient timing may operate improperly.
Corrective Action Taken:	The progress flag is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST tuple.

Table 64 Error Code: NOTSPARE

Error Code:	NOTSPARE

Error Description:	An rSMEST tuple has a <i>bas_stat</i> value only allowed for module processor and remote clock equipment.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The tuple is reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST. <i>bas_stat</i> .
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST for parent/mate (OPTIONAL).

Table 65 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rSMEST tuple and in the F bits of the signaling information of the PCT links time slots. The audit found an inconsistency when it compared the CGA alarm bits in the rSMEST tuple against the <i>bas_stat</i> and <i>qual_2</i> fields in that tuple. These two fields indicate the base state of the facility represented by the rSMEST tuple. The alarm bits in the rSMEST tuple (<i>ais_alm</i> , <i>red_alm</i> , or <i>ylw_alm</i>) should only be set if the base state indicates either out of service family equipment (OOS/FE) or out of service facility failure (OOS/FAF). In any other case the alarm bits should not be set.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The alarm bits (<i>ais_alm</i> , <i>red_alm</i> , or <i>ylw_alm</i>) are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The local value of circuit type derived from the F bits, <i>ckttype</i> .
<i>Good Data</i> =	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index</i> =	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1</i> =	The rSMEST tuple.
<i>Dump 2</i> =	The rCDBCDBCOM tuple.

Table 66 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	No rISLUPIDB tuple exists for this PCPIDBISLU or PCPIDBRISLU circuit.
Possible Error Effect:	This resource is lost to the system.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value used to read rISLUPIDB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 67 Error Code: ODDLOSTTUP (1)

Error Code:	ODDLOSTTUP (1)
Error Description:	A tuple in the static relation RIDSLEQUIP is missing.
Possible Error Effect:	Full use of DSL equipment is impossible.
Corrective Action Taken:	None.
Manual Action Required:	A tuple should be inserted into rIDSLEQUIP as appropriate for the missing PCRDTTMC or PCRDEOC circuit.
Dump Description:	
<i>Error Address</i> =	Address of rIDSLEQUIP buffer.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	Circuit name.

<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA.

Table 68 Error Code: ODDLOSTTUP (2)

Error Code:	ODDLOSTTUP (2)
Error Description:	The CKTDATA audit is attempting to read a nonexistent rIDCKT2GP tuple. The key to the tuple is derived from the rICKTDATA tuple circuit name and circuit index.
Possible Error Effect:	Call processing cannot use the circuit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA tuple.

Table 69 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The circuit is operational and out-of-service but qual_1 and qual_2 do not have compatible values.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 70 Error Code: PARENT

Error Code:	PARENT
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process nor is its parent circuit busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	Parent's rSMEST.

Table 71 Error Code: PCD

Error Code:	PCD
Error Description:	The audit was not able to read the rVPIDB_F tuple for a corresponding either the STE circuit or the STS-1 circuit for a DNU-S. Refer to <i>Good Data</i> field to determine for which type of circuit, STE or STS-1, audit could not read the rVPIDB_F tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report this error until the rVPIDB_F tuple is inserted.
Manual Action Required:	There are three key field for the rVPIDB_F tuple, <i>unittype</i> , <i>unitnumber</i> and

	<i>pct_link</i> . Use the value of <i>utype</i> and <i>unit</i> from the rCKTDATA dump of this error for the first two keys of the rVPIDB_F tuple, <i>unittype</i> and <i>unitnumber</i> respectively. Use the value of key field for the third key of the rVPIDB_F tuple, <i>pct_link</i> and insert this rVPIDB_F tuple.
Dump Description:	
<i>Error Address</i> =	The <i>unit</i> field of the rCKTDATA tuple.
<i>Bad Data</i> =	Contents of the DS0 circuit name calculated by audit if any.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The <i>pct_link</i> key of the rVPIDB_F tuple as calculated by audit from the given data.
<i>Dump 1</i> =	The rCKTDATA tuple, either for STE or STS-1.
<i>Dump 2</i> =	The rVPIDB_F tuple as returned by database read.

Table 72 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the rCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 73 Error Code: PICBKEY

Error Code:	PICBKEY
Error Description:	The static relation RLISLUPIDB is missing a tuple.
Possible Error Effect:	The tuple is not available for access.
Corrective Action Taken:	None.
Manual Action Required:	The tuple should be inserted into the static relation RLISLUPIDB.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value.
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 74 Error Code: PID

Error Code:	PID
Error Description:	The PID in the <i>tp_pid</i> field is not OSnullpid nor a live PID and thus is garbage.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The PID is set to OSnullpid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 75 Error Code: QUAL

Error Code:	QUAL
Error Description:	The circuit is operational and out-of-service, <i>qual_2</i> indicates a diagnostic is

	running, but qual_1 indicates the circuit is growth or unequipped and not on the automatic or manual request queues.
Possible Error Effect:	The circuit may be erroneously processed by MRA or lost to call processing.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 76 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The circuit is operational and out-of-service, but qual_2 has an invalid value.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_2 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	Cross reference entry.
<i>Dump 3 =</i>	Queue entry.
<i>Dump 4 =</i>	rSMEST tuple under audit.
<i>Dump 5 =</i>	Parent of the rSMEST tuple under audit.

Table 77 Error Code: QUAL_EQPG

Error Code:	QUAL_EQPG
Error Description:	The circuit is growth or special growth, but qual_1 is not equal to SMGR or SMSGR.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qual_1 field is corrected to SMGR or SMSGR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 78 Error Code: QUE

Error Code:	QUE
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but the circuit does not have an entry in the queue specified by qual_1 (auto or man).
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	Identity of queue in which the circuit was found.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 79 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	A reconfiguration request of a remote clock circuit was not successful.
Possible Error Effect:	Remote clocking may be impaired.
Corrective Action Taken:	All remote clock circuits generating this error, except the parent circuit, are marked out-of-service or active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Return value of the configuration routine.
<i>Good Data =</i>	Action performed on the circuit.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST of parent circuit.
<i>Dump 2 =</i>	rSMEST of subtending or mate side to a remote clock side.

Table 80 Error Code: READ

Error Code:	READ
Error Description:	Remote clock hardware can not be read at this time.
Possible Error Effect:	Can not determine active RCLK.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Value of mate's rSMEST.bas_stat field.
<i>Good Data =</i>	Value of circuit's rSMEST.bas_stat field.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST of circuit.
<i>Dump 2 =</i>	rSMEST of mate circuit.

Table 81 Error Code: RT2DS1_LOST (1)

Error Code:	RT2DS1_LOST (1)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCUs facilities.
Corrective Action Taken:	None.
Manual Action Required:	A tuple in RLRT2DS1 should be inserted as appropriate for the circuit identified in the associated audit dump.
Dump Description:	
<i>Error Address =</i>	Address of RLRT2DS1 buffer.
<i>Bad Data =</i>	Return value from DB (DBNO_MATCH).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rCKTDATA.

Table 82 Error Code: RT2DS1_LOST (2)

Error Code:	RT2DS1_LOST (2)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCUs facilities.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of RLRT2DS1 buffer.
<i>Bad Data =</i>	Shelf number of RT circuit.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.

<i>Dump</i> =	rICKTDATA.

Table 83 Error Code: SCNBAD

Error Code:	SCNBAD
Error Description:	An inconsistency has been detected in the value of the IDCU register HWMLSCAN; either it is set when it should not be or vice versa. If the IDCUCOM is active, the bit should be set. If the IDCUCOM is stby, the bit should not be set.
Possible Error Effect:	The scanning state is incorrect for the IDCUCOM identified.
Corrective Action Taken:	The LSCAN bit is reset to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	HWMLSCAN.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISMEST.

Table 84 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Audit tried to read a tuple of the static relation, RLFAC_PROV, as part of the CGA checks on DNU-S, but failed to read it.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until the appropriate rIFAC_PROV tuple is inserted.
Manual Action Required:	Insert a rIFAC_PROV tuple with the key equal to the circuit name dumped in either the logical key or error address fields. The same value is dumped in these fields from different data structures.
Dump Description:	
<i>Error Address</i> =	The <i>name</i> field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rIFAC_PROV tuple, <i>cktname</i> , which audit could not access.
<i>Good Data</i> =	The circuit type from the <i>type</i> field of rICKTDATA tuple.
<i>Logical Index</i> =	Circuit name (rISMEST.unit_id).
<i>Dump 1</i> =	The rICKTDATA tuple.
<i>Dump 2</i> =	The rISMEST tuple.

Table 85 Error Code: SMEST

Error Code:	SMEST
Error Description:	The service status [in-service (0), out-of-service (1)] of the rISMEST tuple disagrees with that of other tuples corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name (or resources, if the name corresponds to a family of equipment) to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rISMEST tuple is forced idle and re-audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST tuple.
<i>Bad Data</i> =	Service status (in/out) of rISMEST tuple.
<i>Good Data</i> =	Expected status.
<i>Logical Index</i> =	Internal name (rISMEST key).
<i>Dump</i> =	rISMEST tuple.

Table 86 Error Code: SSTBAD

Error Code:	SSTBAD
Error Description:	Register contents do not match the contents of the IDCUCOM rISMEST and the LSI(s) rICKTDATA.
Possible Error Effect:	The hardware state of the IDCU is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	Constructed mask indicating correct service state.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 87 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rSMEST tuple indicates that this circuit is idle, but it has an entry in one of the queues or the xrf table.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Cross reference entry.
<i>Dump 2</i> =	Queue entry.
<i>Dump 3</i> =	rSMEST tuple under audit.
<i>Dump 4</i> =	Parent of the rSMEST tuple under audit.

Table 88 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkusage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	linkusage (from cdbcom).
<i>Good Data</i> =	linkusage (expected).
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	Circuit type CDBCOM.

Table 89 Error Code: STATUS

Error Code:	STATUS
Error Description:	The rSMEST tuple indicates that this circuit is neither busy nor idle, thus the stat_act field is invalid.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 90 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rSMEST tuple has been stuck for several minutes in a transient state.
Possible Error Effect:	A process using the rSMEST tuple is stuck with allocated resources and some audit consistency checks are prevented.
Corrective Action Taken:	The rSMEST tuple is recovered to a non-transient state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST tuple.

Table 91 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	One or both RCLK circuits has been busy for at least 5 seconds under control of the GLPNRCLK process.
Possible Error Effect:	The circuit may be lost for further MRA requests.
Corrective Action Taken:	The RCLK circuits are further checked to verify RCLK configuration status and are recovered to a safe/consistent state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	MRA activity value rSMEST.stat act.
<i>Good Data</i> =	DBIDLE
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST for busy RCLK.
<i>Dump 2</i> =	rSMEST for mate RCLK.

Table 92 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rCDBCOM and its associated rTONECH tuple failed a point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rCDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCDBCOM tuple.
<i>Bad Data</i> =	Tone field in the rCDBCOM tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rCDBCOM tuple.
<i>Dump 1</i> =	rCDBCOM tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 93 Error Code: TYPE

Error Code:	TYPE
Error Description:	The type of circuit that the CKTDATA audit is attempting to audit is either not present in its private circuit mapping array (AUcktreimap[]), or it is marked as invalid in the map.
Possible Error Effect:	The audit will continue to print the error each time it runs but will otherwise ignore the circuit. This is an audit error.
Corrective Action Taken:	None.
Manual Action Required:	Report the error with failure report (FR). Correcting this problem may require manual boot of the SM to restore the RLCKTDATA relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 94 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The circuit is unequipped but the qualifier qual_1 is not SMUN.
Possible Error Effect:	The circuit is accessible but has no hardware, which will result in interrupts when it is used.
Corrective Action Taken:	Qualifier field is corrected to SMUN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.

<i>Good Data</i> =	Value of SMUN.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 95 Error Code: VERTICAL

Error Code:	VERTICAL
Error Description:	The maintenance state of a circuit (rSMEST.bas_stat) is vertically inconsistent with the maintenance state of its "parent" circuit.
Possible Error Effect:	The circuit may not be available for use.
Corrective Action Taken:	The child's rSMEST tuple is made consistent with the parent's.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple for the child circuit.
<i>Bad Data</i> =	Child's rSMEST.bas_stat.
<i>Good Data</i> =	Number of parent circuits.
<i>Logical Index</i> =	Child circuit name.
<i>Dump 1</i> =	The child's rSMEST tuple.
<i>Dump 2</i> =	All the parent circuit rSMESTs (may be several).

Table 96 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Actual idle list physical key.
<i>Good Data</i> =	Physical key of correct idle list.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	CDBCOM tuple.

Table 97 Error Code: XRFPID

Error Code:	XRFPID
Error Description:	The rSMEST tuple indicates that this circuit has a live PID and the circuit is in xrf or one of the queues, but the PID doesn't match that in the queue.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	xrf search results.
<i>Dump 2</i> =	manq, autoq search results.
<i>Dump 3</i> =	rSMEST.
<i>Dump 4</i> =	Dump of parent's rSMEST tuple.

CKTDATA-E

Software Release: 5E14 and later

1. CIRCUIT DATA (CKTDATA) AUDIT

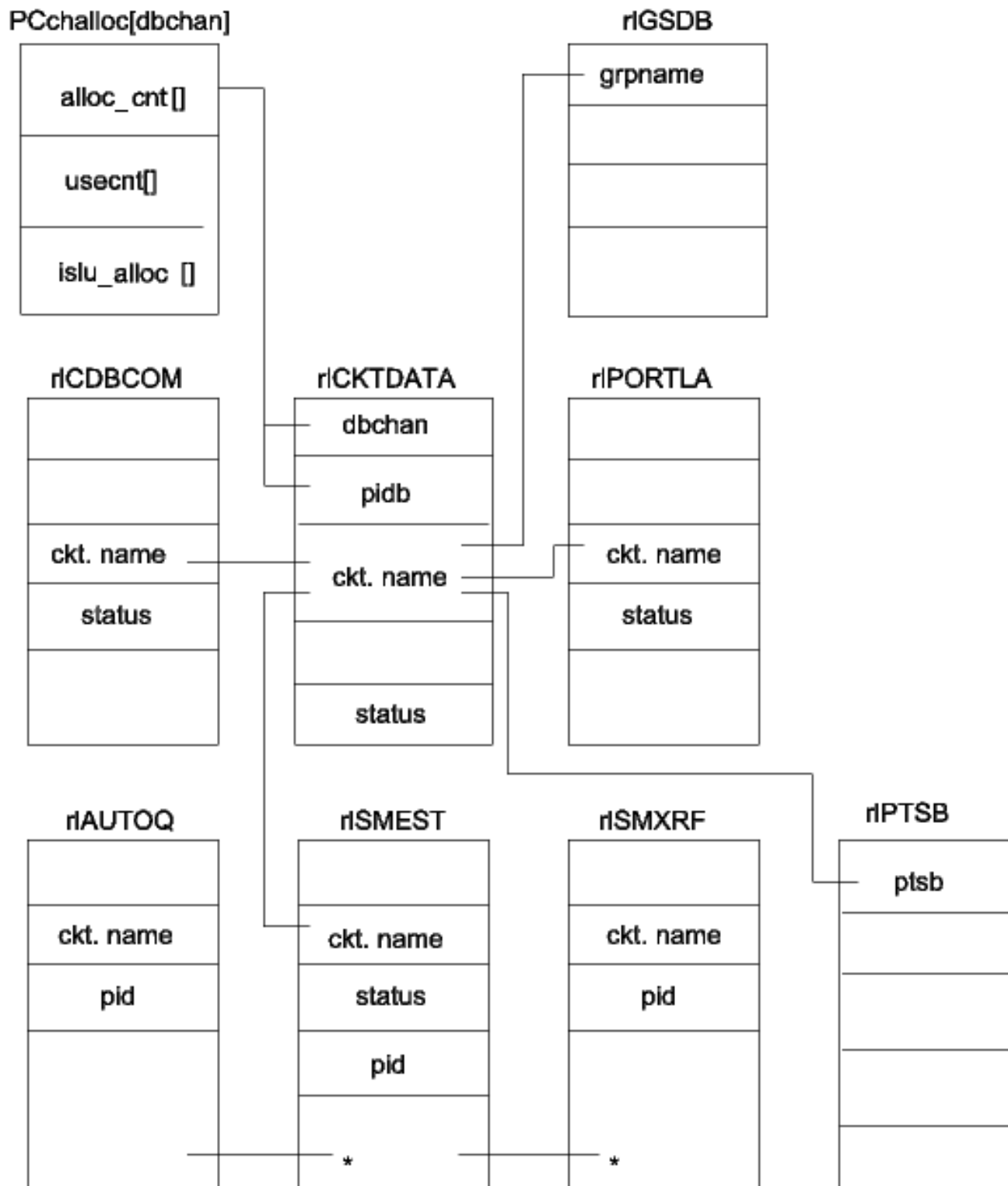
The circuit data audit (CKTDATA) verifies that the information within each of these structures is correct and that the data is consistent when compared with the other data structures. It also performs static-to-dynamic existence checks for all data to find lost dynamic tuples.

The CKTDATA audit has seven phases. The first phase calls the port linkage area audit (PORTLA) to check for the existence of trunk ports. The second phase checks the dynamic circuit data. The third phase checks the maintenance status of circuits. The fourth phase checks the consistency among the RLPORTLA, RLCDBCOM, and RLSMEST relations. The fifth phase checks the static-to-dynamic existence of PIDB time slot block relation (RLPTSB) tuples created for integrated services line unit (ISLU) and packet switching unit (PSU) peripheral interface data buses (PIDBs). The sixth phase performs the static-to-dynamic existence checks between the static RLCKTDATA relation and the dynamic RLGSDb relation. The seventh phase performs integrity checks between the RLCKTDATA relation and the peripheral control channel allocation maps (PCchalloc[]) for classic SMs. Additional checks are made for ring generator bus data (RLRGBUS), DS0 data blocks (RLDS0B), and Maintenance Request Administrator global buffers.

This audit is part of the peripheral control audits subgroup. It runs at routine level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The CKTDATA audit uses the circuit data relation (RLCKTDATA), a static relation, to verify information in the following dynamic data structures in the SMs: the peripheral control channel allocation map (PCchalloc[]), the port linkage area relation (RLPORTLA), the circuit data block common area relation (RLCDBCOM), the group summary data block relation (RLGSDb), the automatic queue relation (RLAUTOQ), the switch maintenance cross reference relation (RLSMXRF) and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these data structures.



———— = Implied linkage
 ————* = Semantic linkage

Figure 1 (CKTDATA-1) Data Structures of the CKTDATA Audit**3. AUDIT CHECKS**

The circuit data audit uses the static driver (DRSTATIC) to step through the write-protected RLCKTDATA and retrieve tuples. The driver first determines which circuits exist and what data is available about them. It then calls the appropriate phase of the audit to perform the following checks.

Audit Phase 1 = The PORTLA audit phase of this audit uses the RLPORTLA relation to check the trunk ports. Although both trunk and line ports have tuples in RLPORTLA, this phase of the CKTDATA audit checks only trunk ports. Each trunk port has a tuple in RLPORTLA that describes its status and associations. Refer to the PORTLA audit description for more information.

The CKTDATA audit calls the AUDoportla function to perform all of the internal checks on one portla.

Audit Phase 2 = The dynamic circuit data phase of the audit uses the RLCDBCOM relation to perform the following checks:

- The audit checks all point-to/point-back linkages to rLCDBCOM.
- The audit verifies that in-service circuits are on appropriate idle lists. In-service circuits not on idle lists cannot be used for call processing.
- The audit verifies that stable rLCDBCOM tuples have all required linkages.
- The audit verifies that idle rLCDBCOM tuples do not have any linkages except idle list linkages.

See also the circuit data block common area audit (CDBCOM).

Audit Phase 3 = The maintenance status phase of the audit uses the rSMEST relation to perform the following checks:

- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition under MRA control and the audit skips all SMEST checks.
- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (rSMEST), the switch maintenance cross-reference relation (rSMXRF), or the automatic queue relation (rAUTOQ).
- The audit checks to see if the transient bit is set (DBYES). If it is, the circuit is in transition

under MRA control and the audit skips all SMEST checks.

- The parent/mate circuit for the circuit under audit is obtained for possible SMXRF checks.
- The audit queries the MRA recovery queue (deferred work restoration queue) to see if the circuit/parent/mate is in the queue. If placed in the MRA queue by audits or system integrity, the qual_2 state will be set to SMAUOOS. Otherwise, MRA of PFR has placed the circuit in the queue. If marked SMAUOOS, the audit moves on to the next tuple. If not, the audit proceeds.
- The audit performs a child-to-parent (vertical) consistency check to ensure that subtending circuits are in the proper base state (ACT, OOS, etc.) with respect to the parent.
- The audit performs a horizontal check for circuits that have a mate circuit to ensure consistency. This is for circuits that are configured ACT/STBY or ACT MAJOR/ACT MINOR.
- The parent/mate circuit for the circuit under audit (if any) is obtained for possible SMXRF checks. The audit verifies the maintenance status information. It verifies that for each possible value of the base state item in rSMEST, the qualifiers have legal values.
- The audit verifies the maintenance activity information. If the maintenance activity is busy, a given process is active and the audit looks for process and circuit information in the switch maintenance equipment status table relation (RLSMEST), the switch maintenance cross-reference relation (RLSMXRF), or the automatic queue relation (RLAUTOQ).
- At the end of all the maintenance related checks, do the carrier group alarm (CGA) states checks.

See also the switch maintenance equipment status table audit (SMEST).

- Audit Phase 4** = The consistency check phase of the audit verifies that the maintenance status for a given circuit (in-service or out-of-service) is consistent throughout the rIORTLA, rCDBCOM, and rSMEST relations.
- Audit Phase 5** = After all consistency checks are performed on the circuit, the audit checks to see if the circuit has dynamic rIPTSB tuples associated with it. Only PIDB types for ISLU, AIU, RISLU, PSU and some transport applications have rIPTSBs. For these circuit types, the audit verifies that there are no lost rIPTSB tuples.
- Audit Phase 6** = After all consistency checks are performed on the circuit, the audit then insures that dynamic rIGSDB tuples exist. The rIGSDB tuples will exist for grid (PCGRID), cost reduced grid (PCLCGRID), line group controller (PCLGC), SLC[®] 96 shelf (PCS96SHF), and SLC[®] 96 bank (PCS96BNK) circuits, remote terminal (PCRDT303) and some AIU line pack types that are marked equipped in the rCKTDATA tuple.
- Audit Phase 7** = At the completion of the audit checks will be made to insure the integrity of the usage counts (usecnt), allocation counts (allocnt), and ISLU/RISLU allocation counts (islu_alloc) in the peripheral control channel allocation map (PCchalloc[]).

During the running of the audit for every circuit in the RLCKTDATA relation the channel assignment (chnasn) field is examined. If the channel assignment is associated with an allocable channel (PCALLOCHAN) the usage and allocation counts are pegged in a local copy of the map (AUdbch_cnt[]). However, if this channel is an allocable channel and is associated with a RISLU/ISLU cut-through PIDB the RISLU/ISLU allocation count (islu_alloc) will be

pegged in addition to the usage (usecnt) and allocation (allocnt) counts. If the channel assignment is associated with a nailed-up channel (PCINITCHAN) only the usage count is pegged in the local copy of the map. At the completion of the audit the local copy of the channel allocation map (Aldbch_cnt[]) is compared to the peripheral control channel allocation map (PCchalloc[]).

4. ERROR HANDLING

If any dynamic tuples are lost, the audit will report an error and reinsert them.

If an inconsistency was found between the local copy of the channel allocation map (Aldbch_cnt[]) and the peripheral control channel allocation map (PCchalloc[]), the peripheral control channel allocation map (PCchalloc[]) is updated to be consistent with the local copy of the map (Aldbch_cnt[]).

If any inconsistencies are found, the rPORTLA and rCDBCOM tuples are zeroed and placed on the appropriate idle lists.

Units allowed to use the MRA recovery queue will recover the rSMEST to a safe, consistent state and (if MRA restorable) mark the rSMEST tuple SMAUOOS. The units will interface with SMIM to load the circuit on the MRA queue, which will result in an MRA restoration of the circuit to get the hardware and all software back to a consistent state. If the rSMEST tuple for the circuit/parent/mate is already in the MRA recovery queue (marked SMAUOOS) due to previous audit recovery, the audit skips further recovery. Also, for circuits having rCDBCOM tuples, the CDBCOM is placed on the OOS list to keep call processing from accessing the circuit while on the MRA recovery queue.

Circuits that fail the vertical check will be recovered to the correct state. Circuits that fail the horizontal check will be recovered using the MRA queue.

If a linkage between an rCDBCOM and an rCHDB tuple should exist but does not, an rCHDB tuple and a channel are allocated and linked to that rCDBCOM.

The audit produces the following error code(s).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The bit_access field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of incorrect entry in CMrcldata.
<i>Bad Data =</i>	CMrcldata.hdw[] .bit_access.
<i>Good Data =</i>	Correct access type based on the unit type 8 for DLTU, 16 for DLTU2.
<i>Logical Index =</i>	RCL circuit name.
<i>Dump =</i>	CMrcldata entry.

Table 2 Error Code: ALARM

Error Code:	ALARM
Error Description:	An inconsistency has been detected between the dynamic data rSMEST CGA alarm fields and the actual state of the DFI hardware.
Possible Error Effect:	Alarms raised on DFI circuitry are not being served.
Corrective Action Taken:	Dynamic data is updated to agree with the DFI hardware.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rISMEST in error.
<i>Bad Data</i> =	Bit map containing SMEST alarm fields: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Good Data</i> =	Bit map read from DFI hardware alarm: 0x4 = red alarm. 0x2 = yellow alarm. 0x1 = AIS alarm.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISMEST.

Table 3 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	An rICDBCOM tuple exists for a circuit that is unequipped.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rICDBCOM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICDBCOM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICDBCOM tuple.
<i>Dump</i> =	rICDBCOM tuple.

Table 4 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rISMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rISMEST tuple.

Table 5 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 6 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	A circuit is marked as rSMEST.qual_2 = SMAUOOS but either is not on the maintenance request administrator's (MRA's) queue, or its parent is not on the queue.
Possible Error Effect:	The actual maintenance state of a circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	Circuit is SMAUOOS but not in queue (4).
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of the circuit.
<i>Dump 2</i> =	rSMEST of the parent circuit.

Table 7 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A IDCU DS1 facility and its rSMEST.qual_2 is SMFAF, but either an alarm bit is not set, or the circuit is not on the MRA recovery queue.
Possible Error Effect:	The maintenance state of the circuit is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The unitttype field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .unitttype.
<i>Good Data</i> =	Unit type according to rCKTDATA.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An RLI tuple in the SMmccustat[] array was found to be OOS RMVD AUTO. This is not a legal state for the RLI.
Possible Error Effect:	The RLI will not be usable.
Corrective Action Taken:	The RLI will be recovered to the correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST.qual_1.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 10 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The circuit has growth or special growth status but bas_stat is not equal to SMOOS.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	The bas_stat field is corrected to SMOOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad bas_stat field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 11 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state is active restricted but the circuit is not a PCS96FAC or a PCDCLCC.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad bas_stat field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 12 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The global data identifying remote cluster link (RCL) information has been corrupted. The picb field is incorrect when compared to rICKTDATA for the RCL.
Possible Error Effect:	Intercluster communication may be impaired.
Corrective Action Taken:	The PICB number stored in the CMrcldata tuple will be set to the PICB derived from the rICKTDATA base address.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CMrcldata.hdw entry that is in error.
<i>Bad Data =</i>	The PICB value of the bad entry.
<i>Good Data =</i>	The PICB value derived from the base address in the rICKTDATA tuple.
<i>Logical Index =</i>	RCL circuit name.
<i>Dump =</i>	NA.

Table 13 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The circuit is operational and either in service or unavailable but the qualifier, qual_1, is not SMSNUL.
Possible Error Effect:	Circuit may be erroneously denied service if bad value is unequipped or some similar value.
Corrective Action Taken:	Qualifier field is corrected to SMSNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	Value of SMSNUL.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.

<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 14 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The circuit is operational and either in service or unavailable, but the qualifier, qual_2, is not SMSTNUL or SMPROOS with bas_stat SMACT.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier field is corrected to SMSTNUL. If the circuit is the remote clock, qual_2 is set to the hardware value of the remote clock mode: SMSMRCNORM, SMRCFREE, SMRCFAST, or SMRCHOLD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_2 field.
<i>Good Data =</i>	Value of SMSTNUL.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 15 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The riSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process and does not have a parent circuit which could be busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The riSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	stat_act field in riSMEST tuple.
<i>Good Data =</i>	Value of DBIDLE (currently 1).
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 16 Error Code: CAMA_IS

Error Code:	CAMA_IS
Error Description:	A CAMA-ONI keying trunk is incorrectly in service when its associated talking trunk is out-of-service.
Possible Error Effect:	The keying trunk will be hunted but it will not be possible to get control of the talking trunk. Per call failures will result whenever this CAMA-ONI trunk pair is hunted.
Corrective Action Taken:	The keying trunk is recovered to an OOS,FE,CAMA status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the keying trunk riPORTLA tuple.
<i>Bad Data =</i>	bsc_state for the keying trunk.
<i>Good Data =</i>	bsc_state for the talking trunk.
<i>Logical Index =</i>	Port name of the keying trunk.
<i>Dump 1 =</i>	riPORTLA tuple for the keying trunk.
<i>Dump 2 =</i>	riPORTLA tuple for the talking trunk.

Table 17 Error Code: CAMA_OOS

Error Code:	CAMA_OOS
Error Description:	A CAMA-ONI keying trunk is incorrectly out-of-service (supposedly due to the talking trunk being out-of-service) when the associated talking trunk is in service.
Possible Error Effect:	The keying trunk will not be hunted and so this CAMA-ONI trunk pair will never be used by call processing.
Corrective Action Taken:	The keying trunk is recovered to an in service status provided its circuit status is in service.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the keying trunk rIORTLA tuple.
<i>Bad Data</i> =	bsc_state for the keying trunk.
<i>Good Data</i> =	bsc_state for the talking trunk.
<i>Logical Index</i> =	Port name of the keying trunk.
<i>Dump 1</i> =	rIORTLA tuple for the keying trunk.
<i>Dump 2</i> =	rIORTLA tuple for the talking trunk.

Table 18 Error Code: CCDUPFAIL

Error Code:	CCDUPFAIL
Error Description:	The array SMccdupfail[] duplex failure status does not match the service groups' rISMEST tuple status. The error could be either that the array has the line unit as being duplex failing while at least one of the service groups' rISMEST tuples are active or the array indicates that the line unit is not duplex failing while both service groups rISMEST tuples are OOS.
Possible Error Effect:	The line unit will not have the correct IS/OOS status.
Corrective Action Taken:	The array will be corrected to reflect the correct status as per rISMEST tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	The status of the line unit as it appears in the SMdupfail[] array.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name of the service group.
<i>Dump 1</i> =	rISMEST tuple for service group zero.
<i>Dump 2</i> =	rISMEST tuple for the service group one.

Table 19 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The service status [in-service (0), out-of-service (1), or don't care (2)] of the rICDBCOM tuple disagrees with that of the rIORTLA tuple corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rICDBCOM tuple is placed on the idle link list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICDBCOM tuple
<i>Bad Data</i> =	Service status of rICDBCOM tuple.
<i>Good Data</i> =	Expected status.
<i>Logical Index</i> =	Internal name (rICDBCOM key).
<i>Dump</i> =	rICDBCOM tuple.

Table 20 Error Code: CGAQN_BITY

Error Code:	CGAQN_BITY
Error Description:	The rISMEST tuple indicates that the circuit is queued for carrier group alarm (CGA), yet it was not found to be on the TM queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The <i>rISMEST.tm_que</i> field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISMEST tuple.
<i>Bad Data</i> =	Value of <i>rISMEST.tm_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rISMEST tuple.

Table 21 Error Code: CGAQY_BITN

Error Code:	CGAQY_BITN
Error Description:	The circuit was found to be queued for carrier group alarm, yet the appropriate field in rISMEST was not set.

Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The <i>rlSMEST.tm_que</i> field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rlSMEST tuple.
<i>Bad Data</i> =	Value of <i>rlSMEST.tm_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rlSMEST tuple.

Table 22 Error Code: CHDB_CDB

Error Code:	CHDB_CDB
Error Description:	A cdbcom that should have a dedicated linkage to a chdb is not linked to a chdb.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit recovers the cdbcom and reinitializes the cdbcom-to-chdb linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Circuit type.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	cdbcom.

Table 23 Error Code: CKTCUT

Error Code:	CKTCUT
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to store the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a <i>SLC</i> [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data</i> =	Value found in that address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump</i> =	The whole array AUs96inhp[].

Table 24 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL however, an RCL exists in rICKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of start of CMrcldata.hdw.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	Circuit name of the RCL.
<i>Logical Index</i> =	Circuit name of the RCL.
<i>Dump</i> =	NA.

Table 25 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an RCL exists in

	rICKTDATA which does not exist in CMrcldata.
Possible Error Effect:	Inter-cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CMrcldata[].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit name of the RCL.
<i>Logical Index =</i>	Circuit name of the RCL.
<i>Dump =</i>	NA.

Table 26 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	A discrepancy was found between the actual usage of rICDBCOM and rIPTSB tuples and the count of INITCHANS and ALLOCHANS stored in the channel allocation map (PCchalloc[]) maintained by peripheral control. The contents of the usage counts (usecnt), allocation counts (allocnt), and ISLU allocation counts (islu_alloc) between the peripheral control channel allocation map (PCchalloc[]) and the audit's copy of the channel allocation map (AUDbch_cnt[]) are found to be inconsistent.
Possible Error Effect:	Peripheral control may have problems allocating channels to calls. Also, peripheral control will have problems relocating idle code and audible ring channels during hardware growth.
Corrective Action Taken:	The channel allocation map maintained (PCchalloc[]) by peripheral control will be updated from the local copy of the channel allocation (AUDbch_cnt[]) map kept by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The contents of the global flag (AUpC_upckt) which indicates if peripheral control has updated their copy of the channel allocation map.
<i>Logical Index =</i>	The PIDB timeslot (dbchan) or index into the channel allocation map that was found in error.
<i>Dump 1 =</i>	The contents of the channel allocation map (PCchalloc[]) maintained by peripheral control.
<i>Dump 2 =</i>	The contents of the channel allocation map (AUDbch_cnt[]) maintained by audits.

Table 27 Error Code: CURLNK

Error Code:	CURLNK
Error Description:	Audit found the rICDBCOM tuple with no rICHDB tuple attached to it for a DS0 circuit name in a DNU-S. The link field for the rICHDB tuple was found to have the null value, DALNKNUL.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The <i>type</i> field of the rICKTDATA tuple.
<i>Bad Data =</i>	The link field of rICDBCOM tuple that was found to be null, <i>cdccm.st_link</i> .
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rICDBCOM tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 28 Error Code: DBCADGEN

Error Code:	DBCADGEN
Error Description:	The rICDBCOM is associated with a system tone but is not on the DBCADGEN idle list.
Possible Error Effect:	A system tone may be lost.

Corrective Action Taken:	The rICDBCOM will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICDBCOM tuple.
<i>Bad Data</i> =	The owner of the rICDBCOM tuple.
<i>Good Data</i> =	DBCADGEN head cell value.
<i>Logical Index</i> =	Logical key of the rICDBCOM tuple.
<i>Dump</i> =	rICDBCOM tuple.

Table 29 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The max_ddl_size field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .max_ddl_size.
<i>Good Data</i> =	18 or 30.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 30 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	Inconsistencies have been detected in SMbufuse[] and/or SMquse[] global data used by the maintenance request administrator (MRA).
Possible Error Effect:	Possible problems scheduling MRA jobs.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from SMbufaud().
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 31 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The rSMEST tuple indicates that the circuit is busy (rSMEST.tp_pid) for maintenance, but it is not associated with a valid rSMXRF tuple (rSMEST.smxrf_entry).
Possible Error Effect:	Possible that stable calls could be lost when the circuit is recovered or if the terminal process is purged.
Corrective Action Taken:	The circuit under audit is recovered which includes recovering the circuit's rSMEST tuple, purging the terminal process if necessary, and running the related SMXRF audit to recover the rSMXRF tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the dynamic rSMEST tuple under audit.
<i>Bad Data</i> =	The key of the rSMXRF tuple (rSMXRF.entry).
<i>Good Data</i> =	The key of the rSMXRF tuple according to the rSMEST tuple (rSMEST.smxrf_entry).
<i>Logical Index</i> =	The key of the rSMEST tuple (rSMEST.unit_id).
<i>Dump 1</i> =	The dynamic rSMEST tuple under audit.
<i>Dump 2</i> =	The dynamic rSMXRF tuple.

Table 32 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	An rDS0B tuple is missing from the database.

Possible Error Effect:	The line using that rIDS0B tuple will not have any call processing capability.
Corrective Action Taken:	The missing rIDS0B tuple will be inserted in the database and linked to the correct list. The list is determined by the in-service or out-of-service state of the rSMEST's tuple for the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDS0B tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The key for the rIDS0B tuple.
<i>Logical Index =</i>	The circuit name of the DS1 facility.
<i>Dump =</i>	NA.

Table 33 Error Code: DS1PT_LOST

Error Code:	DS1PT_LOST
Error Description:	The audit detected that one of the rIDS1_PT tuple for a given IDCU DS1 facility is missing from the ODD. The rIDS1_PT tuple has two keys (first key is the rIDS1_PT.ds1name and the second key is the rIDS1_PT.key_slot). There should exist two tuple with the same first key but with different second key (zero and 12). The audit detected that one tuple was missing. Both tuples or none should be populated.
Possible Error Effect:	The naluup call processing using that tuple will not work.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should determine if the existing rIDS1_PT tuple should be deleted or the missing tuple should be inserted into the database.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS1_PT tuple.
<i>Bad Data =</i>	The return from the database read of the first half of the rIDS1_PT tuple where rIDS1_PT.key_slot=0. If it is a zero (GLSUCCESS), it means that the tuple existed, else the tuple was missing.
<i>Good Data =</i>	The return from the database read of the second half of the rIDS1_PT tuple where rIDS1_PT.key_slot=12. If it is zero, it means that the tuple existed, else the tuple was missing.
<i>Logical Index =</i>	Circuit name of the DS1 facility.
<i>Dump =</i>	The contents of the rIDS1_PT tuple.

Table 34 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SMIM macros to read circuit name from one circuit to other circuit failed to read static relations, either RLCKTDATA or RLUNIT_CKT. Audit was trying to get the DS0 circuit name from any of the STE, STS1, VT1.5 or DS1 circuits for a DNU-S facility. Since this is a CKTDATA audit, the more likely missing tuple could be rLUNIT_CKT tuple for the DS0 circuit.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until either the appropriate rCKTDATA or rLUNIT_CKT tuple is inserted.
Manual Action Required:	If rCKTDATA tuple is missing, insert it with the key value dumped in the logical key field of audit message. For missing rLUNIT_CKT tuple, there are three keys. The unit key is dumped in the error address field. The utype key is dumped in the good data field. Audit does not have the information about the third key, ckt_indx, as this error was passed on to audit by SMIM code and that code calculated this circuit index value. Please contact any SMIM developer for this value.
Dump Description:	
<i>Error Address =</i>	The <i>unit</i> field of the rCKTDATA tuple.
<i>Bad Data =</i>	The circuit name received by audit. This should be GLNULCKT.
<i>Good Data =</i>	The unit type from the <i>utype</i> field of rCKTDATA tuple.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 35 Error Code: IDLE_CODE

Error Code:	IDLE_CODE
Error Description:	The fiber common data (FCD) idle code register on the peripheral control and timing link interface (PCTLI) device is inconsistent with the idle code constant (TS_IDLECD).
Possible Error Effect:	Call failures may occur since the peripheral control and timing data exchanger unit (PDXU) is not configured properly.
Corrective Action Taken:	The FCD idle code register on the PCTLI device will be made consistent with the idle code constant (TS_IDLECD).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the associated rSMEST tuple.
<i>Bad Data =</i>	Contents of the FCD idle code register.
<i>Good Data =</i>	The value of TS_IDLECD.
<i>Logical Index =</i>	Service group number in the upper bits 8 through 16 and time slot interface (TSI) link number in bits 0 through 7.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rCKTDATA tuple.

Table 36 Error Code: INV_BAS_STAT

Error Code:	INV_BAS_STAT
Error Description:	The circuit is operational and out-of-service, but bas_stat has an invalid value.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. If the circuit is the remote clock, reconfiguration is requested if the bas_stat field in rSMEST is out-of-range. Only the qualifiers are corrected otherwise.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	Bad bas_stat field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 37 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	A BOOTSTRAP circuit (PCBTSR) on an SMP20 (switching module processor using <i>Motorola</i> ® 68020 processor), which is operational, has an rSMEST tuple which indicates that the circuit is not active.
Possible Error Effect:	Offline pump may fail for this switching module (SM).
Corrective Action Taken:	The rSMEST bas_stat is set to SMACT, qual_1 is set to SMSNUL, and qual_2 is set to SMSTNUL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	Value of bas_stat in rSMEST.
<i>Good Data =</i>	The smest state value of SMACT=1.
<i>Logical Index =</i>	The circuit name.
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 38 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DNU-S facility and the carrier group alarm state. When the facility is marked active (ACT), there should be no carrier group alarm present.
Possible Error Effect:	The facility in the question may not be used, thus possible lost resource.

Corrective Action Taken:	The alarm fields are cleared to indicate that no alarm is present.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The <i>red_alm</i> field of the rISMEST tuple.
<i>Bad Data =</i>	The <i>ylw_alm</i> field of the rISMEST tuple.
<i>Good Data =</i>	The <i>bas_stat</i> field of the rISMEST tuple.
<i>Logical Index =</i>	Circuit name (rISMEST.unit_id).
<i>Dump =</i>	The rISMEST tuple.

Table 39 Error Code: KEY

Error Code:	KEY
Error Description:	A rISMEST tuple contains a bad logical key.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit reinitializes the rISMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rISMEST tuple.
<i>Bad Data =</i>	Logical key of rISMEST.
<i>Good Data =</i>	Logical key of associated rCDBCOM tuple.
<i>Logical Index =</i>	Logical key of rISMEST.
<i>Dump =</i>	rISMEST.

Table 40 Error Code: LAMP

Error Code:	LAMP
Error Description:	The FCD OOS register on the PCTLI device is set indicating that the circuit is OOS, but the maintenance status in the corresponding rISMEST tuple indicates that this PCT link circuit is in an in-service state.
Possible Error Effect:	May cause maintenance personnel to pull an active PCT link circuit pack due to the OOS lamp being on.
Corrective Action Taken:	The FCD OOS register on the PCTLI device will be cleared indicating that the PCT link circuit pack is in an in-service state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the corresponding rISMEST tuple.
<i>Bad Data =</i>	Contents of the FCD OOS register.
<i>Good Data =</i>	Correct value of the FCD OOS register.
<i>Logical Index =</i>	Service group number in the upper bits 8 through 16 and TSI link number in bits 0 through 7.
<i>Dump 1 =</i>	rISMEST tuple.
<i>Dump 2 =</i>	rICKTDATA tuple.

Table 41 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	No success in reading the tuple in RLSMEST to see if DCLU is in service.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	Request DFISHELF audit indirectly through T1USG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to DFI tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump =</i>	Equipment status table of the unit.

Table 42 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The rISMEST tuple indicates that the circuit corresponds to a live process, but it is not maintenance busy.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit resulting in a lost call.
Corrective Action Taken:	The process is purged.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process which claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 43 Error Code: LOST_GSDB

Error Code:	LOST_GSDB
Error Description:	A rGSDB tuple has been found missing from the database for either a grid circuit (PCLCGRID, PCGRID), a SLC [®] 96 bank circuit (PCS96BNK), a SLC [®] 96 shelf circuit (PCS96SHF), or a line group controller circuit (PCLGC) that is marked equipped in the rCKTDATA tuple (eqstat != SMUNEQ).
Possible Error Effect:	All line ports on these circuits will not be able to originate a call.
Corrective Action Taken:	The rGSDB tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSDB tuple found in error.
<i>Bad Data</i> =	The circuit type found in the rCKTDATA tuple (type).
<i>Good Data</i> =	The circuit name found in the rCKTDATA tuple (name).
<i>Logical Index</i> =	The logical key of the rGSDB tuple found missing in the database.
<i>Dump</i> =	The rCKTDATA tuple associated with this rGSDB tuple.

Table 44 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 45 Error Code: LOST_PTSTB

Error Code:	LOST_PTSTB
Error Description:	A PIDB circuit lacks one of its 32 corresponding rIPTSB tuples.
Possible Error Effect:	The time slot resource is lost to the system.
Corrective Action Taken:	The rIPTSB tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPTSB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Key of rIPTSB.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 46 Error Code: LOST_SMEST

Error Code:	LOST_SMEST
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Type of circuit.
<i>Logical Index</i> =	Circuit name.

<i>Dump</i> =	NA.

Table 47 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	A circuit lacks its corresponding rCDBCOM tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rCDBCOM tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type found.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 48 Error Code: LOSTTUP (1)

Error Code:	LOSTTUP (1)
Error Description:	A circuit lacks its corresponding rSMEST tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rSMEST tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit type.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 49 Error Code: LOSTTUP (2)

Error Code:	LOSTTUP (2)
Error Description:	A circuit lacks its corresponding rPORTLA tuple.
Possible Error Effect:	The circuit cannot be used.
Corrective Action Taken:	The rPORTLA tuple is created and inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 50 Error Code: LOSTTUP (3)

Error Code:	LOSTTUP (3)
Error Description:	The ttf tuple associated with this rCKTDATA tuple does not exist but should.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The ttf tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCKTDATA tuple.
<i>Bad Data</i> =	Associated ttf relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rCKTDATA circuit name.
<i>Dump</i> =	NA.

Table 51 Error Code: LOSTTUP (4)

Error Code:	LOSTTUP (4)
Error Description:	No success in reading the tuple from RLS96BASE to see if SG is in service.
Possible Error Effect:	No SLC [®] 96 call processing through this service group.
Corrective Action Taken:	Request BASE audit indirectly through PATHMAP.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Pointer to DFI tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Equipment status table of the unit.

Table 52 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	An inconsistency has been detected between the dynamic data rSMEST.bas_stat for a DFI facility and the carrier group alarm state. When there is a carrier group alarm, the facility should be marked out-of-service, auto, or removed but is not.
Possible Error Effect:	DFI facilities with CGAs will not be removed from service properly.
Corrective Action Taken:	The facility is removed and restored if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple in error.
<i>Bad Data</i> =	Facility basic state (rSMEST.bas_stat).
<i>Good Data</i> =	Out-of-service (4).
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	rSMEST

Table 53 Error Code: MAINT_STATE2

Error Code:	MAINT_STATE2
Error Description:	An inconsistency has been detected between the dynamic data RLSMEST basic status (rSMEST.bas_stat) or second qualifier (rSMEST.qual_2) for a packet switching unit common (PSUCOM) circuit and the data delivery (DD) key (DDPSUFRC) representing the forced status of the PSUCOM circuit.
Possible Error Effect:	If corruption occurs when the PSUCOM circuit is supposed to be in the forced state, the switch may try to switch to the unavailable PSUCOM service group which can cause the switch to escalate to a level of initialization.
Corrective Action Taken:	The PSUCOM circuit is removed and restored using the MRA recovery queue. As part of the restoration process the PSUCOM circuit will be made consistent with the DDPSUFRC DD key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple in error.
<i>Bad Data</i> =	Contents of the DDPSUFRC DD key.
<i>Good Data</i> =	Service group of PSUCOM circuit found in error.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump 1</i> =	rSMEST tuple associated with the PSUCOM circuit.
<i>Dump 2</i> =	rSMEST tuple associated with the mate PSUCOM circuit.

Table 54 Error Code: MATE

Error Code:	MATE
Error Description:	The rSMEST tuple indicates that the mate of this circuit is in an invalid state. This is applicable to ACT/STBY type of circuits.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The circuit/mate is restored to correct state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Mate's bas_stat field in rSMEST tuple.
<i>Good Data</i> =	Circuit's bas_state field in rSMEST tuple.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	Circuit's unit ID.
<i>Dump 2</i> =	Mate's unit ID.

Table 55 Error Code: MISSING

Error Code:	MISSING

Error Description:	Audit failed to read a rICDBCOM tuple for a DS0 circuit name for a DNU-S facility.
Possible Error Effect:	Possible failure of resource for this DS0 circuit.
Corrective Action Taken:	None. The audit will report this error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>type</i> field of the rICKTDATA tuple.
<i>Bad Data</i> =	The key of the rICKTDATA tuple, <i>name</i> .
<i>Good Data</i> =	The circuit name for which audit could not get rICDBCOM tuple.
<i>Logical Index</i> =	Circuit name (rSMEST.unit_id).
<i>Dump</i> =	The rSMEST tuple.

Table 56 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The dynamic rPICB tuple associated with the static rICKTDATA tuple for this PIDB circuit is missing from the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PICB audit is scheduled to insert the missing rPICB tuple into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA in error.
<i>Bad Data</i> =	Key value of rPICB.
<i>Good Data</i> =	PICB value recalculated from circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rICKTDATA.
<i>Dump 2</i> =	rPICB.

Table 57 Error Code: MMP_IJA

Error Code:	MMP_IJA
Error Description:	State is allow but service request mask says inhibited.
Possible Error Effect:	No SLC [®] 96 call processing through this S_DFI.
Corrective Action Taken:	The service request mask will be corrected by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	The bit position that is wrong, start counting from 0.
<i>Dump</i> =	The equipment status table of the unit.

Table 58 Error Code: MP_HDW_IA

Error Code:	MP_HDW_IA
Error Description:	A rSMEST tuple has a bas_stat value only allowed for module processor equipment, and it isn't part of the module processor complex.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The audit will reinitialize the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Bad bas_stat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	rSMEST tuple.

Table 59 Error Code: MRAQN_BITY

Error Code:	MRAQN_BITY
Error Description:	The rSMEST tuple indicates that the circuit is queued for recovery, yet it was not found to be on the queue.
Possible Error Effect:	Circuit data can not be verified in this stuck transient state.
Corrective Action Taken:	The rSMEST.mra_que field is recovered to DBNO.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of <i>rSMEST.mra_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 60 Error Code: MRAQY_BITN

Error Code:	MRAQY_BITN
Error Description:	The circuit was found to be queued for restoration, yet the appropriate field in rSMEST was not set.
Possible Error Effect:	Circuit is due for recovery but looks stable.
Corrective Action Taken:	The <i>rSMEST.mra_que</i> field is recovered to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Value of <i>rSMEST.mra_que</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump</i> =	Circuit's rSMEST tuple.

Table 61 Error Code: MSCU_IJA

Error Code:	MSCU_IJA
Error Description:	Since mate service request source register is used, bit 15 of the own SRSR must be allowed.
Possible Error Effect:	No SLC [®] 96 call processing through this S-DFI.
Corrective Action Taken:	This bit 15 of the "own" SRSR is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the service request mask.
<i>Bad Data</i> =	The value of the service request mask found to be wrong.
<i>Good Data</i> =	The correct value of the service request mask.
<i>Logical Index</i> =	15.
<i>Dump</i> =	The equipment status table of the unit.

Table 62 Error Code: NO_CHDB

Error Code:	NO_CHDB
Error Description:	The rCHDB tuple accessed by the audit is not the same tuple audit tried to get. It failed the key check, that is, the key of the rCHDB tuple, <i>chnum</i> is not equal to the link field in the rCDBCOM tuple, <i>cdccm.st_link</i> minus the shift for the rCHDB tuple, DBSFCHDB. The missing rCHDB tuple represented a DS0 circuit in a DNU-S.
Possible Error Effect:	Possible resource failure of the DS0 circuit.
Corrective Action Taken:	None. The audit will report the error and quit as it can not continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>type</i> field of the rCKTDATA tuple.
<i>Bad Data</i> =	The key field of rCHDB tuple, <i>chnum</i> .
<i>Good Data</i> =	The link field of rCDBCOM tuple, <i>cdccm.st_link</i> minus the shift value for rCHDB tuple, DBSFCHDB.
<i>Logical Index</i> =	Circuit name (<i>rSMEST.unit_id</i>).
<i>Dump 1</i> =	The rCHDB tuple.
<i>Dump 2</i> =	The rSMEST tuple.

Table 63 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	A bad cdbcom of a particular circuit type doesn't have an idle link list to be recovered to.
Possible Error Effect:	A circuit will not be recovered properly.
Corrective Action Taken:	The audit will recover the cdbcom to the idle-miscellaneous head cell (DBIDLMISC).

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	Idle link list physical key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump 1 =</i>	CKTDATA tuple.
<i>Dump 2 =</i>	CDBCOM tuple.

Table 64 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rSMEST tuple and in the F bits of the signaling information of the peripheral control and timing (PCT) links time slots. Audit found inconsistency between these informations when it compared it. The inconsistency could be in the form that the F bits says no alarm is present, but the rSMEST of the circuit says there is an alarm present or the types of alarm present are not the same in both places. For exact information as to what happened, see the explanation in error address field in this section.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The audit will call TM function to put the circuit on the CGA recovery queue where its alarm information will be correctly applied.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field represents the type of error that occurred. Depending upon its value different explanations are given here. 1 = The F bits says that there is no alarm present on this circuit but one or more of the alarm bits are set in the rSMEST tuple. 2 = The F bits says AIS alarm is present but the rSMEST says either AIS alarm is not set or one of the other alarms, red or yellow, is set. 3 = The F bits says red alarm is present but the rSMEST says either red alarm is not set or one of the other alarms, AIS or yellow, is set. 4 = The F bits says yellow alarm is present but the rSMEST says either yellow alarm is not set or one of the other alarms, AIS or red, is set. 5 = The audit received a value from TM that it is not expecting to receive.
<i>Bad Data =</i>	The local value of circuit type derived from the F bits, <i>ckttype</i> .
<i>Good Data =</i>	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index =</i>	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1 =</i>	The rSMEST tuple.
<i>Dump 2 =</i>	The rCDBCOM tuple.

Table 65 Error Code: NOTBUSY (2)

Error Code:	NOTBUSY (2)
Error Description:	An rSMEST tuple is not transient but its progress flag is set.
Possible Error Effect:	Audit transient timing may operate improperly.
Corrective Action Taken:	The progress flag is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST tuple.

Table 66 Error Code: NOTSPARE

Error Code:	NOTSPARE
Error Description:	An rSMEST tuple has a <i>bas_stat</i> value only allowed for module processor and remote clock equipment.
Possible Error Effect:	Use of this tuple will propagate errors.
Corrective Action Taken:	The tuple is reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of rSMEST. <i>bas_stat</i> .
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST for parent/mate (OPTIONAL).

Table 67 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The CGA event in DNU-S circuits is represented in the form of alarm status in their rSMEST tuple and in the F bits of the signaling information of the PCT links time slots. The audit found an inconsistency when it compared the CGA alarm bits in the rSMEST tuple against the <i>bas_stat</i> and <i>qual_2</i> fields in that tuple. These two fields indicate the base state of the facility represented by the rSMEST tuple. The alarm bits in the rSMEST tuple (<i>ais_alm</i> , <i>red_alm</i> , or <i>ylw_alm</i>) should only be set if the base state indicates either out of service family equipment (OOS/FE) or out of service facility failure (OOS/FAF). In any other case the alarm bits should not be set.
Possible Error Effect:	Possible resource failure due to false alarms.
Corrective Action Taken:	The alarm bits (<i>ais_alm</i> , <i>red_alm</i> , or <i>ylw_alm</i>) are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The local value of circuit type derived from the F bits, <i>ckttype</i> .
<i>Good Data</i> =	The value of F bits received either from PC or from hardware reads. Also see the logical key field to determine how F bits were read.
<i>Logical Index</i> =	The value of local flag. If this value is 0, the value of F bits was read from PC function. If this value is 1, this value was read by sending a task to DNU-S and read hardware to determine CGA alarm status.
<i>Dump 1</i> =	The rSMEST tuple.
<i>Dump 2</i> =	The rCDBCOM tuple.

Table 68 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	No rISLUPIDB tuple exists for this PCPIDBISLU or PCPIDBRISLU circuit.
Possible Error Effect:	This resource is lost to the system.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value used to read rISLUPIDB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 69 Error Code: ODDLOSTTUP (1)

Error Code:	ODDLOSTTUP (1)
Error Description:	A tuple in the static relation RLDSLEQUIP is missing.
Possible Error Effect:	Full use of DSL equipment is impossible.
Corrective Action Taken:	None.

Manual Action Required:	A tuple should be inserted into rIDSLEQUIP as appropriate for the missing PCRDTTMC or PCRDTEOC circuit.
Dump Description:	
<i>Error Address</i> =	Address of rIDSLEQUIP buffer.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA.

Table 70 Error Code: ODDLOSTTUP (2)

Error Code:	ODDLOSTTUP (2)
Error Description:	The CKTDATA audit is attempting to read a nonexistent rIDCKT2GP tuple. The key to the tuple is derived from the rICKTDATA tuple circuit name and circuit index.
Possible Error Effect:	Call processing cannot use the circuit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICKTDATA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rICKTDATA tuple.

Table 71 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The circuit is operational and out-of-service but qual_1 and qual_2 do not have compatible values.
Possible Error Effect:	Circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 72 Error Code: PARENT

Error Code:	PARENT
Error Description:	The rSMEST tuple indicates that this circuit is busy for maintenance but it is not associated with a process nor is its parent circuit busy.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	stat_act field in rSMEST tuple.
<i>Good Data</i> =	Value of DBIDLE (currently 1).
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	Parent's rSMEST.

Table 73 Error Code: PCD

Error Code:	PCD
Error Description:	The audit was not able to read the rVPIDB_F tuple for a corresponding either the STE circuit or the STS-1 circuit for a DNU-S. Refer to <i>Good Data</i> field to

	determine for which type of circuit, STE or STS-1, audit could not read the rVPIDB_F tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report this error until the rVPIDB_F tuple is inserted.
Manual Action Required:	There are three key field for the rVPIDB_F tuple, <i>unittype</i> , <i>unitnumber</i> and <i>pct_link</i> . Use the value of <i>utype</i> and <i>unit</i> from the rCKTDATA dump of this error for the first two keys of the rVPIDB_F tuple, <i>unittype</i> and <i>unitnumber</i> respectively. Use the value of key field for the third key of the rVPIDB_F tuple, <i>pct_link</i> and insert this rVPIDB_F tuple.
Dump Description:	
<i>Error Address</i> =	The <i>unit</i> field of the rCKTDATA tuple.
<i>Bad Data</i> =	Contents of the DS0 circuit name calculated by audit if any.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The <i>pct_link</i> key of the rVPIDB_F tuple as calculated by audit from the given data.
<i>Dump 1</i> =	The rCKTDATA tuple, either for STE or STS-1.
<i>Dump 2</i> =	The rVPIDB_F tuple as returned by database read.

Table 74 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the rCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 75 Error Code: PICBKEY

Error Code:	PICBKEY
Error Description:	The static relation RLISLUPIDB is missing a tuple.
Possible Error Effect:	The tuple is not available for access.
Corrective Action Taken:	None.
Manual Action Required:	The tuple should be inserted into the static relation RLISLUPIDB.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB value.
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 76 Error Code: PID

Error Code:	PID
Error Description:	The PID in the <i>tp_pid</i> field is not OSnullpid nor a live PID and thus is garbage.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The PID is set to OSnullpid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.

<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 77 Error Code: QUAL

Error Code:	QUAL
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but qual_1 indicates the circuit is growth or unequipped and not on the automatic or manual request queues.
Possible Error Effect:	The circuit may be erroneously processed by MRA or lost to call processing.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 78 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The circuit is operational and out-of-service, but qual_2 has an invalid value.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	Qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_2 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	Cross reference entry.
<i>Dump 3 =</i>	Queue entry.
<i>Dump 4 =</i>	riSMEST tuple under audit.
<i>Dump 5 =</i>	Parent of the riSMEST tuple under audit.

Table 79 Error Code: QUAL_EQPG

Error Code:	QUAL_EQPG
Error Description:	The circuit is growth or special growth, but qual_1 is not equal to SMGR or SMSGR.
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qual_1 field is corrected to SMGR or SMSGR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to riSMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	riSMEST tuple.
<i>Dump 2 =</i>	riSMEST tuple for parent/mate (OPTIONAL).

Table 80 Error Code: QUE

Error Code:	QUE
Error Description:	The circuit is operational and out-of-service, qual_2 indicates a diagnostic is running, but the circuit does not have an entry in the queue specified by qual_1

	(auto or man).
Possible Error Effect:	The circuit may be erroneously processed by MRA.
Corrective Action Taken:	The qualifier and bas_stat fields are corrected to null and active, respectively. In the case of the remote clock circuit, qual_2 is set to the value in the register. If the bas_stat is in range, no change is made. Reconfiguration is requested; otherwise, qual_1 is set auto.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	Bad qual_1 field.
<i>Good Data</i> =	Identity of queue in which the circuit was found.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 81 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	A reconfiguration request of a remote clock circuit was not successful.
Possible Error Effect:	Remote clocking may be impaired.
Corrective Action Taken:	All remote clock circuits generating this error, except the parent circuit, are marked out-of-service or active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Return value of the configuration routine.
<i>Good Data</i> =	Action performed on the circuit.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of parent circuit.
<i>Dump 2</i> =	rSMEST of subtending or mate side to a remote clock side.

Table 82 Error Code: RDREG

Error Code:	RDREG
Error Description:	The FCD peripheral control register (PCR), peripheral auxiliary control (PAC) register, or peripheral auxiliary status (PAS) register of the PCTLI can not be read at this time.
Possible Error Effect:	The audit will not be able to verify the status of these registers.
Corrective Action Taken:	None.
Manual Action Required:	The audit continues with SMEST checking.
Dump Description:	
<i>Error Address</i> =	Return value from the read of the FCD_IDLCODE or PAS register.
<i>Bad Data</i> =	Return value from the read of the FCD_HIGHCCS or PAC register.
<i>Good Data</i> =	Return value from the read of the FCDOOS or PCR register.
<i>Logical Index</i> =	Service group number in the upper bits 8 through 16 and TSI link number in bits 0 through 7.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rCKTDATA tuple.

Table 83 Error Code: READ

Error Code:	READ
Error Description:	Remote clock hardware can not be read at this time.
Possible Error Effect:	Can not determine active RCLK.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of mate's rSMEST.bas_stat field.
<i>Good Data</i> =	Value of circuit's rSMEST.bas_stat field.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rSMEST of circuit.
<i>Dump 2</i> =	rSMEST of mate circuit.

Table 84 Error Code: RT2DS1_LOST (1)

Error Code:	RT2DS1_LOST (1)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCU facilities.
Corrective Action Taken:	None.
Manual Action Required:	A tuple in RLRT2DS1 should be inserted as appropriate for the circuit identified in the associated audit dump.
Dump Description:	
<i>Error Address</i> =	Address of RLRT2DS1 buffer.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rCKTDATA.

Table 85 Error Code: RT2DS1_LOST (2)

Error Code:	RT2DS1_LOST (2)
Error Description:	No tuple in RLRT2DS1 exists for this RT circuit.
Possible Error Effect:	Inappropriate use of IDCU facilities.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of RLRT2DS1 buffer.
<i>Bad Data</i> =	Shelf number of RT circuit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rCKTDATA.

Table 86 Error Code: SCNBAD

Error Code:	SCNBAD
Error Description:	An inconsistency has been detected in the value of the IDCU register HWMLSCAN; either it is set when it should not be or vice versa. If the IDCUCOM is active, the bit should be set. If the IDCUCOM is stby, the bit should not be set.
Possible Error Effect:	The scanning state is incorrect for the IDCUCOM identified.
Corrective Action Taken:	The LSCAN bit is reset to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST in error.
<i>Bad Data</i> =	Value of hardware register read.
<i>Good Data</i> =	HWMLSCAN.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST.

Table 87 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	A rCKTDATA static tuple exists in the database associated with a PCT link circuit on a PCT data exchanger unit (PDXU), but the corresponding static rIPDXUINFO tuple is missing from the database.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rIPDXUINFO tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rIPDXUINFO tuple should be inserted into the database with a key (rIPDXU.unit) or PDXU unit number that matches the value of <i>Bad Data</i> in the Dump Description .
Dump Description:	
<i>Error Address</i> =	Address of the corresponding rSMEST tuple.
<i>Bad Data</i> =	PDXU unit number.
<i>Good Data</i> =	TSI link number.
<i>Logical Index</i> =	PCT link circuit name.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	rCKTDATA tuple.

Table 88 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Audit tried to read a tuple of the static relation, RLFAC_PROV, as part of the CGA checks on DNU-S, but failed to read it.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit will continue to take this error until the appropriate rIFAC_PROV tuple is inserted.
Manual Action Required:	Insert a rIFAC_PROV tuple with the key equal to the circuit name dumped in either the logical key or error address fields. The same value is dumped in these fields from different data structures.
Dump Description:	
<i>Error Address =</i>	The <i>name</i> field of the rICKTDATA tuple.
<i>Bad Data =</i>	The key of the rIFAC_PROV tuple, <i>cktname</i> , which audit could not access.
<i>Good Data =</i>	The circuit type from the <i>type</i> field of rICKTDATA tuple.
<i>Logical Index =</i>	Circuit name (rSMEST.unit_id).
<i>Dump 1 =</i>	The rICKTDATA tuple.
<i>Dump 2 =</i>	The rSMEST tuple.

Table 89 Error Code: SMEST

Error Code:	SMEST
Error Description:	The service status [in-service (0), out-of-service (1)] of the rSMEST tuple disagrees with that of other tuples corresponding to the same internal name.
Possible Error Effect:	This may cause the resource with this internal name (or resources, if the name corresponds to a family of equipment) to be stuck until the next transient or stable clear.
Corrective Action Taken:	The rSMEST tuple is forced idle and re-audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST tuple.
<i>Bad Data =</i>	Service status (in/out) of rSMEST tuple.
<i>Good Data =</i>	Expected status.
<i>Logical Index =</i>	Internal name (rSMEST key).
<i>Dump =</i>	rSMEST tuple.

Table 90 Error Code: SSTBAD

Error Code:	SSTBAD
Error Description:	Register contents do not match the contents of the IDCUCOM rSMEST and the LSI(s) rICKTDATA.
Possible Error Effect:	The hardware state of the IDCU is unknown.
Corrective Action Taken:	The circuit is recovered to the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	Value of hardware register read.
<i>Good Data =</i>	Constructed mask indicating correct service state.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST.

Table 91 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rSMEST tuple indicates that this circuit is idle, but it has an entry in one of the queues or the xrf table.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	stat_act field in rSMEST tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	Cross reference entry.
<i>Dump 2 =</i>	Queue entry.
<i>Dump 3 =</i>	rSMEST tuple under audit.

<i>Dump 4 =</i>	Parent of the rSMEST tuple under audit.

Table 92 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A cdbcom of a particular circuit type has bad linkusage semantic data.
Possible Error Effect:	A circuit is probably not usable.
Corrective Action Taken:	The audit will recover the cdbcom and reinitialize the cdbcom.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	cdbcom address.
<i>Bad Data =</i>	linkusage (from cdbcom).
<i>Good Data =</i>	linkusage (expected).
<i>Logical Index =</i>	Logical key of the cdbcom.
<i>Dump =</i>	Circuit type CDBCOM.

Table 93 Error Code: STATUS

Error Code:	STATUS
Error Description:	The rSMEST tuple indicates that this circuit is neither busy nor idle, thus the stat_act field is invalid.
Possible Error Effect:	Circuit is stuck.
Corrective Action Taken:	The rSMEST tuple is marked maintenance idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	stat_act field in rSMEST tuple.
<i>Good Data =</i>	Value of DBIDLE (currently 1).
<i>Logical Index =</i>	Circuit's unit ID.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	rSMEST tuple for parent/mate (OPTIONAL).

Table 94 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rSMEST tuple has been stuck for several minutes in a transient state.
Possible Error Effect:	A process using the rSMEST tuple is stuck with allocated resources and some audit consistency checks are prevented.
Corrective Action Taken:	The rSMEST tuple is recovered to a non-transient state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rSMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rSMEST tuple.

Table 95 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	One or both RCLK circuits has been busy for at least 5 seconds under control of the GLPNRCLK process.
Possible Error Effect:	The circuit may be lost for further MRA requests.
Corrective Action Taken:	The RCLK circuits are further checked to verify RCLK configuration status and are recovered to a safe/consistent state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSMEST in error.
<i>Bad Data =</i>	MRA activity value rSMEST.stat_act.
<i>Good Data =</i>	DBIDLE
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rSMEST for busy RCLK.
<i>Dump 2 =</i>	rSMEST for mate RCLK.

Table 96 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rICDBCOM and its associated rTONECH tuple failed a point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rICDBCOM tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	Tone field in the rICDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rICDBCOM tuple.
<i>Dump 1 =</i>	rICDBCOM tuple.
<i>Dump 2 =</i>	rTONECH tuple.

Table 97 Error Code: TRAFFIC

Error Code:	TRAFFIC
Error Description:	The FCD high hundred call seconds (CCS) register on the PCTLI device is inconsistent with the high CCS traffic configuration (rIPDXUINFO.trf_cnfg) data in the corresponding static rIPDXUINFO tuple.
Possible Error Effect:	Call failures may occur since the PDXU is not configured properly.
Corrective Action Taken:	The FCD high CCS register on the PCTLI device will be made consistent with the high CCS traffic configuration (rIPDXUINFO.trf_cnfg) data in the corresponding rIPDXUINFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	High CCS traffic configuration (rIPDXUINFO.trf_cnfg) from the rIPDXUINFO tuple.
<i>Bad Data =</i>	Contents of the FCD high CCS register.
<i>Good Data =</i>	Correct value of the FCD high CCS register.
<i>Logical Index =</i>	Service group number in the upper bits 8 through 16 and TSI link number in bits 0 through 7.
<i>Dump 1 =</i>	rISMEST tuple.
<i>Dump 2 =</i>	rCKTDATA tuple.

Table 98 Error Code: TYPE

Error Code:	TYPE
Error Description:	The type of circuit that the CKTDATA audit is attempting to audit is either not present in its private circuit mapping array (AUcktreimap[]), or it is marked as invalid in the map.
Possible Error Effect:	The audit will continue to print the error each time it runs but will otherwise ignore the circuit. This is an audit error.
Corrective Action Taken:	None.
Manual Action Required:	Report the error with failure report (FR). Correcting this problem may require manual boot of the SM to restore the RLCKTDATA relation.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Circuit type.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	NA.

Table 99 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The circuit is unequipped but the qualifier qual_1 is not SMUN.
Possible Error Effect:	The circuit is accessible but has no hardware, which will result in interrupts when it is used.
Corrective Action Taken:	Qualifier field is corrected to SMUN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to rISMEST tuple.
<i>Bad Data =</i>	Bad qual_1 field.

<i>Good Data</i> =	Value of SMUN.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	rSMEST.
<i>Dump 2</i> =	rSMEST tuple for parent/mate (OPTIONAL).

Table 100 Error Code: VERTICAL

Error Code:	VERTICAL
Error Description:	The maintenance state of a circuit (rSMEST.bas_stat) is vertically inconsistent with the maintenance state of its "parent" circuit.
Possible Error Effect:	The circuit may not be available for use.
Corrective Action Taken:	The child's rSMEST tuple is made consistent with the parent's.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple for the child circuit.
<i>Bad Data</i> =	Child's rSMEST.bas_stat.
<i>Good Data</i> =	Number of parent circuits.
<i>Logical Index</i> =	Child circuit name.
<i>Dump 1</i> =	The child's rSMEST tuple.
<i>Dump 2</i> =	All the parent circuit rSMESTs (may be several).

Table 101 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An idle CDBCOM is linked to the wrong idle list.
Possible Error Effect:	The circuit may not be allocatable for operational use.
Corrective Action Taken:	The audit will recover the cdbcom to the proper idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	cdbcom address.
<i>Bad Data</i> =	Actual idle list physical key.
<i>Good Data</i> =	Physical key of correct idle list.
<i>Logical Index</i> =	Logical key of the cdbcom.
<i>Dump</i> =	CDBCOM tuple.

Table 102 Error Code: XRFPID

Error Code:	XRFPID
Error Description:	The rSMEST tuple indicates that this circuit has a live PID and the circuit is in xrf or one of the queues, but the PID doesn't match that in the queue.
Possible Error Effect:	The process may be stuck or take actions on an in-service circuit, resulting in a lost call.
Corrective Action Taken:	The process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rSMEST tuple.
<i>Bad Data</i> =	PID of process that claims association.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit's unit ID.
<i>Dump 1</i> =	xrf search results.
<i>Dump 2</i> =	manq, autoq search results.
<i>Dump 3</i> =	rSMEST.
<i>Dump 4</i> =	Dump of parent's rSMEST tuple.

10. CL - CM

CLDB-A

Software Release: 5E10 - 5E13

1. CALL LEG DATA BLOCK (CLDB) AUDIT

The RLCLDB relation is used by any BRCS feature to define the associated call legs (FAR1, FAR2, and so forth) of a BRCS call. The CLDB audit verifies the semantic data and linkages of the RLCLDB relation. It runs segmented and during selective init.

2. DATA STRUCTURES

Figure 1 shows the data structures of the CLDB audit.

3. AUDIT CHECKS

The CLDB audit steps through the RLCLDB relation using the DRINASEK audit driver, verifying linkages with RLPCBLA, RLMDDB and RLSTPDB.

4. ERROR HANDLING

If linkage or semantic errors are detected, the rCLDB is recovered to the idle list, and any associated terminal processes are purged. Related audits are run to recover other data structures.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A tuple of the call leg data block dynamic relation is owned by a process control linkage block area dynamic tuple, but the PCBLA does not have its "cldb_flag" set.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of cldb_flag.
<i>Good Data =</i>	Correct value of cldb_flag.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	A tuple of the call leg data block dynamic relation should own at least one tuple of the model data block relation, but does not.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic rCLDB tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of cldb_tag.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 3 Error Code: BDPID

Error Code:	BDPID
Error Description:	A tuple of the call leg data block dynamic relation references an invalid conference bridge controller process ID (rCLDB.cbc_pid).
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic rCLDB tuple is zeroed and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of rCLDB.cbc_pid.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 4 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	A tuple of the call leg data block dynamic relation is linked to an idle list, but its linkage fields are non-zero.
Possible Error Effect:	Lost resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic rCLDB tuple is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of non-zero linkage field.
<i>Good Data =</i>	Zero.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A tuple of the call leg data block dynamic relation "cldb_tag" is out-of-range.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of cldb_tag.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 6 Error Code: STUCK

Error Code:	STUCK
Error Description:	A tuple of the call leg data block dynamic relation had the attribute rCLDB.rel_stp equal to DBYES since the last time this audit ran.
Possible Error Effect:	FEX will continue to try to release the rSTPDB already released.
Corrective Action Taken:	The attribute rCLDB.rel_stp is restored to DBNO and the STPDB audit is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of rCLDB.rel_stp.
<i>Good Data =</i>	DBNO.
<i>Logical Index =</i>	Key of rCLDB tuple (rCLDB.cldb_id).
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 7 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	A dynamic tuple of the call leg data block relation is linked to an incorrect list.
Possible Error Effect:	Lost resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLDB tuple.
<i>Bad Data</i> =	Key of incorrect list.
<i>Good Data</i> =	Key of correct idle list.
<i>Logical Index</i> =	Key of rCLDB tuple.
<i>Dump 1</i> =	rCLDB tuple.
<i>Dump 2</i> =	Owning rPCBLA tuple.

CLDB-B

Software Release: 5E14 and later

1. CALL LEG DATA BLOCK (CLDB) AUDIT

The RLCLDB relation is used by any BRCS feature to define the associated call legs (FAR1, FAR2, and so forth) of a BRCS call. The CLDB audit verifies the semantic data and linkages of the RLCLDB relation. It runs segmented and during selective init.

2. DATA STRUCTURES

Figure 1 shows the data structures of the CLDB audit.

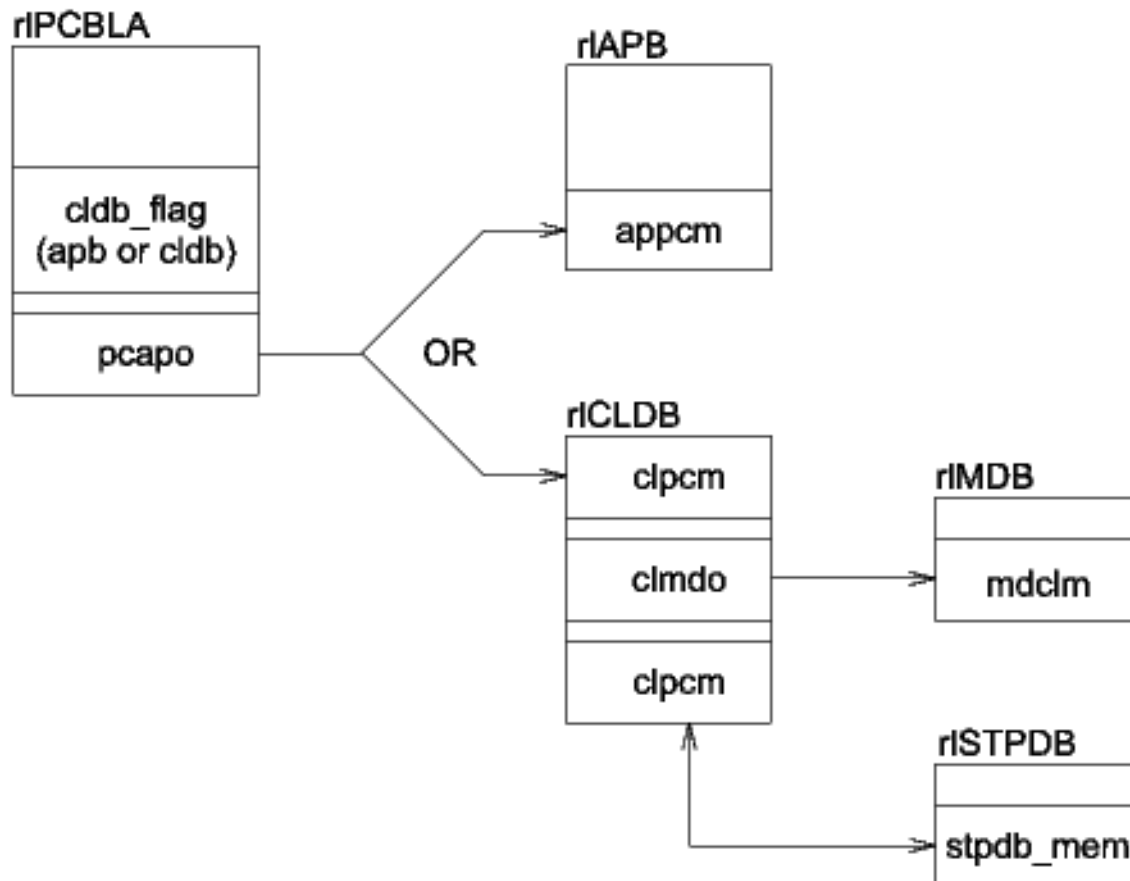


Figure 1 (CLDB-1) Data Structures of CLDB Audit

3. AUDIT CHECKS

The CLDB audit steps through the RLCLDB relation using the DRINASEK audit driver, verifying linkages with RLPCBLA, RLMDDB and RLSTPDB.

4. ERROR HANDLING

If linkage or semantic errors are detected, the rCLDB is recovered to the idle list, and any associated terminal processes are purged. Related audits are run to recover other data structures.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A tuple of the call leg data block dynamic relation is owned by a process control linkage block area dynamic tuple, but the PCBLA does not have its "cldb_flag" set.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of cldb_flag.
<i>Good Data =</i>	Correct value of cldb_flag.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	A tuple of the call leg data block dynamic relation should own at least one tuple of the model data block relation, but does not.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic rCLDB tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of cldb_tag.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 3 Error Code: BDPID

Error Code:	BDPID
Error Description:	A tuple of the call leg data block dynamic relation references an invalid conference bridge controller process ID (rCLDB.cbc_pid).
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic rCLDB tuple is zeroed and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of rCLDB.cbc_pid.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump 1 =</i>	rCLDB tuple.
<i>Dump 2 =</i>	Owning rPCBLA tuple.

Table 4 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	A tuple of the call leg data block dynamic relation is linked to an idle list, but its linkage fields are non-zero.
Possible Error Effect:	Lost resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic rCLDB tuple is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Value of non-zero linkage field.

<i>Good Data</i> =	Zero.
<i>Logical Index</i> =	Key of rCLDB tuple.
<i>Dump 1</i> =	rCLDB tuple.
<i>Dump 2</i> =	Owning rPCBLA tuple.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A tuple of the call leg data block dynamic relation "cldb_tag" is out-of-range.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLDB tuple.
<i>Bad Data</i> =	Value of cldb_tag.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rCLDB tuple.
<i>Dump 1</i> =	rCLDB tuple.
<i>Dump 2</i> =	Owning rPCBLA tuple.

Table 6 Error Code: STUCK

Error Code:	STUCK
Error Description:	A tuple of the call leg data block dynamic relation had the attribute rCLDB.rel_stp equal to DBYES since the last time this audit ran.
Possible Error Effect:	FEX will continue to try to release the rSTPDB already released.
Corrective Action Taken:	The attribute rCLDB.rel_stp is restored to DBNO and the STPDB audit is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLDB tuple.
<i>Bad Data</i> =	Value of rCLDB.rel_stp.
<i>Good Data</i> =	DBNO.
<i>Logical Index</i> =	Key of rCLDB tuple (rCLDB.cldb_id).
<i>Dump 1</i> =	rCLDB tuple.
<i>Dump 2</i> =	Owning rPCBLA tuple.

Table 7 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	A dynamic tuple of the call leg data block relation is linked to an incorrect list.
Possible Error Effect:	Lost resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLDB tuple.
<i>Bad Data</i> =	Key of incorrect list.
<i>Good Data</i> =	Key of correct idle list.
<i>Logical Index</i> =	Key of rCLDB tuple.
<i>Dump 1</i> =	rCLDB tuple.
<i>Dump 2</i> =	Owning rPCBLA tuple.

CLLQDB

Software Release: 5E11 and later

1. CALL QUEUE DATA BLOCK (CLLQDB) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. Each queued call is stored in the RLCLLQDB (call queue data block) relation. Data in this relation is audited by the CLLQDB audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLCLLQDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The audit found an inconsistency between the value of the rICLLQDB ACD position termination request call queue number <code>agreq.call_que</code> and the owning <code>rIHC CLLQ</code> (head cell for call queues) tuple <code>call queue number cqlink.st_ownk</code> .
Possible Error Effect:	Call processing may not function properly for this call queue.
Corrective Action Taken:	The rICLLQDB tuple's <code>agreq.call_que</code> field is set to the value of <code>cqlink.st_ownk</code> field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICLLQDB tuple.
<i>Bad Data =</i>	The value of the <code>agreq.call_que</code> field obtained from the rICLLQDB tuple.
<i>Good Data =</i>	The value of the <code>cqlink.st_ownk</code> field obtained from the rICLLQDB tuple.
<i>Logical Index =</i>	The logical key of the rICLLQDB tuple.
<i>Dump =</i>	The rICLLQDB tuple.

CLNK

Software Release: 5E10 and later

1. COMMUNICATION LINK (CLNK) AUDIT

Processes in the administrative module (AM) communicate with processes in the switching modules (SMs) by sending messages; the messages are transmitted over dedicated communication links. Paths between the AM and the SMs are identified by means of communication link maps; the link maps reside in the AM. There are two types of link maps: logical and communication. The logical links specify over which communication links (CLNKs) the messages travel; there are two logical links for each SM. The communication links from the AM pass through the message switch complex (MSGS) and the office and network timing complex common (ONTCCOM) over time-multiplexed switch links (TMSLNKs) to the switching modules. Software in the Switch Maintenance subsystem controls these units; the data is stored in the peripheral configuration database (PCD). In communication module, model 1 (CM1), each communication link may go through one of four cross-coupled module message processors (MMPs); in communication module, model 2 (CM2), each CLNK may go through one of two MMPs. Cross-coupling the MMPs ensures that logical links to the switching modules are consistently available. Refer to Figure 1. There are up to eight CLNKs from the administrative module to each switching module in CM1, up to four in CM2. The CLNK audit checks the status of the communication link maps in the administrative module. It also verifies a translation table of SM number to the TMSLNK numbers, host SM numbers, and cluster IDs that are used by call processing to set up new calls.

The CLNK audit is part of the switch maintenance audits subgroup. It runs in the SMKP environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The CLNK audit operates on two data structures within the peripheral configuration database: the logical link maps and the communication link maps. The communication link map makes up the link map array (SMlIm); Figure 2 shows one of the eight SMlIm entries per switching module. For CM2, each entry contains the state of the communication link, the number of the module message processor through which the link passes, the number of the BX.25 level 2 protocol controller (XPC) through which the link passes, and the interrupt inhibit state of the link. The logical link map is stored in the logical link map array (SMllIm). The logical link map has two entries per switching module. Each entry on the logical link map indicates the number of the communication link through which the logical link passes, if one exists. Both logical links may indicate the same communication link. The logical link map also indicates when there are no communication links available.

The CLNK audit refers to the static office data relation (RLMODATT) in the office dependent data when checking the TMSLNK numbers stored in the path data relation (RLPHDATA). Both these relations also contain the host switching module (HSM) number and cluster ID if the switching module is a remote switching module (RSM).

The other data structure the CLNK audit refers to is the global system status table (GSS), a table residing in the administrative module that keeps information about the availability of hardware in the SMs.

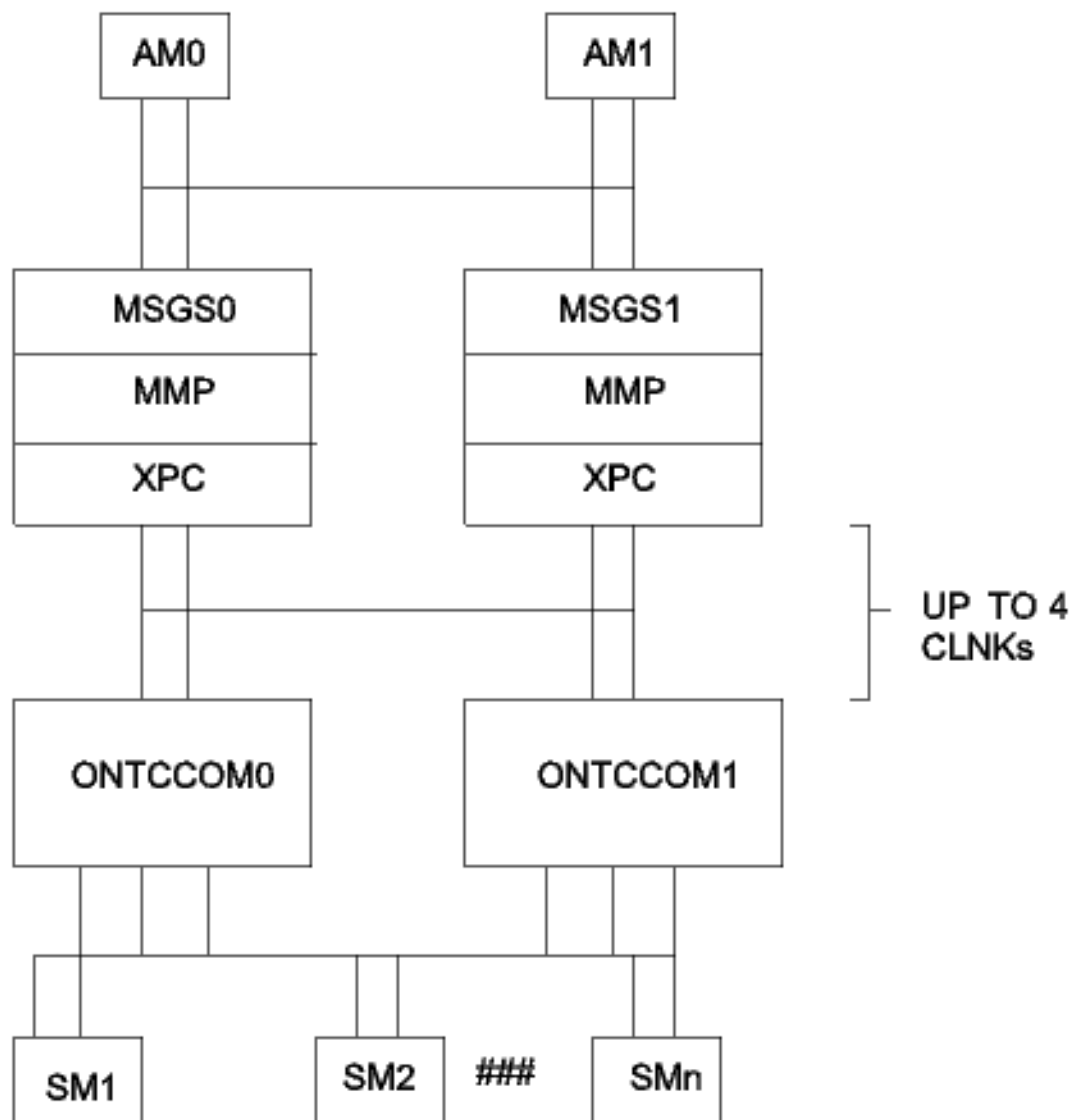


Figure 1 (CLNK-1) Communication Links for CM2

SMlm

CLNK NUMBER	CLNK NUMBER

SMlm

STATE OF CLNK
MMP NUMBER
SDLC NUMBER
INTERRUPT INHIBIT STATE

Figure 2 (CLNK-2) Communication and Logical Link Maps

3. AUDIT CHECKS

The connections among the communication and logical link map arrays and the global data do not use standard linkages; the audit therefore performs no syntactic checks. It makes the following semantic checks:

- (a) Ensures that the MMP and SDLC (XPC for CM2) numbers are correct.
- (b) Verifies the state of the CLNK by comparing it with the state of the hardware. If the hardware unit is out of service, for example, the state of the link must be out of service. If the hardware unit is unequipped, the CLNK is unequipped.
- (c) Ensures that active CLNKs have entries in the logical link map.
- (d) Ensures that the CLNKs to which the logical links point are active.
- (e) Verifies the configurations of different CLNKs (for example, two links through the same MMP to the same SM may not be active at the same time).
- (f) Checks to ensure that the indicator in the global system status table is set correctly (for instance, if the logical link map says that no CLNKs to the SM are available, the global system status table must state that the SM is isolated).
- (g) Compares the TMSLNK numbers, cluster IDs, and HSM numbers stored in rIMODATT and rIPHDATA. The values stored in rIPHDATA must agree with those stored in rIMODATT.

4. ERROR HANDLING

When an audit discovers an error, it corrects the software state by putting the CLNKs in a valid state. It then sends a

message to a process in SMKP to normalize the communication links. This process ensures that both logical links use different CLNKs, if possible. The normalization process also distributes the demands on the use of the CLNKs and therefore of the hardware.

If the audit detects an inconsistency between the data in RLPHDATA and RLMODATT, it updates RLPHDATA to agree with RLMODATT.

The audit produces the following error code(s).

Table 1 Error Code: ACT_TP_CL

Error Code:	ACT_TP_CL
Error Description:	A CLNK in the active state is not pointed to by a logical link.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	CLNK state is set to standby and CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 2 Error Code: CL_MODATT

Error Code:	CL_MODATT
Error Description:	The audit has found a problem with the RLMODATT relation in the office-dependent database (ODD). The tuple for the switching module (SM) whose number is the logical key has been found to be incorrect.
Possible Error Effect:	The SM might be isolated.
Corrective Action Taken:	None.
Manual Action Required:	Check the RLMODATT relation tuple whose key is the same as the logical key in the audit output. This tuple has been found to be incorrect and should be corrected.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DBfrdtup return code.
<i>Good Data =</i>	The module number used as the key.
<i>Logical Index =</i>	The number for the SM which has been found to have an incorrect ODD.
<i>Dump 1 =</i>	Local buffer used in DBfrdtup() call.
<i>Dump 2 =</i>	rIMODATT tuple which was read.
<i>Dump 3 =</i>	Audit's copy of the data read.

Table 3 Error Code: CLID

Error Code:	CLID
Error Description:	The audit has found a problem with the cluster ID number that call processing is using in rIPHDATA.
Possible Error Effect:	Calls may not be getting set up properly.
Corrective Action Taken:	The CLID number that call processing is using is set to the value found in the RLMODATT relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Bad CLID number.
<i>Good Data =</i>	Good CLID number.
<i>Logical Index =</i>	Number of SM with erroneous cluster number.
<i>Dump =</i>	NA.

Table 4 Error Code: DDSCMISOL

Error Code:	DDSCMISOL
Error Description:	The DDSCMISOL data delivery key is set to a non-normal value (DDCMISOL or

	DDICMISOL), but the DDCPLVL2 and DDCPLVL3 keys are set to DDNORMAL, and the link map and logical link map indicate that at least one communication link is available. The DDSCMISOL key should be clear.
Possible Error Effect:	The DDSCMISOL key is set incorrectly. Communication module maintenance functionality is not available.
Corrective Action Taken:	The DDSCMISOL key will be cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The number of the SM being audited.
<i>Dump 1 =</i>	Physical link number.
<i>Dump 2 =</i>	Audits copy of the data read.
<i>Dump 3 =</i>	Hardware and software recovery actions taken.

Table 5 Error Code: EVEN_NCT_LINK

Error Code:	EVEN_NCT_LINK
Error Description:	The audit has found a problem with the even time-multiplexed switch link (TMSLNK) number that call processing is using.
Possible Error Effect:	Calls may not be getting set up properly.
Corrective Action Taken:	The TMSLNK numbers call processing is using are set to the values found in the rMODATT relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Bad TMSLNK number.
<i>Good Data =</i>	Good TMSLNK number.
<i>Logical Index =</i>	The number for the SM whose TMSLNK number has been found in error.
<i>Dump =</i>	NA.

Table 6 Error Code: IIA_CL

Error Code:	IIA_CL
Error Description:	A CLNK whose interrupt inhibit state is not valid for the state the CLNK is in.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	CLNK interrupt inhibit state is set to agree with the state of the CLNK and CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 7 Error Code: INDEX

Error Code:	INDEX
Error Description:	The SM number for which the audit is to check the CLNK data is out of range.
Possible Error Effect:	The audit could assert.
Corrective Action Taken:	AUclindex, the variable which tells the audit which SM to check, is reset. If a bad parameter is pass for directed mode the audit returns with no checks done.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	SM number which has been found to be out of range.
<i>Dump =</i>	NA.

Table 8 Error Code: LEGAL0_CL

Error Code:	LEGAL0_CL

Error Description:	A CLNK is in a state which is not valid for an SM with no active CLNKs.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	CLNK state is set to standby state and CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 9 Error Code: LEGAL1_CL

Error Code:	LEGAL1_CL
Error Description:	A CLNK is in a state which is not valid for an SM with one active CLNK.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	CLNK state is set to standby state and CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 10 Error Code: LEGAL2_CL

Error Code:	LEGAL2_CL
Error Description:	A CLNK is in a state which is not valid for an SM with two active CLNKs. Also a logical link may be pointing to two CLNKs which cannot be active at the same time.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	CLNK state is set to standby and CLNORM is scheduled to refresh the hardware. If a logical link points to two CLNKs which cannot be active at the same time, logical link 1 gets placed on the same CLNK that logical link 0 points to.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 11 Error Code: LEV2_CL

Error Code:	LEV2_CL
Error Description:	Logical link status and global system status level two bit in the AM do not agree.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	The global system status level two bit is set to agree with the logical link status and CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 12 Error Code: MOD_1CONN

Error Code:	MOD_1CONN
Error Description:	The audit has found a problem with the HSM number that call processing is

	using.
Possible Error Effect:	Calls may not be getting set up properly.
Corrective Action Taken:	The HSM number call processing is using is set to the value found in the rMODATT relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Bad HSM number.
<i>Good Data =</i>	Good HSM number.
<i>Logical Index =</i>	Number for the SM whose HSM number has been found in error.
<i>Dump =</i>	NA.

Table 13 Error Code: NSTATE_CL

Error Code:	NSTATE_CL
Error Description:	A CLNK is not in some known state. For example, a CLNK is active on out-of-service hardware.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	CLNK state is set to OOS FE if the CLNK is on OOS hardware, or OOSUNEQ if the CLNK is to an unequipped SM. The logical link status is set to "no link" if it is in error. CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 14 Error Code: ODD_NCT_LINK

Error Code:	ODD_NCT_LINK
Error Description:	The audit has found a problem with the odd TMSLNK number that call processing is using.
Possible Error Effect:	Calls may not be getting set up properly.
Corrective Action Taken:	The TMSLNK numbers call processing is using are set to the values found in the rMODATT relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Bad TMSLNK number.
<i>Good Data =</i>	Good TMSLNK number.
<i>Logical Index =</i>	Number for the SM whose TMSLNK number has been found in error.
<i>Dump =</i>	NA.

Table 15 Error Code: PT_ACT_CL

Error Code:	PT_ACT_CL
Error Description:	A logical link in the AM points to a CLNK which is not active.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	If logical link 0 is in error then logical link 0 gets set to logical link 1's value. If logical link 1 is in error then logical link 1 gets set to logical link 0's value. If both are in error then both are set to "no link." CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	One if audit scheduled CLNORM.
<i>Bad Data =</i>	Link with problem.
<i>Good Data =</i>	Software recovery action.
<i>Logical Index =</i>	Number of SM whose CLNK had problem.
<i>Dump =</i>	Audit's copy of the data read.

Table 16 Error Code: SSDLC

Error Code:	SSDLC

Error Description:	The audit has found a problem with the rISSDLC relation in the office dependent data base (ODD). The tuple whose key is the module message processor (MMP) number and protocol controller (SDLC/XPC) number has been found to be incorrect.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	None.
Manual Action Required:	Check the rISSDLC relation tuple whose key is the MMP and SDLC/XPC numbers in the audit output. This tuple has been found to be incorrect.
Dump Description:	
<i>Error Address</i> =	MMP number.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	SDLC/XPC number.
<i>Logical Index</i> =	SM number which was being checked.
<i>Dump 1</i> =	rISSDLC tuple which was read.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 17 Error Code: STATIC_CL

Error Code:	STATIC_CL
Error Description:	A CLNK has an MMP or SDLC/XPC number which is not correct.
Possible Error Effect:	Communication with the SM may be having problems.
Corrective Action Taken:	The MMP or SDLC/XPC number is set to the value derived from the control time slot. CLNORM is scheduled to refresh the hardware.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	One if audit scheduled CLNORM.
<i>Bad Data</i> =	Link with problem.
<i>Good Data</i> =	Software recovery action.
<i>Logical Index</i> =	Number of SM whose CLNK had problem.
<i>Dump</i> =	Audit's copy of the data read.

Table 18 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	Reinitialize audit global data, return audit complete and wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Number of SM where error occurred.
<i>Dump</i> =	NA.

CLSLs

Software Release: 5E11 and later

1. COMBINED LINK SET (CLSLs) AUDIT

CLSLs audit steps through the relation RLCLS_LS. For each tuple, the audit check the existence of the corresponding rCLS_STAT tuple. If the rCLS_STAT tuple is missing, the audit will construct it and insert the missing rCLS_STAT tuple. If the SM is not the GSM or the static relation is being recent changed or the traffic diversion terminal process is present, the audit will stop.

2. DATA STRUCTURES

For details on the fields contained in the RLCLSLS relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error codes.

Table 1 Error Code: CLS1_LSNUM_MISS

Error Code:	CLS1_LSNUM_MISS
Error Description:	Tuple with the key ls1 in static RCLS_LS is missing from dynamic RLLS_STAT.
Possible Error Effect:	Traffic diversion cannot be performed correctly.
Corrective Action Taken:	Related audit LSDATA runs to correct RLLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rCLS_LS tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	ls1 from rCLS_LS.
<i>Logical Index =</i>	cls (key) from rCLS_LS tuple.
<i>Dump =</i>	NA.

Table 2 Error Code: CLS2_LSNUM_MISS

Error Code:	CLS2_LSNUM_MISS
Error Description:	Tuple with the key ls2 in static RCLS_LS is missing from dynamic RLLS_STAT.
Possible Error Effect:	Traffic diversion cannot be performed.
Corrective Action Taken:	Related audit LSDATA runs to correct RLLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rCLS_LS tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	ls2 from rCLS_LS.
<i>Logical Index =</i>	cls (key) from rCLS_LS tuple.
<i>Dump =</i>	NA.

Table 3 Error Code: DUP

Error Code:	DUP
Error Description:	The audit tried to insert a missing tuple into RLCLS_STAT, but there is a tuple

	with the same key.
Possible Error Effect:	Traffic diversion cannot be performed.
Corrective Action Taken:	The audit updates the duplicate rCLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_LS tuple.
<i>Bad Data</i> =	Key of duplicate rCLS_STAT tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The combined link set number from static RLCLS_LS was used to read dynamic RLCLS_STAT but found no match; the tuple is missing from the dynamic.
Possible Error Effect:	Traffic diversion cannot be performed correctly.
Corrective Action Taken:	The audit constructs the missing tuple and inserts it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	cls (key) from rCLS_LS tuple.
<i>Dump</i> =	NA.

CLSSTAT

Software Release: 5E11 and later

1. COMBINED LINK SET STATUS (CLSSTAT) AUDIT

CLSSTAT audit steps through the relation RLCLS_STAT. For each tuple, the audit makes sure that the corresponding static relation RLCLS_LS exist. The extra rCLS_STAT tuple will be deleted. For each rCLS_STAT tuple in the system, the audit checks the semantic data such as signaling link selection (SLS), lnum1 (first component of combined link set), lnum2 (second component of combined link set) against the RLLS_GRP (link set group) and RLLS_STAT (link set status) relations.

2. DATA STRUCTURES

For details on the fields contained in the RLCLSSTAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD1_AVAIL

Error Code:	BAD1_AVAIL
Error Description:	The error is due to one of the following: (1) The combined link set is marked available in rCLS_STAT and rLNK_STAT. The avail for one of the combined link set's sls in the switching module processor (SMP) copy of the protocol handler (PH) dynamic data array does not equal the avail in rLNK_STAT for that combined link set. (2) The combined link set is marked available in rCLS_STAT and one of its links is marked unavailable in rLNK_STAT. The avail for one of the combined link set's sls in the SMP copy of the PH dynamic data array does not equal the avail in rLNK_STAT for that combined link set.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit corrects the value in the array and asks signaling transfer and route management (STRM) to perform route correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	SMP copy of the PH dynamic data array.
<i>Bad Data =</i>	avail in SMP copy.
<i>Good Data =</i>	clnum from rCLS_STAT + 80 (first index into array).
<i>Logical Index =</i>	Position of sls bit set to 1 (second index into array).

<i>Dump</i> =	01 indicates that the error is in curr1_sls[].

Table 2 Error Code: BAD2_AVAIL

Error Code:	BAD2_AVAIL
Error Description:	The error is due to one of the following: (1) The combined link set is marked available in rCLS_STAT and rLNK_STAT. The avail for one of the combined link set's sls in the SMP copy of the PH dynamic data array does not equal the avail in rLNK_STAT for that combined link set. (2) The combined link set is marked available in rCLS_STAT and unavailable in rLNK_STAT. The avail for one of the combined link set's sls in the SMP copy copy of the PH dynamic data array does not equal the avail in rLNK_STAT for that combined link set.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit corrects the value in the array and asks STRM to perform route correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of the PH dynamic data array.
<i>Bad Data</i> =	Avail in SMP copy.
<i>Good Data</i> =	clsnun from rCLS_STAT + 80 (first_index into array).
<i>Logical Index</i> =	Position of sls bit set to 1 (second_index into array).
<i>Dump</i> =	01 indicates that the error is in curr2_sls[].

Table 3 Error Code: BAD_CONGEST

Error Code:	BAD_CONGEST
Error Description:	The congestion level of the combined link set does not equal the higher congestion level of its two components.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The congestion level of the combined link is set to the higher congestion level of its two components.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	Value of congestion in rCLS_STAT tuple.
<i>Good Data</i> =	The higher congestion level of the two component link sets.
<i>Logical Index</i> =	clsnun (key) of rCLS_STAT tuple.
<i>Dump</i> =	NA.

Table 4 Error Code: CLS1_LSNUM_MISS

Error Code:	CLS1_LSNUM_MISS
Error Description:	lsnum1 in dynamic rCLS_STAT for a combined link set exists in static rLS_GRP but is missing from dynamic rLS_STAT.
Possible Error Effect:	Traffic diversion fails.
Corrective Action Taken:	The related audit LSDATA runs to correct rLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	lsnum1 from rCLS_STAT.
<i>Logical Index</i> =	clsnun (key) from rCLS_STAT tuple.
<i>Dump</i> =	NA.

Table 5 Error Code: CLS2_LSNUM_MISS

Error Code:	CLS2_LSNUM_MISS
Error Description:	lsnum2 in dynamic rCLS_STAT for a combined link set exists in static

	rCLS_GRP but is missing from dynamic rCLS_STAT.
Possible Error Effect:	Traffic diversion fails.
Corrective Action Taken:	The related audit LSDATA runs to correct rCLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	lsnum2 from rCLS_STAT.
<i>Logical Index</i> =	clsnum (key) from rCLS_STAT tuple.
<i>Dump</i> =	NA.

Table 6 Error Code: CLS_ACTNUM_BAD

Error Code:	CLS_ACTNUM_BAD
Error Description:	The number of active links in the combined link set does not equal the sum of the active links for the component link sets.
Possible Error Effect:	SLS field diversion is not correct.
Corrective Action Taken:	The audit corrects the number of active links for the combined link set in dynamic rCLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	Active links number from rCLS_STAT.
<i>Good Data</i> =	Sum of active links in component link sets.
<i>Logical Index</i> =	clsnum (key) from rCLS_STAT tuple.
<i>Dump</i> =	NA.

Table 7 Error Code: CLS_STAT_AVAIL

Error Code:	CLS_STAT_AVAIL
Error Description:	The error is due to one of the following: (1) One or both of the component link sets of the combined link set is marked available, but the combined link set is not marked available. (2) Both of the component link sets of the combined link set are marked unavailable, but the combined link set is not marked unavailable. NOTE: <i>Bad Data</i> and <i>Good Data</i> are used to report the components of the combined link set.
Possible Error Effect:	Traffic diversion is not correct in either case.
Corrective Action Taken:	The audit corrects the availability of combined link set in rCLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	lsnum1 from rCLS_STAT.
<i>Good Data</i> =	lsnum2 from rCLS_STAT.
<i>Logical Index</i> =	clsnum (key) from rCLS_STAT tuple.
<i>Dump</i> =	Corresponds to the Error Description as follows: (1) Indicates the error. (2) Indicates the error.

Table 8 Error Code: CONGESTION

Error Code:	CONGESTION
Error Description:	The congestion level for the combined link set in the array CCls_info is not equal to the congestion level in the tuple for the combined link set.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The congestion level in the array for the combined link set is set equal to the congestion level in the tuple for the combined link set.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLS_STAT tuple.
<i>Bad Data</i> =	Congestion level at CCls_info[rCLS_STAT.clsnum + CC7MXLS].
<i>Good Data</i> =	rCLS_stat.cg.
<i>Logical Index</i> =	rCLS_STAT.clsnum.
<i>Dump</i> =	NA.

Table 9 Error Code: LSNUM_BAD

Error Code:	LSNUM_BAD
Error Description:	The link set number1/link set number2 in dynamic rCLS_STAT does not match link set number1/link set number2 in static rCLS_LS for a combined link set number.
Possible Error Effect:	Traffic is diverted to an incorrect link set.
Corrective Action Taken:	The audit corrects the dynamic data to match static.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	lsnum1 or lsnum2 from rCLS_STAT tuple.
<i>Good Data</i> =	ls1 or ls2 from rCLS_LS tuple.
<i>Logical Index</i> =	clsnum (key) from rCLS_STAT tuple.
<i>Dump</i> =	Corresponds to the Error Description as follows: (1) Indicates lsnum1 caused the error. (2) Indicates lsnum2 caused the error.

Table 10 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The error is due to one of the following: 1 and 2 - There are no links in a combined link set loaded with traffic, but the normal SLS field codes for the combined link set are not set to zero. 3 and 4 - The combined link set is marked unavailable, but its current SLS codes are not set to zero.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit corrects the SLS arrays.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	Value of non-null component in array.
<i>Good Data</i> =	Corresponds to the Error Description as follows: 1 - Component was in norm1_sls[].
<i>Logical Index</i> =	clsnum (key) from rCLS_STAT tuple.
<i>Dump</i> =	rCLS_STAT tuple.

Table 11 Error Code: NORM1_SLS_BAD

Error Code:	NORM1_SLS_BAD
Error Description:	A normal sls for the first component of a combined link is not correct as determined from static rLNKSLS.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit corrects the value in rCLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	Value of sls in norm1_sls[].

<i>Good Data</i> =	Value of sls from static data.
<i>Logical Index</i> =	clsnum (key) of rCLS_STAT tuple.
<i>Dump</i> =	Index into norm1_sls[].

Table 12 Error Code: NORM2_SLS_BAD

Error Code:	NORM2_SLS_BAD
Error Description:	A normal sls for the second component of a combined link is not correct as determined from static rLNKSLS.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit corrects the value in rCLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	Value of sls in norm2_sls[].
<i>Good Data</i> =	Value of sls from static data.
<i>Logical Index</i> =	clsnum (key) of rCLS_STAT tuple.
<i>Dump</i> =	Index into norm2_sls[].

Table 13 Error Code: NO_LNK_STAT

Error Code:	NO_LNK_STAT
Error Description:	The link in lnkset for lsn1/lsn2 does not exist in the dynamic relation RLLNK_STAT.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The related audits LNKDATA, LNKSLs, and LSDATA run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	Link at linkindex in lnkset.
<i>Good Data</i> =	linkindex.
<i>Logical Index</i> =	clsnum (key) of rCLS_STAT tuple.
<i>Dump</i> =	One of the following: 01 - Indicates that the error was caused by first component. 02 - Indicates that the error was caused by second component.

Table 14 Error Code: RDATA_BAD

Error Code:	RDATA_BAD
Error Description:	The error is due to one of the following: (1) The audit calls a common channel signaling 7 (CCS7) function to retrieve the entry in CCsmstph_ht[][] for this component, but the function fails to retrieve it. (2) The audit calls a CCS7 function to retrieve the entry in CCsmstph_ht[][] for this component, but the function fails to retrieve it. In both cases, the SLS codes are the ANSI [®] variation.
Possible Error Effect:	The entry in CCsmstph_ht[][] cannot be verified.
Corrective Action Taken:	The audit tries again during its next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	(1) or (2) indicates error description.
<i>Bad Data</i> =	Return from function call.
<i>Good Data</i> =	rCLS_STAT.clsnum + CC7MXLS.
<i>Logical Index</i> =	Index.
<i>Dump 1</i> =	Contents of rLNK_STAT tuple.
<i>Dump 2</i> =	Contents of rCLS_STAT tuple.

Table 15 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	lsum1/lsum2 in dynamic rCLS_STAT have been verified as correct against ls1/ls2 in static rCLS_LS. However, lsum1/lsum2 does not exist in static rCLS_GRP.
Possible Error Effect:	Traffic is routed on the wrong link set.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the link set numbers in both of these static relations.
Dump Description:	
<i>Error Address</i> =	rCLS_STAT tuple.
<i>Bad Data</i> =	lsum1 or lsum2 from rCLS_STAT tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	clsnum (key) from rCLS_STAT tuple.
<i>Dump</i> =	One of the following: 01 - Indicates that lsum1 caused the error. 02 - Indicates that lsum2 caused the error.

Table 16 Error Code: SELECT_BIT_BAD

Error Code:	SELECT_BIT_BAD
Error Description:	The audit detects that the link set selection bit (rCLS_STAT.ls_sel) for a combined link set (rCLS_STAT.clsnum) in the dynamic relation does not equal the link set selection bit (rCLS_LS.ls_sel) for the same combined link set (rCLS_LS.cls) in the static relation.
Possible Error Effect:	SLS distribution among combined link sets will be incorrect.
Corrective Action Taken:	The audit sets rCLS_STAT.ls_sel equal to rCLS_LS.ls_sel.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLS_STAT tuple.
<i>Bad Data</i> =	rCLS_STAT.ls_sel.
<i>Good Data</i> =	rCLS_LS.ls_sel.
<i>Logical Index</i> =	rCLS_STAT.clsnum.
<i>Dump</i> =	NA.

Table 17 Error Code: SLS_CHNUM_BAD

Error Code:	SLS_CHNUM_BAD
Error Description:	The error is due to one of the following: (1) A bit for an SLS code for a component (rCLS_STAT.lsum1) of a combined link set (rCLS_STAT.clsnums + CC7MXLS) is set in rCLS_STAT.curr1_sls[]. The CCsmstph_htf[][].sls_chnum for this component does not equal rLNK_STAT.chnum for the same component. (2) A bit for an SLS code for a component (rCLS_STAT.lsum2) of a combined link set (rCLS_STAT.clsnums + CC7MXLS) is set in rCLS_STAT.curr2_sls[]. The CCsmstph_htf[][].sls_chnum for this component does not equal rLNK_STAT.chnum for the same component.
Possible Error Effect:	It is impossible to identify the correct PH channel corresponding to this link, so calls cannot be routed.
Corrective Action Taken:	The audit requests common channel signaling (CCS) process to correct the entry in CCsmstph_htf[][].
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	(1) or (2) indicates error description.
<i>Bad Data</i> =	CCsmstph_ht[rCLS_STAT.clsnum + CC7MXLS][index].sls_chnum.
<i>Good Data</i> =	rCLS_STAT.clsnum + CC7MXLS.
<i>Logical Index</i> =	Index.
<i>Dump</i> =	Contents of rLNK_STAT tuple.
<i>Dump</i> =	Contents of rCLS_STAT tuple.
<i>Dump</i> =	Gld_sh_op. A value of CCANSILS indicates error is in ANSI [®] variation of SLSs. A value other than CCANSILS indicates error is not in ANSI [®] variation.

Table 18 Error Code: SLS_CLINK

Error Code:	SLS_CLINK
Error Description:	The error is one of the following: (1) One or more SLS codes is/are missing from rCLS_STAT.curr1_sls[] and/or rCLS_STAT.curr2_sls[] for the components of that combined link set. (2) SLSs should not be duplicated for a combined link set (rCLS_STAT.clsnum). (3) The normal SLS code (rCLS_STAT.norm1_sls[]) for links in the first component (rCLS_STAT.lsn1) of a combined link set (rCLS_STAT.clsnum) must be a subset of the current SLS code (rCLS_STAT.curr1_sls[]) for links in the first component of that combined link set, but is not. (4) The normal SLS code (rCLS_STAT.norm2_sls[]) for links in the second component (rCLS_STAT.lsn2) of a combined link set (rCLS_STAT.clsnum) must be a subset of the current SLS code (rCLS_STAT.curr2_sls[]) for links in the second component of that combined link set, but is not.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit sends the STRM system process a message to do SLS correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCLS_STAT tuple.
<i>Bad Data</i> =	Gld_sh_op. A value of CCANSILS indicates error is in ANSI [®] variation of SLSs. A value other than CCANSILS indicates error is not in ANSI [®] variation.
<i>Good Data</i> =	(1), (2), (3), or (4) indicates error description.
<i>Logical Index</i> =	rCLS_STAT.clsnum.
<i>Dump</i> =	Contents of rCLS_STAT tuple.

Table 19 Error Code: SLS_STPH_BAD

Error Code:	SLS_STPH_BAD
Error Description:	The error is due to one of the following: (1) A bit for an SLS code for a component (rCLS_STAT.lsn1) of a combined link set (rCLS_STAT.clsnums + CC7MXLS) is set in rCLS_STAT.curr1_sls[]. The CCsmstph_ht[][].stph for this component does not equal rLNK_STAT.stph for the same component. (2) A bit for an SLS code for a component (rCLS_STAT.lsn2) of a combined link set (rCLS_STAT.clsnums + CC7MXLS) is set in rCLS_STAT.curr2_sls[]. The CCsmstph_ht[][].stph for this component does not equal rLNK_STAT.stph for the same

	component.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit requests CCS process to correct the entry in CCsmstph_hf[][].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	(1) or (2) indicates error description.
<i>Bad Data</i> =	CCsmstph_hf[rCLS_STAT.clsnum + CC7MXLS][index].sls_stph.
<i>Good Data</i> =	rCLS_STAT.clsnum + CC7MXLS.
<i>Logical Index</i> =	Index.
<i>Dump</i> =	Contents of rLNK_STAT tuple.
<i>Dump</i> =	Contents of rCLS_STAT tuple.
<i>Dump</i> =	Gld_sh_op. A value of CCANSILS indicates error is in <i>ANSI</i> [®] variation of SLSSs.
	A value other than CCANSILS indicates error is not in <i>ANSI</i> [®] variation.

Table 20 Error Code: TRANS_FAILED

Error Code:	TRANS_FAILED
Error Description:	The corrected segment of the SMP copy of the PH dynamic data array was not successfully transmitted to all PHs.
Possible Error Effect:	Signaling terminal protocol handler (STPH) data is incorrect.
Corrective Action Taken:	When audits LSTAB, DPCTAB, and TABDPC complete, audit LSTAB asks for a checksum to be done by the PHs. If the checksums do not match, the PHs ask for a refresh of their dynamic data array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of PH dynamic data array.
<i>Bad Data</i> =	Return value of transmission.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

CMIM

Software Release: 5E10 and later

1. COMMUNICATION PACKAGE IN THE SWITCHING MODULE (CMIM) AUDIT

The administrative module and switching modules communicate by sending messages to each other. The messages, transmitted over bidirectional communication paths, are controlled by the communication package kernel process (CMKP). CMKP establishes paths between the synchronous data link controllers (SDLCs) in the switching module and the message switches (MSGs), and from the message switches to the control units (CUs) in the AM. The CMIM audit also checks the consistency of link groups used for RSM derived data link (DDL) communication; this includes message and packet buffers.

The communication package in the switching module (CM-SM) uses message buffers to ensure that the messages sent between processors have the correct link protocols. A fixed number of message buffers exists in the AM and the SMs. The communication package in the switching module audit (CMIM) checks the message buffers in the SMs to ensure that no message buffers are lost. Lost message buffers may result in service degradations.

The CMIM audit runs only in the SM.

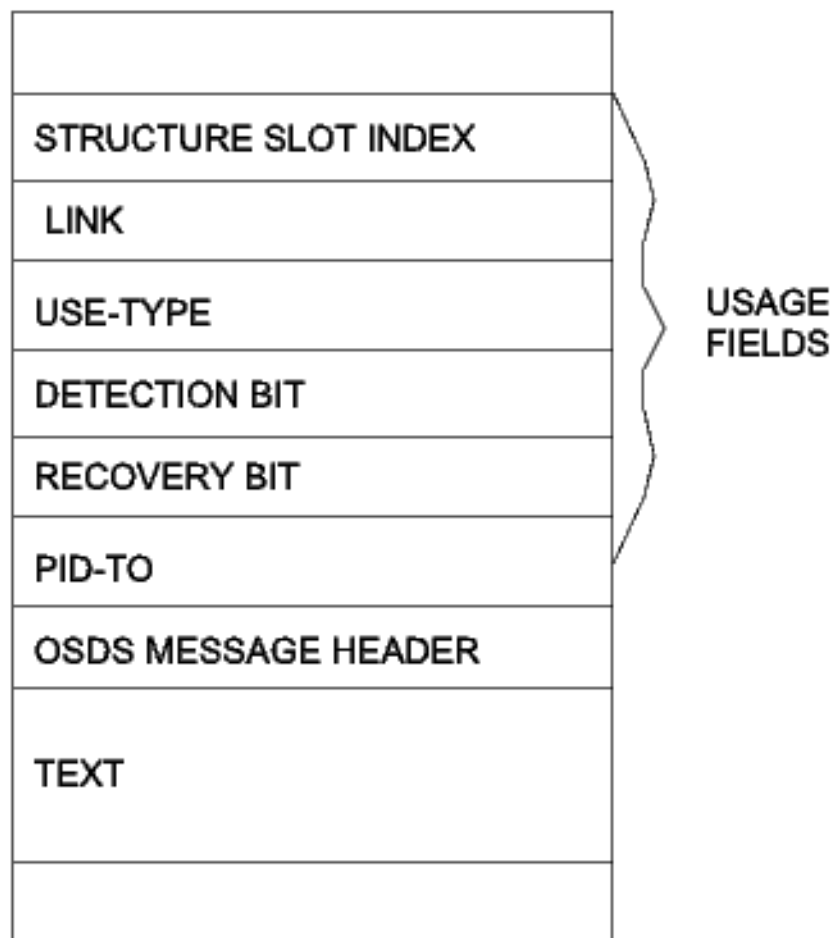
2. DATA STRUCTURES

The CMIM audit verifies the link protocol message buffers (CMbufnode) and the following communications package data structures in the SMs.

Idle buffer stack	= The stack to which idle message buffers are linked.
Input queue	= The queue into which input messages from users are loaded.
Link transmit queues	= The queues into which unacknowledged messages and messages waiting to be sent are stored. Unacknowledged messages are messages sent from the SM to the AM; they are stored in the link transmit queue until the AM acknowledges that it has received them.
Output queues	= An overflow queue that stores output messages when the link transmit queue is full.
Receive buffer	= The buffer into which the next message received is copied; there is one receive buffer for each NCT link.

These data structures are kept in arrays in global memory. They do not use one-to-one or owner-member linkages. Rather, the link protocol message buffers are parts of doubly linked lists with the stacks and queues. The communication package in the switching modules maintains pointers to the link protocol data for each NCT link. The pointers are write-protected; they are located in the CMLk_dptrs array.

For the RSM DDL communication checking, the RLCLIDAT and RLMATATT relations are read to find the number of inter cluster RSM links. Refer to Figure 2 .

CMbufnode**Figure 1 (CMIM-1) Message Buffer**

RLCLIDAT

Remote Site #

Remote Modules in cluster ARRAY

RLMATATT

mate RSM

num_icls
.
.
.

Figure 2 (CMIM-2) RSM DDL Communication Checking for Generics**3. AUDIT CHECKS**

The audit routine is the same for the idle buffer stack, the receive buffer, the input and output queues, and the link transmit queues.

- Audit control checks the manner of the audit to determine whether it is starting or resuming. If the audit is starting, it initializes a cursor to index the first message buffer. If it is resuming, it verifies that the cursor indexes a valid message buffer. If the cursor is not indexing a valid message buffer, there is an error and the audit process is purged.
- The audit examines a message buffer and verifies the consistency of the double linkage between it and the CM-SM data structures. It checks:
 - The validity of the usage of the data structure
 - The soundness of the double linkage into the data structure
 - The validity of the access to the CM.

The audit does not take any action unless it detects the same inconsistency more than twice; this allows transient states resulting from the normal operation of interrupts and resets to occur. If the audit detects the same inconsistency three times, it reports an error and takes corrective action. The audit keeps track of whether an error has been detected by means of the detection and recovery bits in the usage fields of the message buffers. Refer to Figure 1. These bits are cleared during system initializations and set each time the audit detects an inconsistency.

The audit uses the detection and recovery bits as follows. After an initialization and during normal system operation, both bits are cleared. When the audit detects an inconsistency, it sets the detection bit; the bit may be set because of a linkage error or because CM-SM is accessing the buffer at the interrupt level. If the buffer is a receive buffer or an idle buffer, or if it is linked into the input, output, or link transmit queue, CM-SM clears the detection bit before the audit is reinvoked. If the buffer is lost, CM-SM is not able to clear the detection bit.

If the audit discovers a set detection bit, it verifies the consistency of the buffer linkages. During its second pass, the audit is not able to distinguish between lost buffers and inconsistencies resulting from a simultaneous removal of the buffer by CM-SM from the top of a stack. The audit therefore sets the recovery bit. If possible, CM-SM clears the detection bit. When the message buffer is lost, CM-SM is not able to clear the detection bit so both bits are set.

If both the detection and the recovery bits are set when the audit encounters the message buffer for the third time, the audit considers the buffer lost and recovers it. If only the recovery bit is set, the audit assumes the buffer was the top buffer in a stack and the audit initializes the recovery bit; this condition is normal and does not invoke the error handling routines.

The audit steps through each entry in CMrcldata.hdw[], a global data structure which holds information about each RSM Cluster Link in the office. The unit type, bit access, maximum ddl size and circuit name are verified against rICKTDATA for each entry.

4. ERROR HANDLING

If the audit detects a lost message buffer, it recovers the buffer by initializing it and links the message buffer to the idle buffer stack.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The bit_access field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .bit_access.
<i>Good Data</i> =	Correct access type based on the unit type 8 for DLTU, 16 for DLTU2.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 2 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	The block index specifying the head (head) CMmsgbuf.bq[] entry on the queue is out of range, or the block index specifying the tail (tail) CMmsgbuf.bq[] entry on the queue is out of range.
Possible Error Effect:	Possible exhaustion of CMmsgbuf.bq[] message queue write buffer resources.
Corrective Action Taken:	The head and/or tail linkage is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the CMecbusy[] busy queue array block entry.
<i>Bad Data</i> =	The head or tail linkage (head or tail) that was found corrupt.
<i>Good Data</i> =	The head or tail linkage value expected.

<i>Logical Index</i> =	The index of the CMmsgbuf.bq[] array entry.
<i>Dump</i> =	The contents of the CMecbusy[] busy queue array block entry.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The unittype field is incorrect as compared against rCKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .unittype.
<i>Good Data</i> =	Unit type according to rCKTDATA.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 4 Error Code: BITMAP

Error Code:	BITMAP
Error Description:	A CMmsgbuf.bq[] entry was found lost to the system. This indicates a condition where the block is not on either the free or busy queue.
Possible Error Effect:	Possible exhaustion of CMmsgbuf.bq[] resources.
Corrective Action Taken:	The CMmsgbuf.bq[] entry is recovered to the free list (the entry is zeroed out and qnext and qprev fields are adjusted as needed to link the entry into the free list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the CMmsgbuf.bq[] entry found lost to the system.
<i>Bad Data</i> =	The block index marked in the entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The block index marked in the entry.
<i>Dump</i> =	The contents of the CMmsgbuf.bq[] entry found lost to the system.

Table 5 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The global data identifying remote cluster linkage information has been corrupted. One entry should exist for each RCL; however, an entry exists in CMrcldata which does not exist in rCKTDATA.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The ICC system process is purged in order to reinitialize CMrcldata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of start of CMrcldata.hdw.
<i>Bad Data</i> =	rclname in CMrcldata.hdw.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Circuit name of the RCL.
<i>Dump</i> =	Entry in CMrcldata.

Table 6 Error Code: CLIDAT

Error Code:	CLIDAT
Error Description:	Failure in reading the RLCLIDAT relation.
Possible Error Effect:	The link groups will not be audited, which could cause lost messages or resources.
Corrective Action Taken:	None.
Manual Action Required:	RLCLIDAT has only one tuple, with the key being equal to the value stored in the Gclid global variable. If this error code is seen, there is a error in either the Gclid global variable or the RLCLIDAT relation.
Dump Description:	
<i>Error Address</i> =	Address of the local rCLIDAT tuple.

<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	Gclid global variable.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rICLIDAT tuple.

Table 7 Error Code: CMIMBUFFER

Error Code:	CMIMBUFFER
Error Description:	A linkage inconsistency has been detected in the message buffer indexed by AUcursor during two previous passes of the CMIM Audit. CM-SM has not accessed the message buffer during this interval. The CMIM Audit has verified the linkage inconsistency in a critical region in which all interrupts and resets except non-maskable interrupts are blocked.
Possible Error Effect:	The message buffer is lost.
Corrective Action Taken:	The message buffer has been linked into idle buffer stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of message buffer indexed by AUcursor.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	Message buffer usage.

Table 8 Error Code: CMIMIDLSTK

Error Code:	CMIMIDLSTK
Error Description:	A linkage inconsistency has been detected in the message buffer indexed by AUcursor. CM-SM has not accessed the message buffer during this interval. The CMIM audit has verified the linkage inconsistency in a critical region in which all interrupts and resets except non-maskable interrupts are blocked. In addition, the linkage to the idle message buffer stack is broken.
Possible Error Effect:	The CMIM Audit cannot link the lost message buffer into the idle message buffer stack. This could cause level 3 and/or CTS error(s) to occur.
Corrective Action Taken:	CM-SM link protocol and link protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of message buffer indexed by AUcursor.
<i>Bad Data</i> =	Idle message buffer stack top index.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	Message buffer usage.

Table 9 Error Code: CMMSG_TYPE

Error Code:	CMMSG_TYPE
Error Description:	An RSM derived data link (DDL) message buffer was found with an unknown usage type.
Possible Error Effect:	This could cause a message lost buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL message buffer usage field.
<i>Bad Data</i> =	Illegal usage type value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DDL message buffer number.
<i>Dump</i> =	DDL message buffer usage.

Table 10 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	Failure in reading the RLMATATT relation.
Possible Error Effect:	Link group will not be audited, which could result in lost messages.
Corrective Action Taken:	None.
Manual Action Required:	For a MMRS office, make sure that this relation as well as rIMODATT are

	populated correctly.
Dump Description:	
<i>Error Address</i> =	Address of the local rIMATATT tuple.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	SM mate side.
<i>Dump</i> =	rIMATATT tuple.

Table 11 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The global data identifying remote cluster linkage information has been corrupted. The max_ddl_size field is incorrect as compared against rICKTDATA for the RCL.
Possible Error Effect:	Inter cluster communication may be impaired.
Corrective Action Taken:	The global data is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of incorrect entry in CMrcldata.
<i>Bad Data</i> =	CMrcldata.hdw[] .max_ddl_size.
<i>Good Data</i> =	18 or 30.
<i>Logical Index</i> =	RCL circuit name.
<i>Dump</i> =	CMrcldata entry.

Table 12 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The block index specifying the next (qnext) CMmsgbuf.bq[] entry on the queue is out of range, or the block index specifying the previous (qprev) CMmsgbuf.bq[] entry on the queue is out of range. These values must be in range (greater than or equal to 0 but less than CMWPOOLSIZE). In addition, the previous index must equal the index of the previous block audited. NOTE: If the next linkage was invalid, both of the dumps listed below are printed. Otherwise, just the second dump (contents of the tail block under audit) is printed.
Possible Error Effect:	Possible exhaustion of CMmsgbuf.bq[] message queue write buffer resources.
Corrective Action Taken:	The block containing the invalid index is correctly linked back to the CMmsgbuf.bq[] queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the block found containing the invalid next or previous linkage(s).
<i>Bad Data</i> =	The contents of the linkage to then next block in the queue qnext .
<i>Good Data</i> =	Valid values are: 0 = The next linkage was invalid. 1 = The prev linkage was invalid.
<i>Logical Index</i> =	The block index of the block on the head of the queue.
<i>Dump 1</i> =	The contents of the head block under audit.
<i>Dump 2</i> =	The contents of the tail block under audit.

Table 13 Error Code: IDLECHAN

Error Code:	IDLECHAN
Error Description:	A DDL link group has a logical channel that is idle, but has message to send.
Possible Error Effect:	This could cause lost messages.
Corrective Action Taken:	The CM DDL logical channel has been reset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL link group.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	DDL link group.

<i>Logical Index</i> =	DDL logical channel.
<i>Dump</i> =	DDL logical channel.

Table 14 Error Code: LG_EXIST

Error Code:	LG_EXIST
Error Description:	An RSM DDL link group was found in error.
Possible Error Effect:	This could cause messages to be lost.
Corrective Action Taken:	If the link group should not exist, the idlechan bit 0 is set to a 1. Otherwise the link group is restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL link group.
<i>Bad Data</i> =	Bit 0 of the idlechans bit map.
<i>Good Data</i> =	Corrected bit 0 of the idlechans bit map.
<i>Logical Index</i> =	CM link group number.
<i>Dump</i> =	NA.

Table 15 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The logical channel in the x3header field for this packet buffer is out of range.
Possible Error Effect:	This packet can not be transmitted.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL packet buffer usage field.
<i>Bad Data</i> =	Illegal x3header logical channel value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DDL packet buffer number.
<i>Dump</i> =	DDL packet buffer.

Table 16 Error Code: MLNKGRP

Error Code:	MLNKGRP
Error Description:	The usage link group for this message buffer is out of range.
Possible Error Effect:	This could cause a lost message buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL message buffer usage field.
<i>Bad Data</i> =	Illegal usage link group value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DDL message buffer number.
<i>Dump</i> =	DDL message buffer usage.

Table 17 Error Code: MSGPTR

Error Code:	MSGPTR
Error Description:	The message buffer is not on the correct list or buffer.
Possible Error Effect:	This could cause a lost message buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL message buffer usage field.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Usage type value.
<i>Logical Index</i> =	DDL message buffer number.
<i>Dump</i> =	DDL message buffer usage.

Table 18 Error Code: MUINDEX

Error Code:	MUINDEX
Error Description:	The usage index for this message buffer was out of range.
Possible Error Effect:	This could cause a lost message buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL message buffer usage field.
<i>Bad Data</i> =	Illegal usage index value.
<i>Good Data</i> =	Use type value.
<i>Logical Index</i> =	DDL message buffer number.
<i>Dump</i> =	DDL message buffer usage.

Table 19 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The head block or the tail block of the free/rxmit/busy Queue specified by CMmsgbuf.bq[] was found to have an invalid linkage specification. The head block's qprev (previous) linkage and the tail block's qnext (next) linkage should both be marked null (CMEEMPTY).
Possible Error Effect:	Possible exhaustion of CMmsgbuf.bq[] message queue write buffer resources.
Corrective Action Taken:	The head block's qprev (previous) linkage and the tail block's qnext (next) linkage are both marked null (CMEEMPTY).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the head block (& CMmsgbuf.bq[]).
<i>Bad Data</i> =	The head block's qprev (previous) linkage.
<i>Good Data</i> =	The null state both head and tail block linkages should be (CMEEMPTY).
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	The contents of the head block under audit.
<i>Dump 2</i> =	The contents of the tail block under audit.

Table 20 Error Code: PKTPTR

Error Code:	PKTPTR
Error Description:	The DDL packet buffer is not on the correct list or buffer.
Possible Error Effect:	This could cause a lost packet buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL packet buffer usage field.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Usage type value.
<i>Logical Index</i> =	DDL packet buffer number.
<i>Dump</i> =	DDL packet buffer usage.

Table 21 Error Code: PKT_TYPE

Error Code:	PKT_TYPE
Error Description:	An RSM DDL packet buffer was found with an unknown usage type.
Possible Error Effect:	This could cause a lost packet buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL packet buffer usage field.
<i>Bad Data</i> =	Illegal usage type value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DDL packet buffer number.
<i>Dump</i> =	DDL packet buffer usage.

Table 22 Error Code: PLNKGRP

Error Code:	PLNKGRP
Error Description:	The usage link group for this packet buffer is out of range.
Possible Error Effect:	This could cause a lost packet buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL packet buffer usage field.
<i>Bad Data</i> =	Illegal usage group value.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	DDL packet buffer number.
<i>Dump</i> =	DDL packet buffer usage.

Table 23 Error Code: PUINDEX

Error Code:	PUINDEX
Error Description:	The usage index for this packet buffer was out of range.
Possible Error Effect:	This could cause a lost packet buffer.
Corrective Action Taken:	The CM DDL protocol data structures have been initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DDL packet buffer usage field.
<i>Bad Data</i> =	Illegal usage index value.
<i>Good Data</i> =	Usage use type value.
<i>Logical Index</i> =	DDL packet buffer number.
<i>Dump</i> =	DDL packet buffer usage.

Table 24 Error Code: SIZE

Error Code:	SIZE
Error Description:	The size marked in the CMecbusy[] busy queue array block entry under audit is invalid. It should equal the calculated size of the queue.
Possible Error Effect:	Possible exhaustion of CMecbusy[] busy queue resources.
Corrective Action Taken:	The size is corrected to the calculated size of the queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The contents of the CMecbusy[] busy queue array entry under audit.
<i>Bad Data</i> =	The value of CMecbusy[].size attribute that was detected.
<i>Good Data</i> =	The calculated size of the queue.
<i>Logical Index</i> =	The index if the CMecbusy[] busy queue array entry under audit.
<i>Dump</i> =	The contents of the CMecbusy[] busy queue array entry under audit.

Table 25 Error Code: ZAP

Error Code:	ZAP
Error Description:	A machine reset has occurred while the CMIM audit was running. Since the CMIM audit range-checks all message buffer addresses, it is highly unlikely that the audit caused the reset through its actions.
Possible Error Effect:	A machine reset has occurred.
Corrective Action Taken:	If the machine reset occurred while the CMIM audit was manipulating CM-SM data then CM-SM was automatically initialized before the audit was invoked in AUZAP manner. The audit will therefore not take any recovery action when called in AUZAP manner except to return to audit control.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUcursor.
<i>Bad Data</i> =	AUcursor.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

CMPKT

Software Release: 5E10 and later

1. COMMUNICATION MODULE MESSAGE PACKETIZATION (CMPKT) AUDIT

The communication module message packetization (CMPKT) audit is used to audit the communication module's (CM) new incoming packet buffer array used to store the incoming packetized messages. Several operating system for distributed switching (OSDS) message types are expected to exceed the existing 254 byte limit for OSDS inter-processor text message size. Intra-processor messages currently can be up to 300 bytes in size [message control block (MCB) size limit] but only for special cases (such as the MGRT_GEN message). As a result of the need for large messages, the 5ESS[®] switch requires a means of delivering messages greater than 254 bytes in size between application processors. Inter-processor message packetization will provide a mechanism for delivering large message types, both intra- and inter-processor. Large intra-processor messages will be delivered to a process by OSDS.

There is one exception. *UNIX*[®] to OSDS messages in the administrative module (AM) cannot increase in size so they cannot be delivered with this mechanism. Inter-processor OSDS messages which are greater than 254 bytes in size will be packetized by communication package (CM subsystem) software, which is resident in each processor. On the receiving processor these packetized messages will be re-assembled by the CM subsystem, prior to being handed off to OSDS which will pass the message along to the receiving process. The processors that will be supported by this capability are: AM-operational kernel process (OKP), switching module (SM), communication module processor (CMP) and direct link node (DLN).

The CMPKT audit runs in AM-OKP, CMP and SM processors. Since no audit code resides in the DLN processor, a trigger will be fired every 5 minutes to run the audit code from DLN operational code and asserts will be fired for error conditions.

2. DATA STRUCTURES

The primary data structure used to store the incoming message packets in CM subsystem is packet buffer array, `Cmpktmbuf[ ]`, of type `CMPKTBUF`. Each processor has an array of `CMPKTBUF`. When not in use, each packet buffer is linked to the head cell for the packet buffer free list, `Cmpktidle[ ]`, variable `toppt`, also defined in `CMPBUFS` data structure, is used to keep track of the top of the stack where the next buffer should be linked. This variable is not audited and is monitored by operational code. Figure 1 shows these linkages.

When a packet of a message arrives, a packet buffer is taken from the free list and is linked to the packet head table, `Cmpktpkthdtbl`, of type `CMPKTHDTBL`. Figure 2 shows these linkages.

Figure 3 shows the case where we allow messages that use more than two packets to deliver a message. When all packets for a message have arrived, the message is delivered to the appropriate destination processor in AM-OKP and CMP. In the SM, it is linked to the incoming message queue, `inq`, of type `cmINQ`, and included in the data structure `Cmlinkdata`. It is delivered from there. Figure 4 shows these linkages.

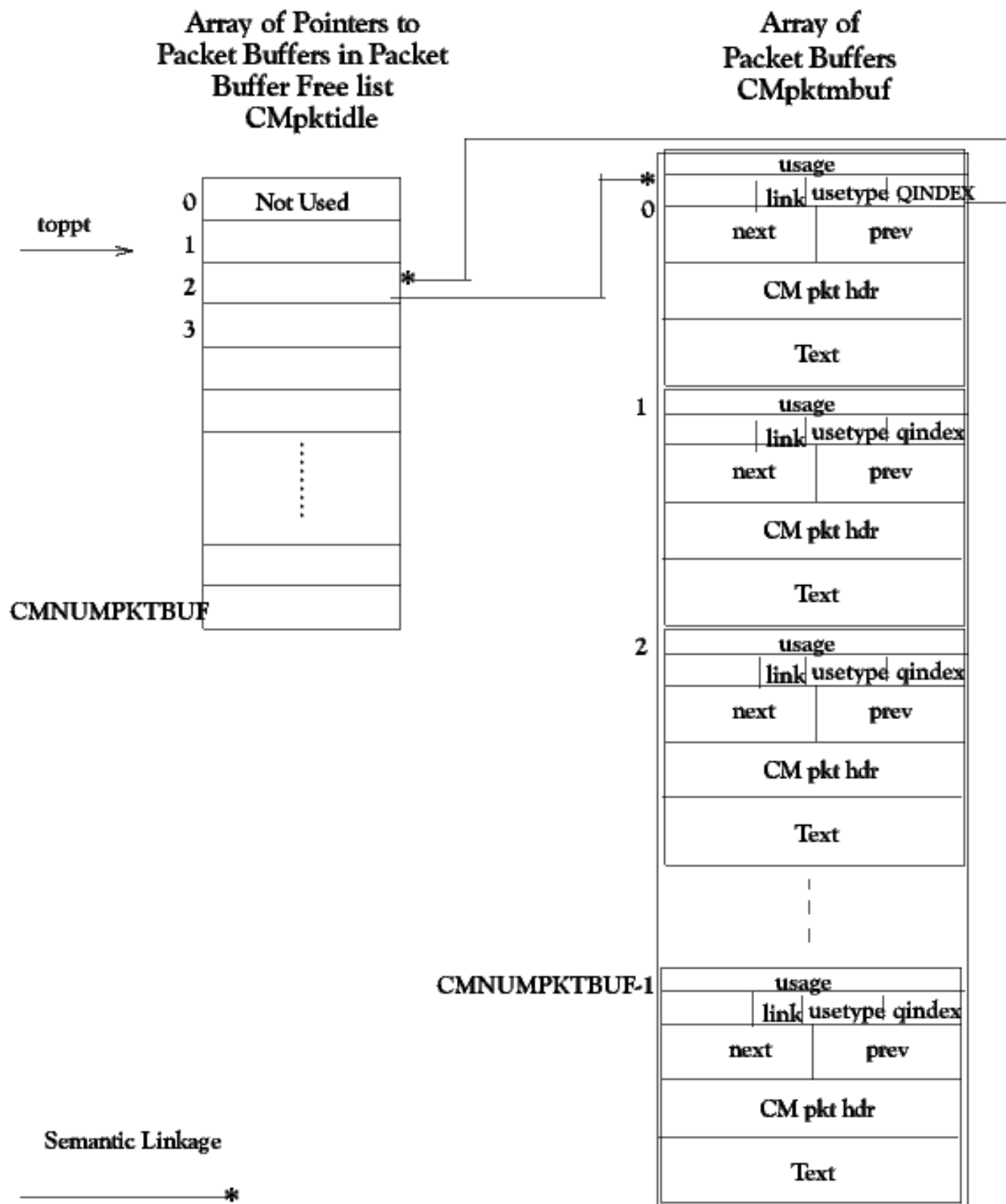


Figure 1 (CMPKT-1) Linkages to the Packet Buffer Free List

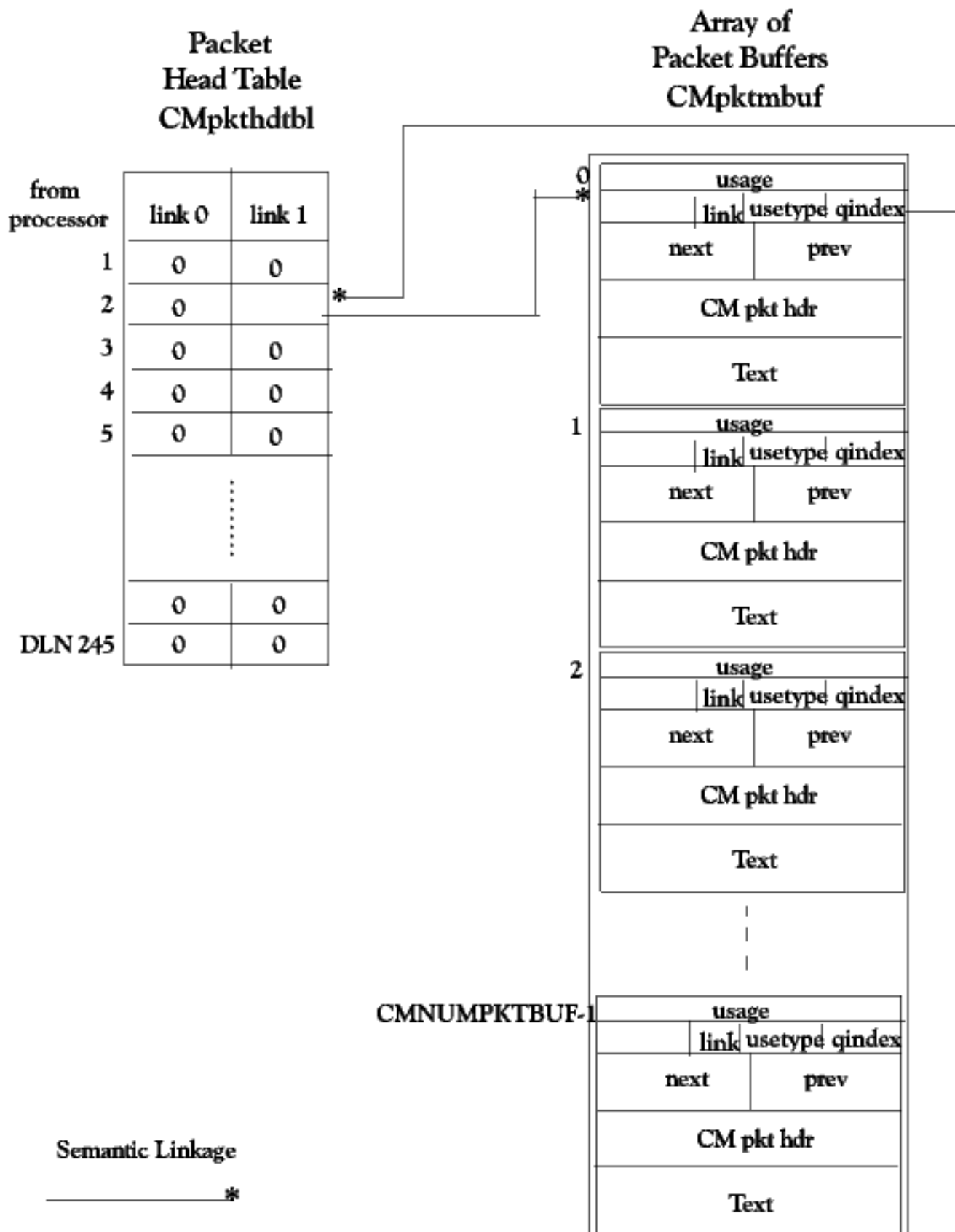
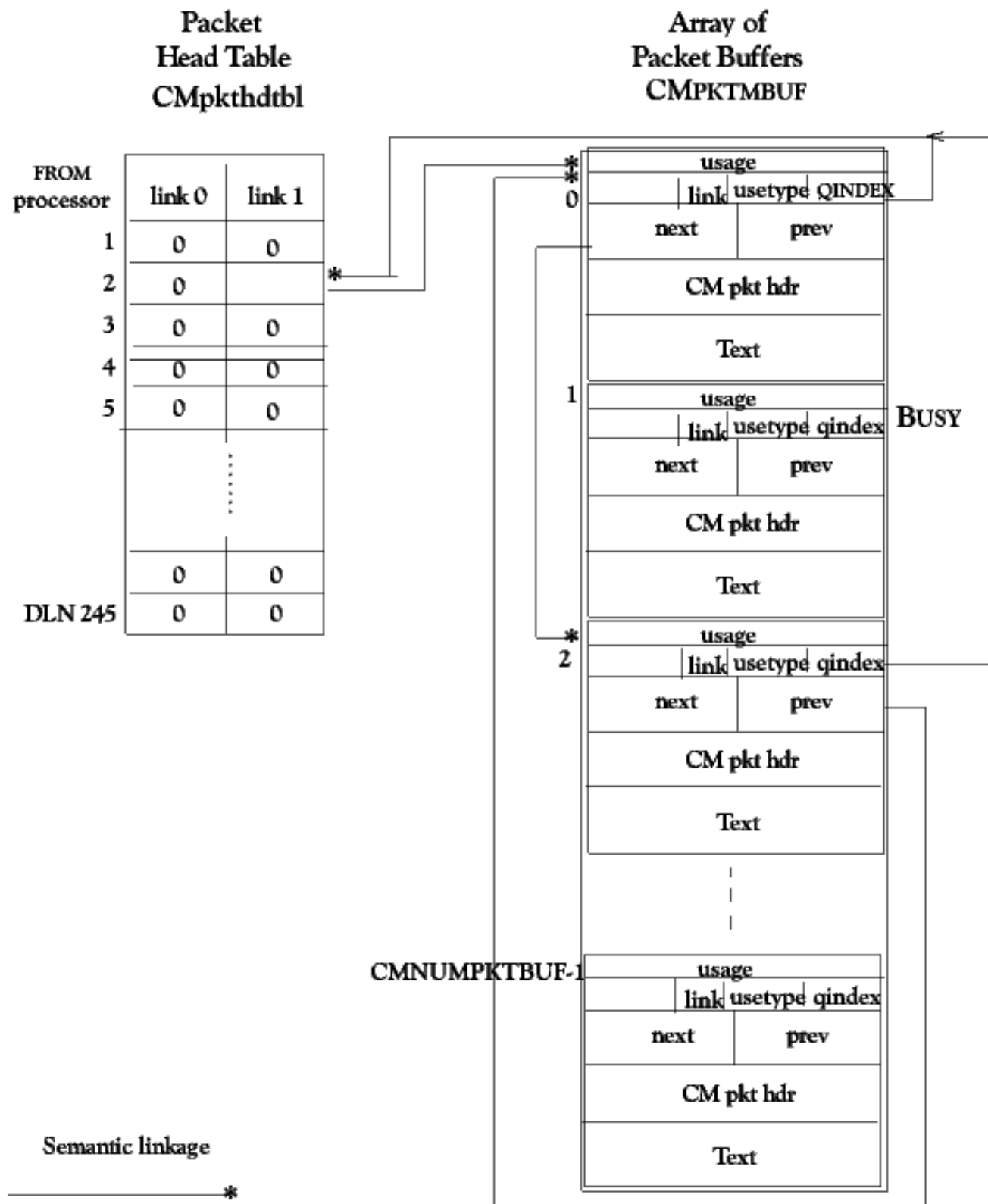
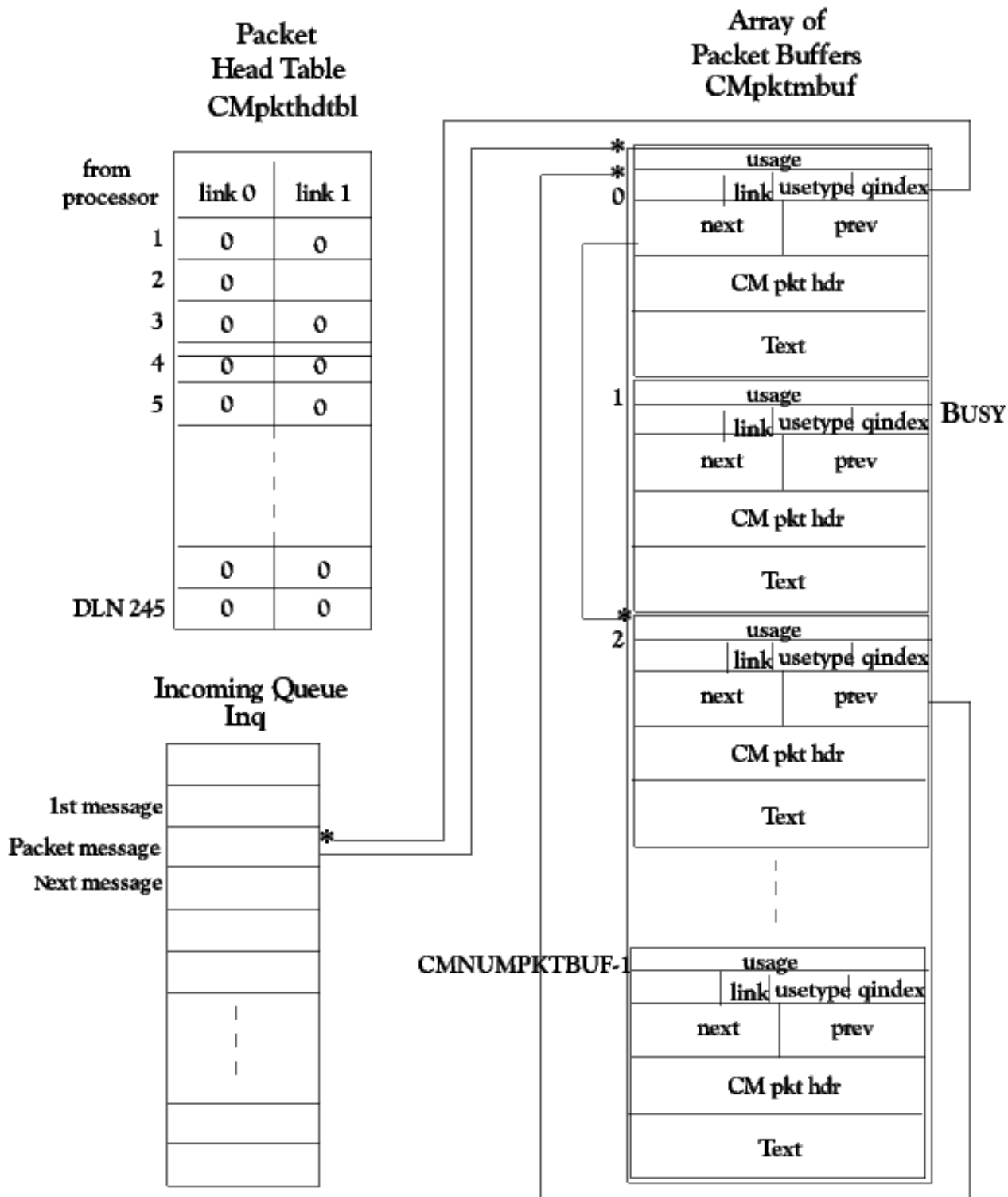


Figure 2 (CMPKT-2) One Packet of Message Received (All Processors) Linkages for Packet Buffer Receive State



(for SM only case and for OKP/CMP maximum number of packets >2)
 (Note: This will be a transient state if maximum number of packets is 2).

Figure 3 (CMPKT-3) Second Packet of Message Received



(No incoming queue for OKP/CMP. Messages are delivered immediately when all packets are received.)

Figure 4 (CMPKT-4) All Packets of Message Received (SM only)**3. AUDIT CHECKS**

The CMPKT audit does not use any audit drivers. The audit uses the following error codes which gives the general flow of the audit. For details about each error code refer to the error comment blocks below. If the audit is just starting

BDLINK	= Check linkages of packet buffer in SM.
BLINK	= Check linkages of packet buffer in AM-OKP, CMP.
INV_STATE	= Check usetype field.
TIMESTAMP	= Check the au_visitcnt variable.
ZAP	= The audit process took system interrupt (that is, wild write).

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BLINK

Error Code:	BLINK
Error Description:	The packet buffer, indexed by logical key, failed its linkage checks in last two audit cycles either in the CMP or in AM-OKP.
Possible Error Effect:	Possible incomplete message reception that will result in retransmission of the same message.
Corrective Action Taken:	The packet buffer will be recovered and put back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the current packet buffer being audited.
<i>Bad Data =</i>	The contents of the packet buffer's usage.usetype field. This field tells us whether the packet is either on the free list (CMFREES) or in the receive mode (CMPKTRCV) or is linked to incoming queue (CMPKTINQ) in SM waiting to be delivered.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The index of the packet buffer being recovered.
<i>Dump =</i>	The contents of the packet buffer in error.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The packet buffer, indexed by logical key, failed its linkage checks in last two audit cycles in the SM. It also failed the linkage checks for the third time in this cycle of the audit, when all the non-maskable interrupts were blocked during a critical region.
Possible Error Effect:	Possible incomplete message reception that will result in retransmission of the same message.
Corrective Action Taken:	The packet buffer will be recovered and put back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the current packet buffer being audited.
<i>Bad Data =</i>	The contents of the packet buffer's usage.usetype field. This field tells us whether the packet is either on the free list (CMFREES) or in the receive mode (CMPKTRCV) or is linked to incoming queue (CMPKTINQ) in SM waiting to be delivered.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	The index of the packet buffer being recovered.
<i>Dump</i> =	The contents of the packet buffer in error.

Table 3 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The packet buffer, indexed by logical key, was found to have an invalid usetype (usage.usetype) during its linkage checks. The valid values are packet on the free list (CMFRRS), packet in the receive state (CMPKTRCV) and packet linked to incoming queue (CMPKTINQ, for SM only).
Possible Error Effect:	Possible incomplete message reception that will result in retransmission of the same message.
Corrective Action Taken:	The packet buffer will be recovered and put back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the current packet buffer being audited.
<i>Bad Data</i> =	The contents of the packet buffer's usage.usetype field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index of the packet buffer being recovered.
<i>Dump</i> =	The contents of the packet buffer in error.

Table 4 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The audit found a packet that has been marked visited for last two cycles of the audit and has not moved in the current cycle also. This can happen if this packet is waiting for next packet of the same message to arrive and that packet never arrives due to initialization, CLNKS down or other such abnormal condition and CM operational code could not recover it.
Possible Error Effect:	Possible incomplete message reception that will result in retransmission of the same message.
Corrective Action Taken:	The packet buffer will be recovered and put back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The contents of the packet buffer's usage.usetype field. This field tells us whether the packet is either on the free list (CMFREES) or in the receive mode (CMPKTRCV) or is linked to incoming queue (CMPKTINQ) in SM waiting to be delivered.
<i>Bad Data</i> =	The value of the packet buffer's visit count field.
<i>Good Data</i> =	2, the maximum allowable number of the au_visitcnt field.
<i>Logical Index</i> =	The index of the packet buffer being recovered.
<i>Dump</i> =	The contents of the packet buffer in error.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as a result of a write protect, out of range read or invalid operation by any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

11. CN - CP

CNAMDATA

Software Release: 5E10 and later

1. INDIVIDUAL CALLING LINE IDENTIFICATION CALLING NAME DATA (CNAMDATA) AUDIT

The RLCNAMDATA dynamic relation is used by the LASS feature calling name (CNAM) delivery to store several billing counts. These counts are:

- The number of times that the calling name was delivered to a customer premise equipment (CPE).
- The number of times the calling name was "unavailable".
- The number of times that both the calling name and the calling number are delivered to the CPE.
- The number of times that both the calling name and the calling number are "unavailable".
- The number of times that the calling name was delivered to the CPE and the calling number was "unavailable".
- The number of times that the calling name was "unavailable" and the calling number was delivered to the CPE.

NOTE: "Unavailable" either means that "O" (out-of-area) was delivered to the CPE indicating that the calling name or number was unavailable or "P" (private) was delivered to the CPE indicating that the calling name or number had privacy (default or per-call) set.

The CNAMDATA audit performs dynamic to static existence checks of the dynamic RLCNAMDATA relation. Tuples in RLCNAMDATA should only exist for usage sensitive only rICLIDIDX tuples. This audit runs segmented in the switching module and is also scheduled as a post initialization audit as part of a selective initialization.

2. DATA STRUCTURES

The RLCNAMDATA relation is a dynamic indirect access nonsequential keyed relation. The CNAMDATA audit verifies the proper existence of rICNAMDATA tuples against the related static relation RLICLIDIDX.

3. AUDIT CHECKS

The CNAMDATA audit steps through the RLCNAMDATA relation and verifies whether or not a tuple in the relation should exist.

4. ERROR HANDLING

If a rICNAMDATA tuple exists that should not, the audit removes the tuple from the RLCNAMDATA relation.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rICNAMDATA dynamic tuple existing for a corresponding rICLIDIDX tuple which is not usage-sensitive.
Possible Error Effect:	If call processing attempts to access this tuple, services/operations may be disrupted.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the CNAMDATA tuple.
<i>Bad Data</i> =	Address of the ICLIDIDX buffer.
<i>Good Data</i> =	GLSUCCESS.
<i>Logical Index</i> =	The key of the rICNAMDATA tuple in question.
<i>Dump</i> =	Buffer filled for related static rICLID_IDX tuple.

Table 2 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The audit is unable to get the related static rICLID_IDX tuple for this rICNAMDATA tuple.
Possible Error Effect:	The itm_idx or itm_type fields of the rICNDDATA tuple may be incorrectly populated. This may result in this rICNAMDATA tuple being unavailable for ICLID feature operations.
Corrective Action Taken:	The tuple will be removed and if necessary the related audit ICLIDIDX will insert the tuple with correct information.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not an address; itm_idx field of rICNDDATA tuple.
<i>Bad Data</i> =	itm_type of the rICNDDATA tuple.
<i>Good Data</i> =	Return from primitive used to access related static RLICLID_IDX.
<i>Logical Index</i> =	The key of the rICNDDATA tuple in question.
<i>Dump 1</i> =	rICNDDATA tuple in question.
<i>Dump 2</i> =	Buffer filled for related static rICLID_IDX tuple.

CNDDATA

Software Release: 5E10 and later

1. INDIVIDUAL CALLING LINE IDENTIFICATION CALLING NUMBER DISPLAY DATA (CNDDATA) AUDIT

The RLCNDDATA relation is used by the LASS feature individual calling line identification (ICLID) to store the number of calling number display (CND) messages with directory number (DN) delivered and the number of CND messages out of LATA or marked as private. The CNDDATA audit performs dynamic to static existence checks of the dynamic RLCNDDATA relation as well as several semantic data checks. This audit runs segmented in the Switching Module and is also scheduled as a post initialization audit as part of a selective initialization.

2. DATA STRUCTURES

The RLCNDDATA relation is a dynamic indirect access nonsequential keyed relation. The CNDDATA audit verifies the proper existence of rCNDDATA tuples as well as verify the semantic **ic_idx**, **lead_module** and **ama_rec** fields against that of the related static relation RLICLIDIDX.

3. AUDIT CHECKS

The CNDDATA audit steps through the RLCNDDATA relation and verifies whether or not a tuple in the relation should exist as well as perform necessary semantic data checks against the related static relation RLICLIDIDX.

4. ERROR HANDLING

If a rCNDDATA tuple exists that should not, the audit removes the tuple from the RLCNDDATA relation. If the audit detects an error with any semantic data, the semantic data is corrected to match that of the associated static relation RLICLIDIDX.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The audit is unable to get the related static rICLID_IDX tuple for this rCNDDATA tuple.
Possible Error Effect:	The ltm_idx or ltm_type fields of the rCNDDATA tuple may be incorrectly populated. This may result in this rCNDDATA tuple being unavailable for ICLID feature operations.
Corrective Action Taken:	The tuple will be removed and if necessary the related audit ICLIDIDX will insert the tuple with correct information.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; ltm_idx field of rCNDDATA tuple.
<i>Bad Data =</i>	ltm_type of the rCNDDATA tuple.
<i>Good Data =</i>	Return from primitive used to access related static RLICLID_IDX.
<i>Logical Index =</i>	The key of the rCNDDATA tuple in question.
<i>Dump 1 =</i>	rCNDDATA tuple in question.
<i>Dump 2 =</i>	Buffer filled for related static rICLID_IDX tuple.

Table 2 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The rCNDDATA tuple's ic_idx , lead_module , or ama_rec fields do not match those of the related static rICLID_IDX tuple.

Possible Error Effect:	The rICNDDATA tuple may be garbaged and unavailable to ICLID feature operation.
Corrective Action Taken:	The audit will zero the tuple in question and get it initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; lead_module field of the related static rIICLID_IDX tuple.
<i>Bad Data =</i>	lead_module field of the rICNDDATA tuple.
<i>Good Data =</i>	ic_idx field of the related static rIICLID_IDX tuple.
<i>Logical Index =</i>	The key of the rICNDDATA tuple in question.
<i>Dump 1 =</i>	rICNDDATA tuple in question.
<i>Dump 2 =</i>	Global buffer of related static rIICLID_IDX tuple.

Table 3 Error Code: STUCK

Error Code:	STUCK
Error Description:	The audit is unable to initialize a rICNDDATA tuple found in error.
Possible Error Effect:	The rICNDDATA tuple may be garbaged and unavailable to ICLID feature operation.
Corrective Action Taken:	The audit will try to recover the tuple on the next pass. If the error continues there is probably a problem with the database primitives used to insert data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; itm_idx field of related static rIICLID_IDX tuple.
<i>Bad Data =</i>	itm_type of related static rIICLID_IDX tuple.
<i>Good Data =</i>	Return from function used to initialize rICNDDATA tuple; this value is same as that returned by the database primitives used to insert the good data.
<i>Logical Index =</i>	The key of the rICNDDATA tuple in question.
<i>Dump 1 =</i>	rICNDDATA tuple in question.
<i>Dump 2 =</i>	Buffer filled for related static rIICLID_IDX tuple.

COINOT

Software Release: 5E10 and later

1. AUTOMATED COIN (COINOT) AUDIT

The COINOT audit examines each rICOINOT tuple to perform key, quarstate, member linkage (point-to and point-back), and one-to-one linkage checks.

2. DATA STRUCTURES

Figure 1 illustrates the basic audit structure.

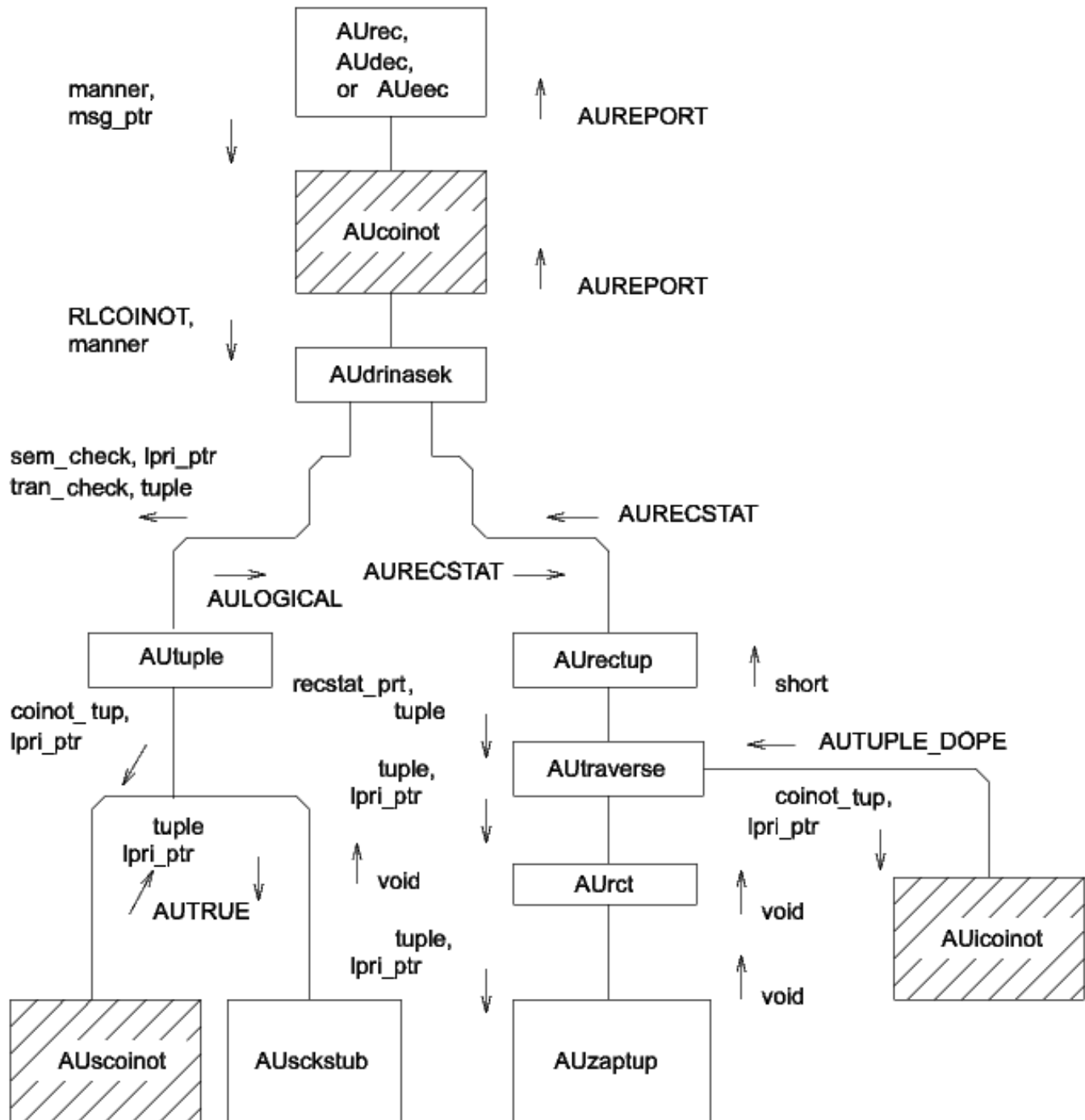


Figure 1 (COINOT-1) COINOT Audit Structure Chart

3. AUDIT CHECKS

The RLCOINOT relation is only populated on SMs with ISPOPR or OPRISP trunk. This audit will verify that the SM has these trunks. If the SM does not, the audit will not check the relation since it is not populated. Otherwise, the audit will check the keys for each tuple. It will also check to see that if the tuple is not in use, it is connected to its idle head cell and that if it is in use, it is linked to its busy head cell. If the tuple is not linked to the proper head cell, an error is reported; the tuple is reinitialized and put back onto the idle list.

4. ERROR HANDLING

The AUcoinot audit calls the AUDrinasek driver. AUTuple() steps through the RLCOINOT relation performing key, quarstate, member linkage, and one-to-one link checks. If valid, it goes on to invoke the semantic function, AUScoinot().

When an error is found in the rICOINOT tuple, the tuple is zeroed, the key is initialized, and the tuple is linked to its idle head cell. The rICOINOT tuple contents are printed. It also schedules the PCBLA audit to run in the elevated mode.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rICOINOT tuple was not busy when it should be.
Possible Error Effect:	The rICOINOT tuple will be lost to the system.
Corrective Action Taken:	Zap the tuple and place it on the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICOINOT.
<i>Bad Data</i> =	Actual owner's physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	rICOINOT number.
<i>Dump</i> =	rICOINOT.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rICOINOT tuple is not idle when it should be.
Possible Error Effect:	The rICOINOT tuple will be lost to the system.
Corrective Action Taken:	Zap the tuple and place it on the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICOINOT.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	rICOINOT number.
<i>Dump</i> =	rICOINOT.

CONF C

Software Release: 5E10 and later

1. CONFERENCE (CONF C) AUDIT

Conference circuits are used for conference calling and for verifying that busy lines are actually busy. The conference circuits are part of the global digital service unit (GDSU) and are resources that can be accessed from any switching module (SM) in an office. Each office has a specified number of conference circuits; therefore, it is important that these resources be operational. The conference circuit audit (CONF C) is an interprocessor audit that verifies data in the allocation table, which is located in the communication module processor (CMP) against information in the SM. It verifies the information that cannot be audited by the intraprocessor conference circuit audits: the global service circuit audit (GCKTI) and the conference circuit access table audit (CCATBL).

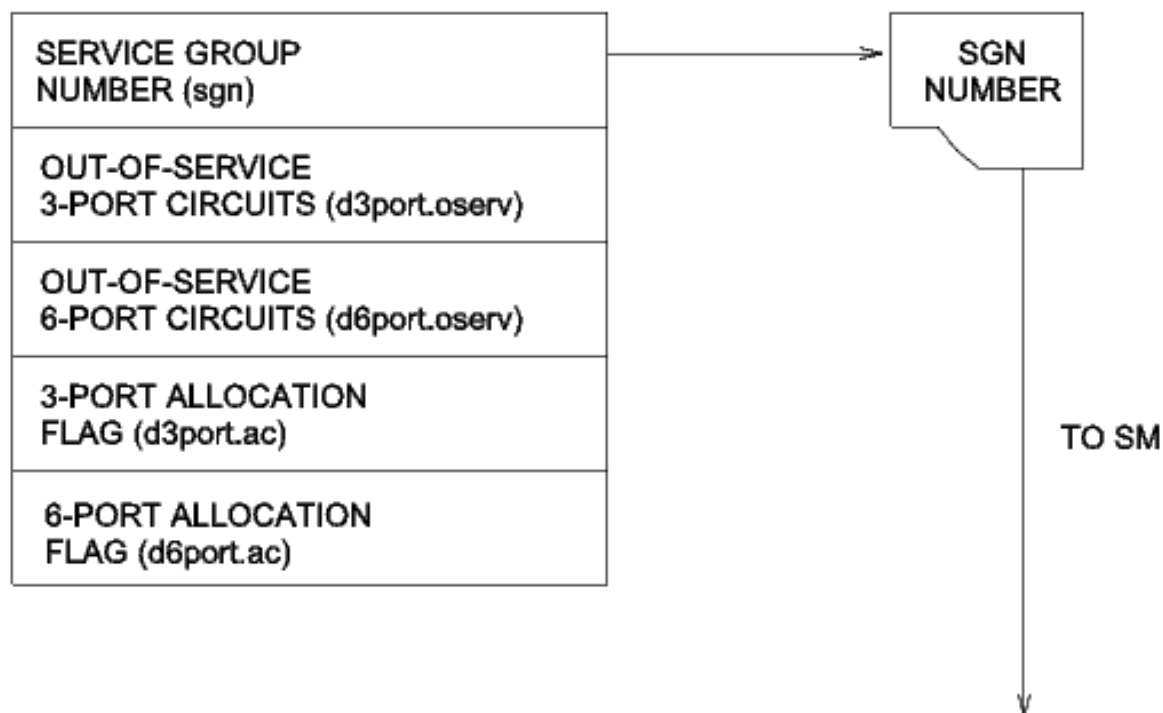
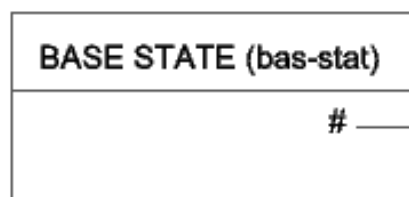
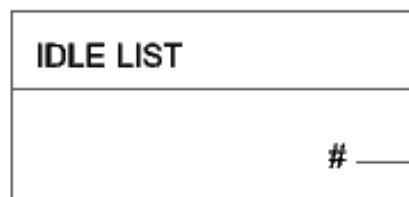
This audit is part of the Peripheral Control audits subgroup. It runs in the CMP and SM environments in the segmented mode.

2. DATA STRUCTURES

The data structure verified by the CONF C audit is the conference circuit access table relation (RLCCATBL) located in the CMP. The information checked by the audit includes the allocation flag (ac), which indicates whether there are idle conference circuits in the SM corresponding to a rLCCATBL tuple, and the number of out-of-service (oserv) circuits in the SM. The audit also refers to the following data:

RLCDBCOM	= rLCDBCOM tuples are used to determine the operational status of the conference circuits.
RLPICB	= The rLPICB tuple is used to calculate the circuit name of the conference circuit.
RLSMEST	= rLSMEST tuples are used to determine the in-service/out-of-service maintenance status of the conference circuits.

Figure 1 illustrates the items checked by the audit.

RLCCATBL**CMP****SM****RLSMEST****RLCDBCOM**

**OOS AND
IDLE LIST
STATUS**

Arrows from the **#** fields of both **RLSMEST** and **RLCDBCOM** point to this box. An arrow points up from the box.

Figure 1 (CONFC-1) Data Structures

3. AUDIT CHECKS

In the CMP, the following audit checks are performed. If the audit has just been entered:

ZAP = Check if the audit has been entered due to a system interrupt.

After the audit sends a message to a SM from the CMP to collect conference circuit status and receives the reply:

- ACBIT3P = Check if the allocation flag for 3-port conference circuits in the CMP is consistent with data collected from the SM.
- ACBIT6P = Check if the allocation flag for 6-port conference circuits in the CMP is consistent with data collected from the SM.
- OOSCNT3P = Check if the OOS 3-port conference circuit count in the CMP is consistent with data collected from the SM.
- OOSCNT6P = Check if the OOS 6-port conference circuit count in the CMP is consistent with data collected from the SM.

In the SM, the following audit checks are performed.

If the audit has just been entered:

- ZAP = Check if the audit has been entered due to a system interrupt.

Collect conference circuit data:

- CKTDSMEST = Check for failure to access the rISMEST tuple).
- SGNODD = Check for failure to access the rPICB tuple).

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACBIT3P

Error Code:	ACBIT3P
Error Description:	The 3-port conference circuit allocation flag (d3port.ac) in the rICCATBL tuple has been found to be different from the operational state of the conference circuits in the SM on two consecutive passes of the audit. For each GDSU service group in a SM, a message is sent from the CMP to the SM to collect data for all 3-port and 6-port conference circuits. In the SM the operational status for 3-port conference circuits is retrieved using data in the rCDBCOM tuple. If at least one rCDBCOM tuple associated with a 3-port conference circuit is linked onto the idle list (cdccm.st_ownk = DBSG0CONF or DBSG1CONF) the allocation flag should be turned on; otherwise, the allocation flag should be off. The allocation flag data is then send to the CMP and compared to the allocation flag (d3port.ac) in the rICCATBL tuple.
Possible Error Effect:	If the allocation flag is off in the CMP and should be on according to data in the SM this will stimulate blocking of conference calls; otherwise, the CMP may try to allocate conference circuits when none are available.
Corrective Action Taken:	The allocation flag (d3port.ac) in the rICCATBL for 3-port conference circuits will be recovered to the value of the master data received from the SM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	3-port allocation flag (d3port.ac) from the rICCATBL tuple.

<i>Good Data</i> =	3-port allocation data from the SM.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple that was in error.

Table 2 Error Code: ACBIT6P

Error Code:	ACBIT6P
Error Description:	The 6-port conference circuit allocation flag (d6port.ac) in the rICCATBL tuple has been found to be different from the operational state of the conference circuits in the SM on two consecutive passes of the audit. For each GDSU service group in a SM, a message is sent from the CMP to the SM to collect data for all 3-port and 6-port conference circuits. In the SM the operational status for 6-port conference circuits is retrieved using data in the rCDBCOM tuple. If at least one rCDBCOM tuple associated with a 6-port conference circuit is linked onto the idle list (cdccm.st_ownk = DBSG0CONF or DBSG1CONF) the allocation flag should be turned on; otherwise, the allocation flag should be off. The allocation flag data is then sent to the CMP and compared to the allocation flag (d6port.ac) in the rICCATBL tuple.
Possible Error Effect:	If the allocation flag is off in the CMP and should be on according to data in the SM this will stimulate blocking of conference calls; otherwise, the CMP may try to allocate conference circuits when none are available.
Corrective Action Taken:	The allocation flag (d6port.ac) in the rICCATBL for 6-port conference circuits will be recovered to the value of the master data received from the SM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICCATBL tuple found in error.
<i>Bad Data</i> =	6-port allocation flag (d6port.ac) from the rICCATBL tuple.
<i>Good Data</i> =	6-port allocation data from the SM.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple that was in error.

Table 3 Error Code: CKTDSMEST

Error Code:	CKTDSMEST
Error Description:	Information in the rICCATBL dynamic tuple in the CMP and the rPICB static tuple in the SM indicates that an rSMEST dynamic tuple should exist for a particular conference circuit but it does not.
Possible Error Effect:	A rSMEST tuple is lost to the system and could cause asserts from maintenance application code and will cause GDSU remove and restoration failures.
Corrective Action Taken:	The CKTDATA audit will be schedule to insert the rSMEST tuple found missing. However, if the CKTDATA audit does not insert the rSMEST tuple there may be static data inconsistencies between the RLCKTDATA and RLPICB relations.
Manual Action Required:	If this error rolls out then, check the SM hardware to see if the conference circuits board, specified in the <i>Bad Data</i> field of this dump, is equipped. If the conference circuit is equipped: - If the circuit type (type) matches the hardware then check the RLPICB relation for the correct data in its board field (brd). The 3-port conference circuit board value is PCCONF3, the cost-reduced (UCC2) 3-port conference circuit board value is PCCONF9, and the 6-port conference circuit board value is PCCONF6. - If the circuit type (type) does not match the hardware, update the rICKTDATA tuple using ODBE. - If there is no entry in rICKTDATA, then create one using ODBE.

	If the conference circuit is not equipped, then the rPICB static tuple in the SM must be updated as well as the rIGCKTI static tuple in the CMP to indicate that this GSDU service group does not have this conference circuit board equipped.
Dump Description:	
<i>Error Address</i> =	Address of rISMEST tuple buffer.
<i>Bad Data</i> =	Circuit name of the conference circuit found in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: OOSCNT3P

Error Code:	OOSCNT3P
Error Description:	The 3-port conference circuit OOS count (d3port.oserv) in the rICCATBL tuple has been found to be different from the maintenance state of the conference circuits in the SM on two consecutive passes of the audit. For each GDSU service group in a SM, a message is sent from the CMP to the SM to collect data for all 3-port and 6-port conference circuits. In the SM the OOS data for 3-port conference circuits is retrieved using data in the rISMEST tuple. If the state of the rISMEST tuple is OOS (bas_stat = SMOOS), unequipped (qual_1 = SMUN), or in a grow (qual_1 = SMGR) state, the OOS count is incremented by 2 for standard conference circuits (PCCONF3) and 5 for cost-reduced conference circuit (PCCONF9). The OOS count is then send to the CMP and compared to the OOS data (d3port.oserv) in the rICCATBL tuple.
Possible Error Effect:	If the OOS count is larger in the CMP than the SM this will stimulate blocking of conference calls; otherwise, the CMP may try to allocate OOS conference circuits.
Corrective Action Taken:	The OOS count (d3port.oserv) in the rICCATBL for 3-port conference circuits will be recovered to the value of the master data received from the SM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCATBL tuple in error.
<i>Bad Data</i> =	3-port OOS count (d3port.oserv) from the rICCATBL tuple.
<i>Good Data</i> =	3-port OOS count data from the SM.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple that was in error.

Table 5 Error Code: OOSCNT6P

Error Code:	OOSCNT6P
Error Description:	The 6-port conference circuit OOS count (d6port.oserv) in the rICCATBL tuple has been found to be different from the maintenance state of the conference circuits in the SM on two consecutive passes of the audit. For each GDSU service group in a SM, a message is sent from the CMP to the SM to collect data for all 3-port and 6-port conference circuits. In the SM the OOS data for 6-port conference circuits is retrieved using data in the rISMEST tuple. If the state of the rISMEST tuple is OOS (bas_stat = SMOOS), unequipped (qual_1 = SMUN), or in a grow (qual_1 = SMGR) state, the OOS count is incremented by 2 for standard conference circuits (PCCONF6). The OOS count is then send to the CMP and compared to the OOS data (d6port.oserv) in the rICCATBL tuple.
Possible Error Effect:	If the OOS count is larger in the CMP than the SM this will stimulate blocking of conference calls; otherwise, the CMP may try to allocate OOS conference circuits.

Corrective Action Taken:	The OOS count (d6port.oserv) in the rICCATBL for 6-port conference circuits will be recovered to the value of the master data received from the SM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICCATBL tuple in error.
<i>Bad Data</i> =	6-port OOS count (d6port.oserv) from the rICCATBL tuple.
<i>Good Data</i> =	6-port OOS count data from the SM.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple that was in error.

Table 6 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	The rPICB value passed from the CMP does not match any value in this SM. There is an ODD inconsistency between rLGCKTI tuple in the CMP and the rPICB tuple in the SM.
Possible Error Effect:	The CONF C audit will never be able to count the circuits in this SM, which may lead to blocking of three way calls.
Corrective Action Taken:	None.
Manual Action Required:	Using the <i>Bad Data</i> of this dump insure that the information in the rPICB static tuple in the SM is consistent with the information in the rLGCKTI static tuple in the CMP.
Dump Description:	
<i>Error Address</i> =	Address of rPICB tuple buffer.
<i>Bad Data</i> =	PICB number of the GSDU SG in this SM.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 7 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after taking a system interrupt (such as, write protect, out-of-range read, and so forth).
Possible Error Effect:	May cause further resets, audits, and/or asserts.
Corrective Action Taken:	Zero the partition cursor and the global variable that contains the index into the RLCCATBL relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Contents of the global variable containing the index into the RLCCATBL relation.
<i>Bad Data</i> =	Contents of the partition cursor.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

CPEST-A

Software Release: 5E10 - 5E13

1. CENTRAL PROCESSOR EQUIPMENT STATUS TABLE (CPEST) AUDIT

The maintenance request administrator (MRA) program maintains the in-service and out-of-service statuses of all the hardware circuits in a switching module (SM) in the switch maintenance equipment status table (RLSMEST) except the metallic test interconnection buses (MTIBs). MRA maintains the statuses of the MTIBs in the central processor equipment status relation (RLCPEST) because the MTIBs are globally allocated in the communications module processor (CMP). The MTIBs connect metallic services units (MSUs) in all the SMs; the MSUs contain the circuitry for performing automatic line insulation tests and other metallic tests, and provide compensation for leakage when lines are being tested. The CPEST audit ensures the integrity of the RLCPEST relation. The CPEST audit is a part of the switch maintenance audits subgroup. It runs in the CMP in the routine and elevated segmented modes.

2. DATA STRUCTURES

The rLCPEST tuple keeps the status of the MTIBs in the CMP. As Figure 1 shows, the tuples of the CPEST relation are linked to tuples of the MSU path data relation (RLMPHDATA), which is audited by the MPHDATA audit.

3. AUDIT CHECKS

If the audit has been entered:

ZAP = Check if the audit has been entered due to a system interrupt.

For every tuple in the RLCPEST relation:

BAS_STAT = Check if the base state is in an invalid state.

BASEQUAL = Check if the first qualifier is in an invalid state for an in-service MTIB.

BASEQUAL2 = Check if the second qualifier is in an invalid state for an in-service MTIB.

GRBG = Check if the qualifier, error block, or state in an invalid state.

LOSTTUP = Check for missing rLCPEST tuple.

QUAL = Check if the first qualifier is in an invalid state for an out-of-service MTIB.

QUAL_2 = Check if the second qualifier is in an invalid state for an out-of-service MTIB.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The rLCPEST tuple is in-service (bas_stat = SMACT) but the first qualifier (qual_1) is not null (SMNULL), manual (SMAN), or automatic (SMAUTO).
Possible Error Effect:	The MTIB associated with this rLCPEST tuple may not be able to be removed from service or restored. Also this MTIB will not be allocated to perform metallic

	testing.
Corrective Action Taken:	The first qualifier field (qual_1) in the rCPEST tuple is corrected to null (SMSNUL), and the rIMPHDATA tuple associated with this rCPEST tuple is quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.
<i>Bad Data =</i>	The first qualifier field (qual_1) from the rCPEST tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump =</i>	rCPEST tuple found in error.

Table 2 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The rCPEST tuple is in-service (bas_stat = SMACT) but the second qualifier field (qual_2) is not null (SMSTNUL), camp-on (SMCM PON), or pre-out-of-service (SMPROOS).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be removed from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The second qualifier field (qual_2) in the rCPEST tuple is corrected to null (SMSTNUL) and the rIMPHDATA tuple associated with this rCPEST tuple is quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCPEST tuple found in error.
<i>Bad Data =</i>	The second qualifier field (qual_2) from the rCPEST tuple.
<i>Good Data =</i>	Null value of the second qualifier (SMSTNUL).
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump =</i>	rCPEST tuple found in error.

Table 3 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state (bas_stat) of the rCPEST tuple is in invalid state (not SMACT or SMOOS).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be removed from service or restored. Also, the MTIB may not be allocated to perform metallic testing.
Corrective Action Taken:	The base_stat field (bas_stat) in the rCPEST tuple is corrected to in-service (SMACT) and the rIMPHDATA tuple associated with this rCPEST tuple is quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCPEST tuple found in error.
<i>Bad Data =</i>	The base state field (bas_stat) from the rCPEST tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump =</i>	rCPEST tuple found in error.

Table 4 Error Code: GRGB

Error Code:	GRGB
Error Description:	The qualifier (qual != SMIA_MAN), error block (errorblk != 0), or state (state != SMALLLOW) fields in the rCPEST tuple have been mutilated. Some process has wild written into these fields.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rCPEST tuple will be recovered to an in-service state and the rIMPHDATA tuple associated with this rCPEST tuple will be quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rCPEST tuple does not exist in the data base but should. This may be caused by the logical key (unit_id) of the rCPEST tuple being corrupted.
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also, the MTIB may not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 6 Error Code: QUAL

Error Code:	QUAL
Error Description:	The rCPEST tuple is out-of-service (bas_stat = SMOOS) but the first qualifier (qual_1) is not manual (SMAN) or automatic (SMAUTO).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple is recovered to an in-service state and the rIMPHDATA tuple associated with this rCPEST tuple will be quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	The first qualifier field (qual_1) from the rCPEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 7 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The rCPEST tuple is out-of-service (bas_stat = SMOOS) but the second qualifier field (qual_2) is not out-of-service (SMOOS), removed (SMRMVD), diagnose (SMDGNS), routine exercise (SMRTN_EX), fault (SMFLT), trouble analysis (SMTBLA), or diagnostic exercise (SMDG_EX).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple is recovered to an in-service state and the rIMPHDATA tuple associated with this rCPEST tuple will be quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	The second qualifier field (qual_2) from the rCPEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after taking a system interrupt (such

	as, write protect, out-of-range read, etc.).
Possible Error Effect:	May cause further resets, audits, and/or asserts.
Corrective Action Taken:	Zero the rCPEST tuple which was causing the CPEST audit to take the system interrupt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

CPEST-B

Software Release: 5E14 and later

1. CENTRAL PROCESSOR EQUIPMENT STATUS TABLE (CPEST) AUDIT

The maintenance request administrator (MRA) program maintains the in-service and out-of-service statuses of all the hardware circuits in a switching module (SM) in the switch maintenance equipment status table (RLSMEST) except the metallic test interconnection buses (MTIBs). MRA maintains the statuses of the MTIBs in the central processor equipment status relation (RLCPEST) because the MTIBs are globally allocated in the communications module processor (CMP). The MTIBs connect metallic services units (MSUs) in all the SMs; the MSUs contain the circuitry for performing automatic line insulation tests and other metallic tests, and provide compensation for leakage when lines are being tested. The CPEST audit ensures the integrity of the RLCPEST relation. The CPEST audit is a part of the switch maintenance audits subgroup. It runs in the CMP in the routine and elevated segmented modes.

2. DATA STRUCTURES

The rLCPEST tuple keeps the status of the MTIBs in the CMP. As Figure 1 shows, the tuples of the CPEST relation are linked to tuples of the MSU path data relation (RLMPHDATA), which is audited by the MPHDATA audit.

rLCPEST

CIRCUIT NAME
BUSY_IDLE STATUS
QUALIFIER 1
QUALIFIER 2
LINKAGE TO rMPHDATA #

rMPHDATA

PID
MTIB NUMBER
STATUS
LINKAGE TO rLCPEST

#-----# = Nonstandard Linkage

Figure 1 (CPEST-1) Data Structures of the CPEST Audit

3. AUDIT CHECKS

If the audit has been entered:

ZAP = Check if the audit has been entered due to a system interrupt.

For every tuple in the RLCPEST relation:

BAS_STAT	= Check if the base state is in an invalid state.
BASEQUAL	= Check if the first qualifier is in an invalid state for an in-service MTIB.
BASEQUAL2	= Check if the second qualifier is in an invalid state for an in-service MTIB.
GRBG	= Check if the qualifier, error block, or state in an invalid state.
LOSTTUP	= Check for missing rCPEST tuple.
QUAL	= Check if the first qualifier is in an invalid state for an out-of-service MTIB.
QUAL_2	= Check if the second qualifier is in an invalid state for an out-of-service MTIB.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The rCPEST tuple is in-service (bas_stat = SMACT) but the first qualifier (qual_1) is not null (SMNULL), manual (SMAN), or automatic (SMAUTO).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be removed from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The first qualifier field (qual_1) in the rCPEST tuple is corrected to null (SMSNUL), and the rIMPHDATA tuple associated with this rCPEST tuple is quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.
<i>Bad Data =</i>	The first qualifier field (qual_1) from the rCPEST tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump =</i>	rCPEST tuple found in error.

Table 2 Error Code: BASEQUAL2

Error Code:	BASEQUAL2
Error Description:	The rCPEST tuple is in-service (bas_stat = SMACT) but the second qualifier field (qual_2) is not null (SMSTNUL), camp-on (SMCMPON), or pre-out-of-service (SMPROOS).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be removed from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The second qualifier field (qual_2) in the rCPEST tuple is corrected to null (SMSTNUL) and the rIMPHDATA tuple associated with this rCPEST tuple is quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCPEST tuple found in error.
<i>Bad Data =</i>	The second qualifier field (qual_2) from the rCPEST tuple.
<i>Good Data =</i>	Null value of the second qualifier (SMSTNUL).
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump =</i>	rCPEST tuple found in error.

Table 3 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The base state (bas_stat) of the rCPEST tuple is in invalid state (not SMACT or SMOOS).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also, the MTIB may not be allocated to perform metallic testing.
Corrective Action Taken:	The base_stat field (bas_stat) in the rCPEST tuple is corrected to in-service (SMACT) and the rIMPHDATA tuple associated with this rCPEST tuple is quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	The base state field (bas_stat) from the rCPEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 4 Error Code: GRGB

Error Code:	GRGB
Error Description:	The qualifier (qual != SMIA_MAN), error block (errorblk != 0), or state (state != SMALLOW) fields in the rCPEST tuple have been mutilated. Some process has wild written into these fields.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rCPEST tuple will be recovered to an in-service state and the rIMPHDATA tuple associated with this rCPEST tuple will be quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCPEST tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rCPEST tuple does not exist in the data base but should. This may be caused by the logical key (unit_id) of the rCPEST tuple being corrupted.
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also, the MTIB may not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	NA.

Table 6 Error Code: QUAL

Error Code:	QUAL
Error Description:	The rCPEST tuple is out-of-service (bas_stat = SMOOS) but the first qualifier (qual_1) is not manual (SMAN) or automatic (SMAUTO).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple is recovered to an in-service state and the rIMPHDATA tuple associated with this rCPEST tuple will be quarantined (quar = AUAUSET).

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	The first qualifier field (qual_1) from the rCPEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 7 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The rCPEST tuple is out-of-service (bas_stat = SMOOS) but the second qualifier field (qual_2) is not out-of-service (SMOOS), removed (SMRMVD), diagnose (SMDGNS), routine exercise (SMRTN_EX), fault (SMFLT), trouble analysis (SMTBLA), or diagnostic exercise (SMDG_EX).
Possible Error Effect:	The MTIB associated with this rCPEST tuple may not be able to be remove from service or restored. Also this MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple is recovered to an in-service state and the rIMPHDATA tuple associated with this rCPEST tuple will be quarantined (quar = AUAUSET).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	The second qualifier field (qual_2) from the rCPEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (unit_id) of the rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after taking a system interrupt (such as, write protect, out-of-range read, etc.).
Possible Error Effect:	May cause further resets, audits, and/or asserts.
Corrective Action Taken:	Zero the rCPEST tuple which was causing the CPEST audit to take the system interrupt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCPEST tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rCPEST tuple found in error.
<i>Dump</i> =	rCPEST tuple found in error.

CPLR

Software Release: 5E10 and later

1. COUPLER (CPLR) AUDIT

Couplers are special data structures that extend the basic owner-member (one-to-many) linkage scheme to many-to-many associations between ports and processes. The coupler (CPLR) audit verifies the linkages between couplers and ports and between couplers and processes. It also checks the linkage of idle couplers to the coupler idle list. If the linkage of a coupler is broken, the coupler may become inaccessible to the system, reducing the number of port-process associations that can be supported. This audit is part of the Feature Control audits subgroup. It operates in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structures checked by the CPLR audit are the coupler relation (RLCPLR), the port linkage area relation (RLPORTLA), and the process control block linkage area relation (RLPCBLA). Figures 1 and 2 illustrate the layout of the process control block linkage areas and port linkage areas and show the items checked by the audit.

rPCBLA

PROCESS NUMBER (procnum)
PORT {port}
LINK STATE (state)
CUSTOMER STATE (custat)
CALL WAITING IN PROGRESS {cw-inpro}
LINK TO rPHDB
LINK TO rCCBCOM
LINK TO rPORTLA OR rCPLR

Figure 1 (CPLR-1) Process Control Block Linkage Area

rPORTLA

PORT (port)
PORT STATUS (summary)
STATE (state)
PORT STATUS (pstatus)
PORT TYPE (ptype)
HUNT (hunt)
LINK TO rPHISTORY
LINK TO rPRTLST
LINK TO rPCBLA OR rCPLR

Figure 2 (CPLR-2) Port Linkage Area**3. AUDIT CHECKS**

(DRINASEK) once for each audit segment. The DRINASEK driver is discussed in the Audit Drivers section of this manual. The audit performs one semantic check: when an rCPLR tuple is on the idle list, it must not be linked to any ports.

4. ERROR HANDLING

If a coupler has broken linkage, it is returned to the list of idle couplers and its associated process is purged.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rCPLR tuple is on the idle list, but is still linked to one or more rPORTLA tuples.
Possible Error Effect:	Possible failures or asserts when call processing attempts to use either the rCPLR or rPORTLA tuples.
Corrective Action Taken:	The rCPLR tuple is recovered to an idle state and other linkages will be broken. Related audits will also be run so that the rPORTLA tuple will be recovered if necessary.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPLR tuple found in error.

<i>Bad Data</i> =	The error returned from the AUzerolchk() function.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rCPLR tuple (rCPLR.cplnum).
<i>Dump</i> =	The contents of the rCPLR tuple.

CPNBNCNT

Software Release: 5E10 and later

1. CALLING PARTY NUMBER/BILLING NUMBER COUNT (CPNBNCNT) AUDIT

Calling party number (CPN) is a number that identifies the originator of the call and is provided by the originating customer premises equipment (CPE) or by the originating end office. Billing number (BN) is a number provided by the originating end office and is used for billing identification.

The intra/inter switch CPN/BN optioning to the terminating user feature delivers the CPN or BN of the calling party to the called party of an integrated services digital network (ISDN) interface for circuit switched voice (CSV) and circuit switched data (CSD) calls. CPN/BN can be delivered on intra-switch, intra-LATA (local area transport access), and inter-LATA calls. Intra-LATA calls require common channel signaling 7 (CCS7) trunks or primary rate interfaces (PRIs) to deliver CPN and/or BN to the terminating switch. Inter-LATA calls require CCS7 trunks to deliver CPN and/or BN to the terminating switch.

This feature can be billed on a flat rate or a usage sensitive basis. If the feature for a given subscriber is on a usage sensitive basis, then a rICPNBNCNT tuple should exist for that subscriber.

The purpose of this audit is to verify that for every dynamic tuple in the RLCPNBNCNT relation, there also exists a static tuple in the RLIS_LINE relation with either the "bill_opt" or "dcpn_bill" field set to usage sensitive billing (USAGE_SENS).

2. DATA STRUCTURES

The RLCPNBNCNT dynamic relation is keyed by the lead switching module and logical termination index of the subscriber. This relation maintains the counts of CPNs and BNs delivered and/or not delivered to the called party. These counts are used by the automatic message accounting (AMA) to generate billing records.

The RLIS_LINE static relation is used by the CPNBNCNT audit to indicate if a given subscriber is billed for the CPN/BN feature on a usage sensitive basis.

3. AUDIT CHECKS

The CPNBNCNT audit invokes the DRSTATIC driver to step through the RLCPNBNCNT relation. For each tuple, the CPNBNCNT audit performs the following checks:

BILLFLG = Verify consistency between static data and dynamic data fields.

EXTRATUP = Check if this subscriber has usage sensitive billing.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	Using the lead switching module (lead_module) and the logical termination index (ltm_idx) fields of the dynamic rICPNBNCNT tuple one can access the static rLIS_LINE tuple. If the rLIS_LINE tuple indicates that the user does not have usage sensitive billing (bill_opt != USAGE_SENS or dcpn_bill !=

	USAGE_SENS) or the static rIIS_LINE tuple is not in the database then delete the extra dynamic rICPNBNCNT tuple from the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The dynamic rICPNBNCNT tuple will be deleted from the database using the database manager (DBM) primitives.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the buffer that contains the extra rICPNBNCNT tuple.
<i>Bad Data</i> =	The lead switching module (lead_module) field from the rICPNBNCNT tuple found in error.
<i>Good Data</i> =	DBM return code from trying to access the rIIS_LINE tuple.
<i>Logical Index</i> =	The logical termination index (ltm_idx) field from the rICPNBNCNT tuple found in error.
<i>Dump</i> =	The rIIS_LINE tuple.

Table 2 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	For a given valid rICPNBNCNT tuple, one of the dynamic data fields (vcpn_bill, dcpn_bill, or cgps_alw) does not match its' static data counterparts (bill_opt, dcpn_bill, or cgps_alw) respectively in the rIIS_LINE tuple.
Possible Error Effect:	AMA will get incorrect information when reading the rICPNBNCNT tuple data.
Corrective Action Taken:	The vcpn_bill, dcpn_bill, and cgps_alw fields in the rICPNBNCNT tuple will be updated with the corresponding static data from the rIIS_LINE tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the buffer that contains the rICPNBNCNT tuple in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The lead switching module (lead_module) field from the rICPNBNCNT tuple found in error.
<i>Logical Index</i> =	The logical termination index (ltm_idx) field from the rICPNBNCNT tuple found in error.
<i>Dump 1</i> =	Contents of the rICPNBNCNT tuple.
<i>Dump 2</i> =	Contents of the rIIS_LINE tuple.

CPPCBLA

Software Release: 5E10 and later

1. CENTRAL PROCESSOR PROCESS CONTROL BLOCK LINKAGE AREA (CPPCBLA) AUDIT

Call processing and terminal and switch maintenance are implemented by means of processes. Processes are associated with data blocks called process control block linkage areas. Call processing functions are implemented by terminal processes. One terminal process is associated with each terminal (line, trunk, and so forth) in a call. A terminal process process control block linkage area contains data that describes the process, links it to other entities (such as ports and time slots), and controls the call itself (such as call waiting or forwarding data). Incorrect data in a process control block linkage area can cause a variety of problems related to call processing. For example, broken linkage can cause processes to tie up terminals indefinitely when a call does not exist; or bad feature data can cause faulty processing of features (such as call forwarding and call waiting).

The switching module (SM) process control block linkage areas are audited by the process control block linkage area audit (PCBLA). The AM process control block linkage area audit (CPPCBLA) performs minimal checks on the AM process control block linkage areas. Since these linkage areas do not have one-to-one or owner-member linkages, no syntactic checks are made.

This audit is part of the Feature Control audits subgroup. It operates in the OKP environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structure checked by the CPPCBLA audit is the administrative module process control block linkage area relation (RLCPPCBLA). Figure 1 illustrates the layout of RLCPPCBLA and shows the items checked by the audit.

KEY (procno)
CUSTOMER STATE (custat)
.
.
.
FAR PROCESS ID (fpid)
FAR PROCESS ID 2 (fpid2)

Figure 1 (CPPCBLA-1) Central Processor Process Control Block Linkage Area

3. AUDIT CHECKS

The audit calls the linkage driver (DRLINK) once for each audit segment. The DRLINK uses an internal look-up table to determine that no linkages are associated with the RLCPPCBLA relation; therefore, no syntactic checks are made. The semantic check function performs the following check. Processes are either transient or stable depending on the content of the customer state (custat) item in the process control block linkage area of a process (for example, dialing is transient and talking is stable). The CPPCBLA audit checks the customer state item of the AM process control block linkage area. If it is transient and has not changed in over five minutes, the AM process

control block linkage area is considered to be in error.

4. ERROR HANDLING

If the AM process control block linkage area customer state of a transient process has not changed in over five minutes, the process is purged. Errors are reported by means of output messages.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	An rICPPCBLA tuple has an invalid value for the customer state.
Possible Error Effect:	The process would be purged the next time it receives a stimulus.
Corrective Action Taken:	The process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICPPCBLA tuple.
<i>Bad Data</i> =	Customer state.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rICPPCBLA tuple number.
<i>Dump</i> =	rICPPCBLA tuple.

Table 2 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rICPPCBLA tuple has been in a transient customer state for at least five minutes and has not changed state.
Possible Error Effect:	The port involved is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICPPCBLA tuple.
<i>Bad Data</i> =	Customer state.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rICPPCBLA number.
<i>Dump 1</i> =	rICPPCBLA tuple.
<i>Dump 2</i> =	Process control block.
<i>Dump 3</i> =	Program status.

Table 3 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The timestamp (used to find stuck processes) saved by the audit during a previous execution indicates a time later than the current time returned by the operating system.
Possible Error Effect:	Probably none - except that recovery of any possible stuck processes will be temporarily delayed for another check interval.
Corrective Action Taken:	The timestamp was reset to current time.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUold_time (timestamp variable).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	AUold_time.
<i>Dump 2</i> =	new_time (OSrtime).

12. CQ - CZ

CQB

Software Release: 5E10 and later

1. CIRCUIT QUEUING BLOCK (CQB) AUDIT

Normally a circuit is allocated to call processing and then returned to the idle list. However, circuits must be accessed for either routine or event-driven maintenance. If a call process owns a circuit, maintenance requests are queued until the call process is done. Each circuit queuing block is a representation of a maintenance request that is queued for a circuit. There are two types of requests: for a specific circuit from event-driven maintenance or for any given circuit of a specific class from routine maintenance. The circuit queuing block audit (CQB) checks the logical key and the linkages of the tuples in the circuit queuing block relation. Errors in the circuit queuing block could result in lost resources for call processing.

The CQB audit is part of the peripheral control audits subgroup. It runs at execution level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The CQB audit operates on the linkage areas of the circuit queuing block relation (RLCQB). The RLCQB relation has member linkages with the channel control block common area relation (RLCCBCOM), the circuit data block common area relation (RLCDBCOM), and the database head cells relation (RLDB_HEADC). Figure 1 illustrates these data structures.

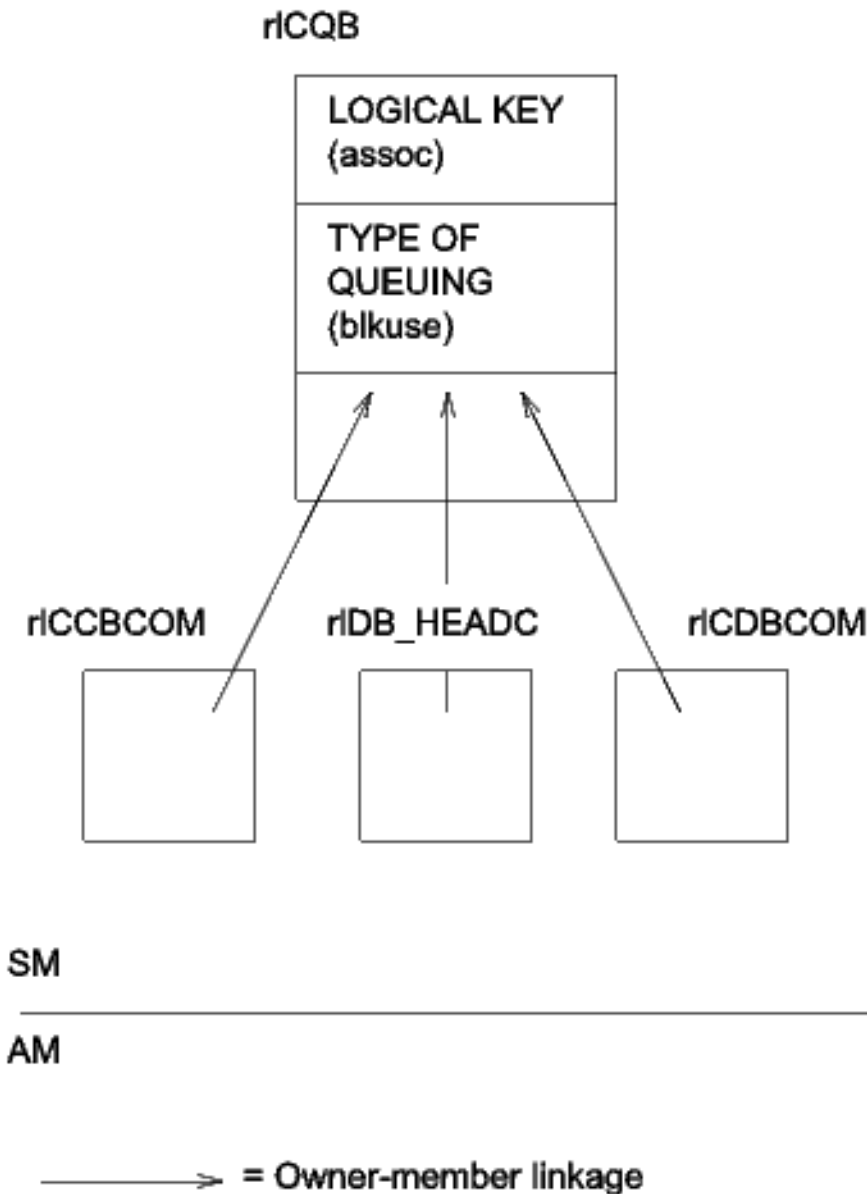


Figure 1 (CQB-1) rICQB Linkages

3. AUDIT CHECKS

The CQB audit calls the DRINASEK audit driver to make the standard audit checks. After the linkages are verified, the driver invokes the semantic check function to perform the following checks:

- (a) The audit checks the type of queuing (blkuse) field to ensure that it is valid.
- (b) The audit checks the blkuse field against the circuit queuing block linkages. The following conditions may exist.
 - An idle circuit queuing block should have a circuit queuing block idle link list head cell listed as its master owner; if not, the circuit queuing block cannot be used by call processing.
 - An idle circuit queuing block should have zero data in its slave linkage; if it does not, a bad linkage may

cause an assert.

- A circuit queuing block that is queuing for a circuit of a particular type should have a channel control block common area as its master owner and a channel control block common area head cell as its slave owner, and no other linkages; if it does not, the circuit queuing block cannot be used by call processing.
- A circuit queuing block that is queuing for a particular circuit should have a channel control block common area as its master owner and a valid linkage to a database head cell, and no other linkages; if it does not, the circuit queuing block cannot be used by call processing.

4. ERROR HANDLING

Any tuples that fail these checks are reinitialized and placed on the idle list. Any tuples that are used by an unrecognized circuit (blkuse) are reinitialized and placed on the idle list.

The CQB audit schedules the CCBCOM and CDBCOM audits as related audits when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CQB_LINK

Error Code:	CQB_LINK
Error Description:	The rICQB tuple has been found improperly linked according to the usage identifier (rICQB.blkuse). If the usage identifier (rICQB.blkuse) is set indicating that a user is queueing for any circuit by type (PCLSTCMP) then, the rICQB tuple should be: (1) linked (rICQB.cqccm) to an rICBCOM tuple associated with a valid process. (2) linked (rICQB.cqcdm) to an rICBCOM tuple that is used as a head cell for which the particular circuit type is being queued. (3) all other linkages (rICQB.cqptm and rICQB.cqhcm) should be zero. If the usage identifier (rICQB.blkuse) is set indicating that a user is queueing for a particular circuit (PCCKTCMP) then the rICQB tuple should be: (1) linked (rICQB.cqccm) to an rICBCOM tuple associated with a valid process. (2) linked (rICQB.cqcdm) to a particular rCDBCOM tuple representing that circuit. (3) all other linkages (rICQB.cqptm and rICQB.cqhcm) should be zero. If the usage identifier (rICQB.blkuse) is set indicating that a user is queueing for a TTF unit (PCTUNCMP) then the rICQB tuple should be: (1) linked (rICQB.cqccm) to an rICBCOM tuple associated with a valid

	process.
	(2) linked (rICQB.cqhcm) to an rIDB_HEADC tuple representing the head cell for a particular TTF unit.
	(3) all other linkages (rICQB.cqcdm and rICQB.cqptm) should be zero.
	If the usage identifier (rICQB.blkuse) is set indicating that a user is queueing for a time slot (PCPTSBCMP) then the rICQB tuple should be:
	(1) inked (rICQB.cqccm) to an rICBCOM tuple associated with a valid process.
	(2) linked (rICQB.cqptm) to an rIPTS tuple representing the time slot.
	(3) all other linkages (rICQB.cqcdm and rICQB.cqhcm) should be zero.
Possible Error Effect:	The rICQB is unavailable to call processing.
Corrective Action Taken:	The rICQB is placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICQB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	rICQB tuple's actual usage identifier (rICQB.blkuse).
<i>Logical Index =</i>	Logical key (rICQB.assoc) of the rICQB tuple.
<i>Dump =</i>	rICQB tuple.

Table 2 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A rICQB whose usage field indicates that it is does not have the rICQB idle link list head cell as its master owner.
Possible Error Effect:	The rICQB is in a limbo condition, unavailable to call processing.
Corrective Action Taken:	The audit places the rICQB on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICQB tuple.
<i>Bad Data =</i>	rICQB tuple's actual owner.
<i>Good Data =</i>	DBIDLECQB+DBSFCCBCOM.
<i>Logical Index =</i>	Logical key of the rICQB tuple.
<i>Dump =</i>	rICQB tuple.

Table 3 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rICQB has been found with an invalid usage identifier.
Possible Error Effect:	The state of the rICQB is unknown.
Corrective Action Taken:	The audit places the rICQB on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICQB tuple.
<i>Bad Data =</i>	rICQB tuple's actual usage identifier.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rICQB tuple.
<i>Dump =</i>	rICQB tuple.

Table 4 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An rICQB whose usage field indicates that it is idle has non-zero data in its slave linkage.
Possible Error Effect:	The bad linkage may cause an assert before the audit finds the error.
Corrective Action Taken:	The audit reinitializes the rICQB tuple.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICQB.
<i>Bad Data</i> =	Nonzero slave linkage found.
<i>Good Data</i> =	blkuse (block usage indicator) field of rICQB tuple.
<i>Logical Index</i> =	Logical key of rICQB.
<i>Dump</i> =	rICQB tuple.

CQDB

Software Release: 5E10 and later

1. CALL QUEUE DATA BLOCK (CQDB) AUDIT

Features of the Operator Services Position System (OSPS) include basic automatic call distribution (ACD) and call handling. These features contain the call processing software needed for an operator to log onto a basic services terminal (BST) and handle requests for operator services. The ACD device routes calls to available positions. Audits maintain the integrity of the data structures needed to perform these OSPS functions. The call queue data block audit ensures that the call queue, used to hold calls requesting operator assistance until an operator is available, is accurate. The CQDB audit is part of the OSPS AM basic automatic call distribution loadable package (OPAMBACD). It is part of the OSPS audits subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The CQDB audit uses three relations, the dynamic call queue data block relation (RLCQDB), the dynamic serving team queue data block relation (RLSTQDB), and the dynamic head cells for call type queues relation (RLHCCALLTQ). These data structures, and the linkages among them, are shown in Figure 1.

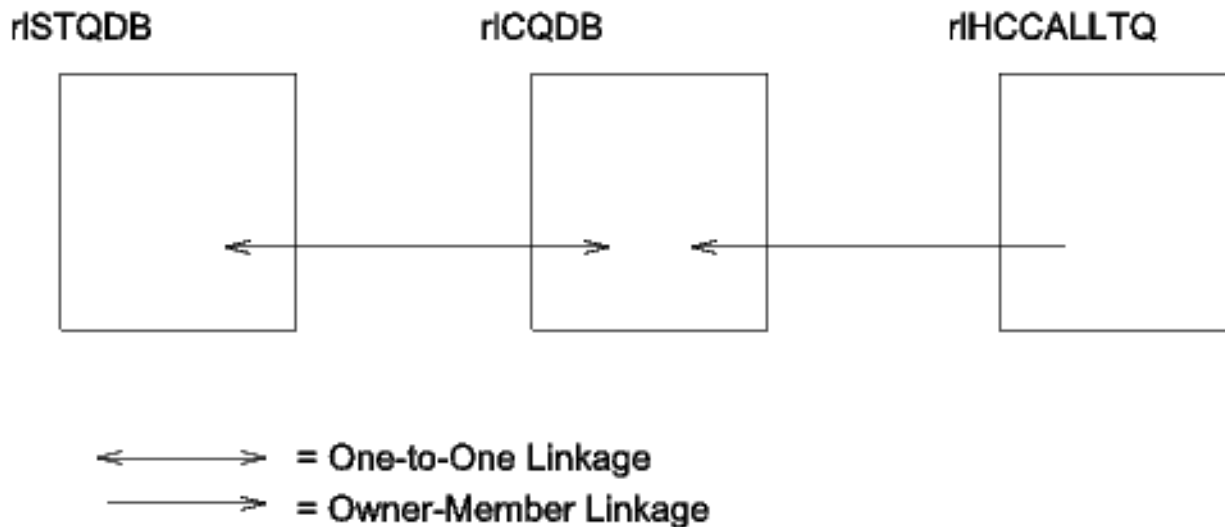


Figure 1 (CQDB-1) Data Structures of the CQDB Audit

3. AUDIT CHECKS

The audit invokes the indirect access sequential key driver (DRINASEK) to step through the tuples of RLCQDB and validate their linkages to RLSTQDB and RLHCCALLTQ.

4. ERROR HANDLING

Tuples of RLCQDB that are invalidly linked are returned to the idle list.

The CQDB audits schedules the HCCALLTQ and STQDB audits as related audits when it detects errors.

The audit uses only the error codes of the DRINASEK driver. It has no error codes of its own because it contains no semantic checks.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

CQPARM

Software Release: 5E11 and later

1. CALL QUEUE PARAMETERS (CQPARM) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. The call queue data is stored in the RLCQPARM (call queue parameters) relation. Data in this relation is audited by the CQPARM audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLCQPARM relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a static rICQPARM tuple that does not have a corresponding dynamic rIHC_CLLQ (head cell for call queues) tuple. The dump of the rICQPARM tuple distinguishes this LOSTTUP error from the other one produced by this audit.
Possible Error Effect:	This may prevent calls from being placed on a call queue and presented to the corresponding serving team represented by this rICQPARM tuple.
Corrective Action Taken:	The audit will insert the missing dynamic rIHC_CLLQ tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIHC_CLLQ tuple.
<i>Bad Data =</i>	The key of the rIHC_CLLQ tuple.
<i>Good Data =</i>	The key of the rICQPARM tuple.
<i>Logical Index =</i>	The logical key of the rICQPARM tuple.
<i>Dump =</i>	The rICQPARM tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found the dynamic idle rIHC_CLLQ (head cell for call queues) tuple to be missing. For this LOSTTUP error there are no tuples dumped as part of the error report. This is what distinguishes this error from the other LOSTTUP error produced by this audit.
Possible Error Effect:	This may prevent calls from being placed on any call queues.
Corrective Action Taken:	The audit will insert the missing dynamic idle rIHC_CLLQ tuple.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the rIHC_CLLQ tuple.
<i>Bad Data</i> =	The key of the rIHC_CLLQ tuple.
<i>Good Data</i> =	The key of the idle rIHC_CLLQ tuple.
<i>Logical Index</i> =	The logical key of the idle rIHC_CLLQ tuple.
<i>Dump</i> =	NA.

CR

Software Release: 5E10 and later

1. CALL RECORD (CR) AUDIT

The call record audit verifies data used by the administrative services (AS) subsystem. A call record is a fixed-size block of memory associated with each call being processed by the switching system. The Administrative Services subsystem uses information in the call record to generate billing entries transmitted to the Automatic Message Accounting and Recording Center (AMARC) and to increment traffic measurements transmitted to the engineering and administration data acquisition system (EADAS). Call records reside only in the switching modules and are used exclusively by administrative features. A call record is assigned when there is a request for service (line and trunk originations). It remains associated with the call throughout call processing. Call processing events and data are recorded in the call record until the call disconnects. The call record then returns to an idle state for reassignment. It may, however, remain assigned to a nonexistent call or contain irrelevant data; these circumstances may result in no billing for a processed call and invalid traffic measurements. The call record audit guards against these errors by identifying and releasing call records in the busy state that are not associated with a call.

The audit runs in the post-initialization and segmented modes in the operating system for distributed switching-switching module (OSDS-SM) environment.

2. DATA STRUCTURES

Call records are stored as elements in an array that only the Administrative Services subsystem accesses. There is one array per switching module and each array contains up to 512 call records. Figure 1 illustrates the elements and their attributes in the call record array (CRaddrtbl). Each element contains data associated with a single call. One attribute, the busy/idle status, indicates whether the call record is assigned to a call or is available for reassignment to a new call. Another attribute contains the process ID (PID), which identifies the call processing terminal process that owns the call record. The PID serves as a link between a busy call record and the call to which it is assigned.

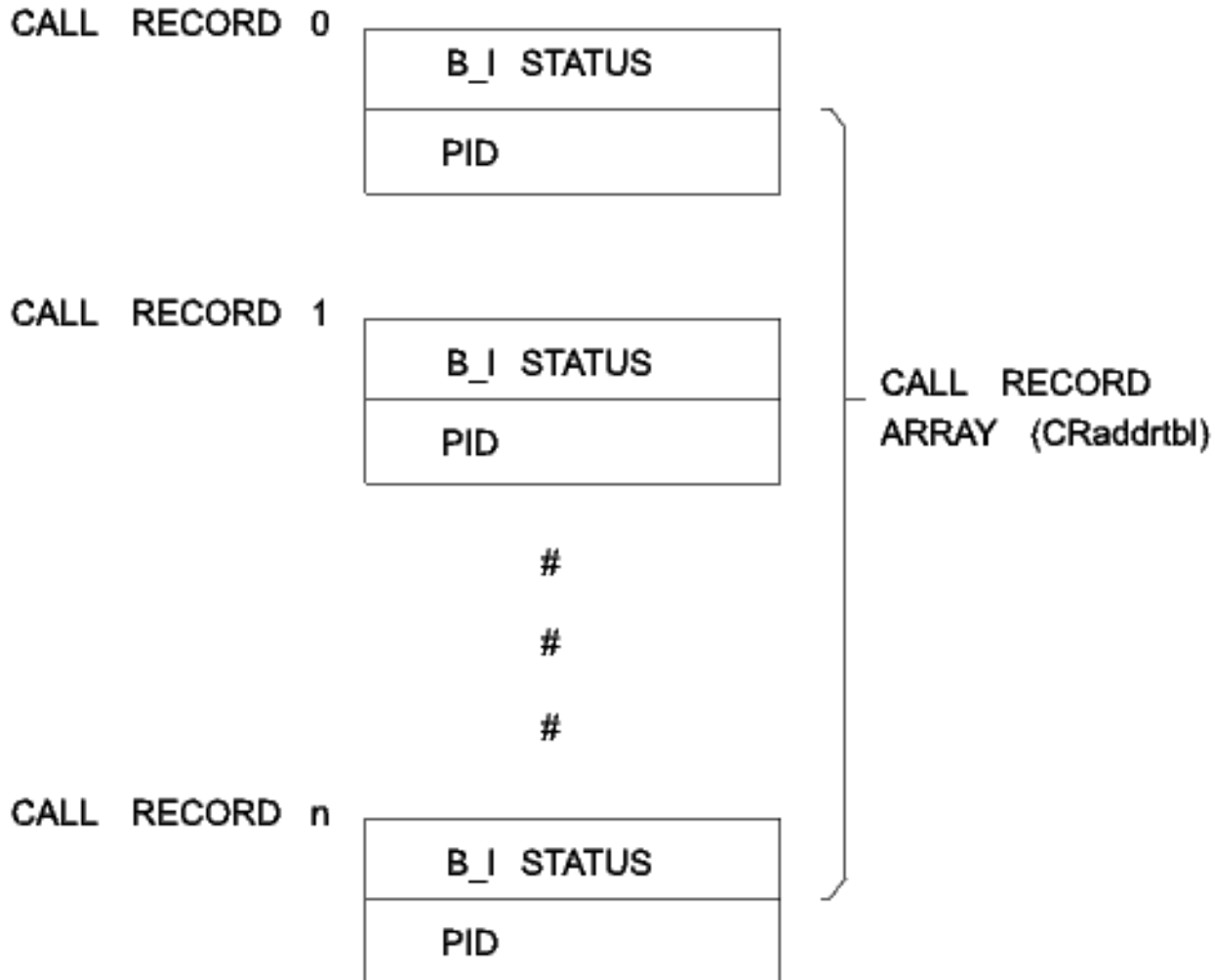


Figure 1 (CR-1) Audited Fields in the Call Record Array

3. AUDIT CHECKS

The call record audit runs in the initialization and segmented modes. The audit performs the same functions in both modes; it does not take segment breaks in the initialization mode. In the segmented mode, the audit sequences through approximately 20 call records per segment. It uses a local variable index to determine the call records to audit during a segment and a local variable counter to keep track of the number of call records audited. The audit performs consistency checks by examining the status and PID members of the call records. When a status member indicates that a call record is idle, the audit sequences to the next call record. When a status member is busy, the audit checks the PID field and verifies the existence of the process; it does this by checking, through OSDS-M, that a process control block has the PID of the process associated with the call record. If the process does not exist, the audit releases (sets to idle) the call record. For calls that are a call-forwarded leg, the audit checks the call forwarding system camp-on table (cfscptable) for an entry containing the ID of the call record being audited. If an entry does not exist, the audit releases the call record.

4. ERROR HANDLING

When the audit detects a call record with a busy status that is not associated with a process or a call forwarding leg without an entry in the cfscptable table, it zeroes all the elements of the call record and changes its status to idle.

The audit will dump the call record in several 196-byte dumps.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	A call record contains a process ID whose processor ID (CRECORD.crpId.pcrId) is zero. This indicates that there is no process associated with the CRECORD. However, the CRECORD status indicates that it is not idle.
Possible Error Effect:	The CRECORD may be lost. If enough CRECORDs were lost, calls might fail during set up.
Corrective Action Taken:	The call record is initialized and released.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the CRECORD (CRaddrtbl[CRECORD index]).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the call record.
<i>Dump</i> =	The call record.

Table 2 Error Code: BRLTLNK

Error Code:	BRLTLNK
Error Description:	An RLT call record has a bad link to a call record. A valid call record was discovered to have an RLT indx (rlt_cri) pointing to an RLT call record that contained a call record indx (cr_indx) that did not correctly point back to the call record.
Possible Error Effect:	The RLT call record is lost to the system.
Corrective Action Taken:	The RLT call record is initialized and released.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the RLT call record.
<i>Bad Data</i> =	cr_indx stored in the RLT call record.
<i>Good Data</i> =	rlt_cri stored in the standard call record.
<i>Logical Index</i> =	Index to the standard call record.
<i>Dump</i> =	NA.

Table 3 Error Code: NOCFSCOP

Error Code:	NOCFSCOP
Error Description:	A call record produced by call forwarding was found without corresponding entry in the cfscoptable[].
Possible Error Effect:	If this error occurs repeatedly, the supply of idle call records would become exhausted, resulting in calls not being billed.
Corrective Action Taken:	First, the ASINCOMPLETE primitive is called to report that the call associated with the call ID from the audit represents an incomplete call. Then, the ASRELCID primitive is called to release the call ID to the call record idle queue so that the call record associated with the call ID can be used for another call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the call record with no entry in the cfscoptable[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to call record with bad CFSCOP.
<i>Dump</i> =	Call record contents.

Table 4 Error Code: NONEPROC

Error Code:	NONEPROC

Error Description:	A call record was found to have an invalid process ID associated with it.
Possible Error Effect:	If this error occurs repeatedly, the supply of idle call records would become exhausted, resulting in calls not being billed.
Corrective Action Taken:	First, the ASINCOMPLETE primitive is called to report that the call associated with the call ID from the audit represents an incomplete call. Then, the ASRELCID primitive is called to release the call ID to the call record idle queue so that the call record associated with the call ID can be used for another call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the call record with the nonexistent process ID.
<i>Bad Data</i> =	Non-existent PID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to call record with bad PID.
<i>Dump</i> =	Call record contents.

Table 5 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	A call record contained an index to an rINS_IDX tuple (CRecord[call_id].ns_idx) which did not have a linkage back to the CRecord[] entry (rINS_IDX.crindx != CRecord[call_id].crdb_idx).
Possible Error Effect:	Data about associated call legs will not be available to users of this call record. This could result in incorrect call handling or billing.
Corrective Action Taken:	Report the call as incomplete and release the call record.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the call record (& CRecord[key]).
<i>Bad Data</i> =	Key of the rINS_IDX tuple (rINS_IDX.nsidkey).
<i>Good Data</i> =	Key to the rINS_IDX tuple in call record (CRecord[key].ns_idx).
<i>Logical Index</i> =	Call record index (CRecord[key].crindx).
<i>Dump 1</i> =	The rINS_IDX tuple.
<i>Dump 2</i> =	The call record.

Table 6 Error Code: ZAP

Error Code:	ZAP
Error Description:	Addressing error occurred while the CR audit was executing.
Possible Error Effect:	Audit was not able to finish.
Corrective Action Taken:	Zap the cursor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of AUCursor.
<i>Bad Data</i> =	AUCursor.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

CRDATA

Software Release: 5E11 and later

1. CLUSTER ROUTE DATA (CRDATA) AUDIT

CRDATA audit steps through every entry in the CCcrdata (cluster routing data) and CCcr_rdb (cluster routing RDB) and check the cluster field, adjacent point code (APCI), rdb_count (number of related RDBs) and representative RDB by using the data in the RLS7RDB (signaling 7 routing data block) relation.

2. DATA STRUCTURES

For details on the fields contained in the RLS7RDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRA_ENTRY

Error Code:	EXTRA_ENTRY
Error Description:	The error is one of the following: (1) A new representative routing data block (RDB) for a cluster-route in CCcrdata[] is needed because the current representative RDB (CCcr_rdb[]) is not a valid RDB. A new representative RDB is not found, so the entry in CCcrdata[] is not needed. (2) A new representative RDB for a cluster-route in CCcrdata[] is needed because the index of the current representative RDB (CCcr_rdb[]) does not equal the index of the current CCcrdata[] entry. A new representative RDB is not found so the entry in CCcrdata[] is not needed.
Possible Error Effect:	In both cases, incorrect routing data for a destination point code (DPC) may be used to try to route to a DPC.
Corrective Action Taken:	In all cases, the audit sets CCcrdata[].cluster, CCcrdata[].apci and CCcrdata[].rdb_count equal to 0 and CCcr_rdb[] equal to CCCRDELETED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1) or (2) indicates error description.
<i>Bad Data =</i>	CCcrdata[].cluster.
<i>Good Data =</i>	CCcrdata[].apci.
<i>Logical Index =</i>	Incorrect representative RDB from CCcr_rdb[].
<i>Dump =</i>	Contents of CCcrdata[] entry.

Table 2 Error Code: INDEX

Error Code:	INDEX

Error Description:	The cluster-route representative RDBs index into CCcr_rdb[] should equal the index of the current CCcrdata[], but does not.
Possible Error Effect:	DPCs may have an incorrect accessibility statuses and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	The audit finds a valid RDB to represent the cluster-route and sets CCcr_rdb[] equal to the new RDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCcr_rdb[index].
<i>Bad Data</i> =	Index into CCcr_rdb[].
<i>Good Data</i> =	Index into CCcrdata[].
<i>Logical Index</i> =	rlS7RDB.rdb_id of the representative RDB.
<i>Dump</i> =	Contents of CCcrdata[index].

Table 3 Error Code: RDB_BAD

Error Code:	RDB_BAD
Error Description:	CCcrdata[] has the cluster and route information for the representative RDB for a DPC. This RDB exists in relation RLS7RDB. The audit attempts to find a tuple in relation RLDPC_STAT whose key (rlDPC_STAT.dpci) equals the DPC index (DPCI) (rlS7RDB.rdb_id /CCRDB_DPC) for the RDB, but does not find a tuple.
Possible Error Effect:	DPC may have an incorrect accessibility statuses, and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	Another audit inserts the missing rlDPC_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCcrdata[index].cluster.
<i>Bad Data</i> =	rlS7RDB.rdb_id of the representative RDB.
<i>Good Data</i> =	CCcrdata[index].apci.
<i>Logical Index</i> =	Index.
<i>Dump</i> =	NA.

Table 4 Error Code: READ

Error Code:	READ
Error Description:	A representative RDB assigned to CCcr_rdb[] for a cluster-route in CCcrdata[] does not have a matching tuple in relation RLS7RDB whose key (rlS7RDB.rdb_id) equals CCcr_rdb[].
Possible Error Effect:	DPCs may have an incorrect accessibility statuses and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	Other audits correct the RDB in CCcr_rdb[] and relation RLS7RDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCcrdata[index].cluster.
<i>Bad Data</i> =	CCcr_rdb[index].
<i>Good Data</i> =	CCcrdata[index].apci.
<i>Logical Index</i> =	Index.
<i>Dump</i> =	NA.

Table 5 Error Code: RDB_MISSING

Error Code:	RDB_MISSING
Error Description:	CCcr_rdb[] has the representative RDB for a cluster-route in CCcrdata[]. The audit attempts to find a tuple in relation RLS7RDB whose key (rlS7RDB.rdb_id) equals CCcr_rdb[], but does not find one.
Possible Error Effect:	DPCs may have an incorrect accessibility statuses and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	The audit finds an existing RDB to represent the cluster-route and sets CCcr_rdb[] equal to the new RDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCcrdata[index].cluster.
<i>Bad Data</i> =	Representative RDB from CCcr_rdb[index].
<i>Good Data</i> =	CCcrdata[index].apci.

<i>Logical Index</i> =	Index.
<i>Dump</i> =	NA.

Table 6 Error Code: SEARCH_FAILED

Error Code:	SEARCH_FAILED
Error Description:	CCcr_rdb[] has the representative RDB for a cluster-route in CCcrdata[]. The audit attempts to find the representative RDB's index into CCcr_rdb[], but can not find the index.
Possible Error Effect:	DPCs may have an incorrect accessibility statuses and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	Another audit corrects the index into CCcr_rdb[] for the representative RDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster of the representative RDB.
<i>Bad Data</i> =	rlS7RDB.rdb_id of the representative RDB.
<i>Good Data</i> =	APCI of the representative RDB.
<i>Logical Index</i> =	Index of current CCcrdata[] entry.
<i>Dump</i> =	NA.

Table 7 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect, out of range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Neither CCcrdata[] nor CCcr_rdb[] is verified for this iteration of the audit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

CRECORD-A

Software Release: 5E10 only

1. CALL RECORD (CRECORD) AUDIT

The call record audit verifies data used to provide customer billing. Each call being processed by the switching system has an associated call record. Information in the call record is used to generate billing entries transmitted to the Automatic Message Accounting and Recording Center (AMARC) and to increment traffic measurements transmitted to the Engineering and Administration Data Acquisition System (EADAS). Call records reside only in the switching modules and are used exclusively by administrative features. A call record is assigned when there is a request for service (line and trunk originations). It remains associated with the call throughout call processing. Call processing events and data are recorded in the call record until the call disconnects. The call record then returns to an idle state for reassignment. Call records which remain assigned to a nonexistent calls contain irrelevant data; these circumstances may result in no billing or erroneous billing for a processed call and invalid traffic measurements. The CRECORD audit guards against these errors by identifying and releasing Call Records in the busy state that are not associated with a call.

The CRECORD audit runs in the post-initialization and segmented modes in all switching modules (SMs).

2. DATA STRUCTURES

The CRECORD audit checks rICRECORD tuples to determine if the tuples are validly busy. The audit makes these checks based on call type, the particular RLDB_HEADC list a tuple is linked to, and the state of various other semantic data such as Process ID (PID). When a Call record is allocated, one or more of the following fields are used:

<i>cfscop_idx</i>	= For forwarded calls, contains the key of an associated rICFSCOP tuple.
<i>crlink</i>	= Contains the owner-member linkage to the RLDB_HEADC busy or idle list head cell (DBCRCBUSY or DBCRECIDLE).
<i>crnum</i>	= Call record number (tuple key). If the SM contains the OSPA loadable package, this value will correspond to the key of an associated rICRECORD (OSPA call record) tuple.
<i>crpid</i>	= The process ID (PID) of the call associated with the Call Record.
<i>custcallcode</i>	= Contains information on whether this is a forwarded call (ASCF_FWDLEG, ASRMT_FDLEG, or ASBLDA_FDLEG).
<i>ns_idx</i>	= Contains the index of a rINS_IDX tuple if one is
<i>rlt_cri</i>	= Contains the index of an release link trunk (RLT) record.
<i>sfg_call</i>	= Bit to indicate whether the call is associated with a simulated facility group (SFG) member.

The rICRECORD tuple and related data structures referenced by the CRECORD audit are shown in Figures 1, 2, and 3.

3. AUDIT CHECKS

For each tuple in the RLCRECORD relation, the CRECORD audit makes the following checks: If the CRECORD audit gets entered:

BUSYIDLE = Check that the rICRECORD tuple is linked to the RLDB_HEADC busy or idle list head cell.

If the rICRECORD tuple is associated with a forwarded call leg:

- FWDRGSLOT = Check that the associated rICFSCOP tuple points to the same PID as the rICRECORD tuple.
- NOCFSCOP = Check that there is a corresponding rICFSCOP tuple for call record.
- NONEPROC = Check that the call record is associated with an active (live) call.
- OUTOFRANGE = Check that the SM marked in the call record is valid.

If the call is associated with an SFG member:

- SFG_CRECORD = Check that there is an associated rISFGCOR tuple for this call record.

If there is a RLT record associated with the call:

- BRLTLNK = Check that the RLT entry is properly linked to this call record.

If there is an associated rINS_IDX tuple:

- PT2PT = Check that the linkage in RLNS_IDX points back to RLCRECORD.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BRLTLNK

Error Code:	BRLTLNK
Error Description:	An RLT call record has a bad link to a call record. If rICRECORD.rlt_cri is marked in range (< CRLTMAX AND > CRLTNUL) then CRrltcr[rICRECORD.rlt_cri].cr_indx should equal rICRECORD.cnum (key).
Possible Error Effect:	The RLT call record is lost to the system.
Corrective Action Taken:	The RLT call record is initialized and released, and the rICRECORD.rlt_cri field is corrected to CRLTNUL (idle).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICRECORD tuple.
<i>Bad Data =</i>	RLT call record index (rICRECORD.rlt_cri).
<i>Good Data =</i>	CRLTNUL
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump =</i>	The contents of the CRrltcr[] array indexed by rICRECORD.rlt_cri.

Table 2 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	A rICRECORD tuple was found unlinked to both the busy and idle list RLDB_HEADC tuples. All rICRECORD tuples must be linked to one of these head cells (rICRECORD.crlink.st_ownk == DBCRECIDLE or DBCRECBUSY).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is linked to the RLDB_HEADC idle list head cell.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rICRECORD tuple.
<i>Bad Data</i> =	The RLCRECORD owner linkage's logical value (rICRECORD.st_ownk - DBSFDB HEA)
<i>Good Data</i> =	The busy list head cell for RLDB_HEADC (DBCRECIDLE).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 3 Error Code: FWDRGSLOT

Error Code:	FWDRGSLOT
Error Description:	A rICFSCOP tuple linked to a rICRECORD tuple does not have the correct PID marked, or the state of the associated rICFSCOP tuple is marked inactive. All rICFSCOP tuples linked to rICRECORD tuples must be associated with the same process, except for queuing and vectoring terminal processes associated with ACD/BRCS (rICFSCOP.bpid == rICRECORD.crpId except when rICFSCOP.bfeatid == FCACD_ERTE). In addition, the state of the rICFSCOP tuple must not be marked inactive (rICFSCOP.state != CFINACTIVE).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is linked to the RLDB_HEADC idle list head cell. The rICFSCOP tuple is recovered by the CFSCOP audit which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICRECORD tuple.
<i>Bad Data</i> =	The state of the rICFSCOP tuple (rICFSCOP.state)
<i>Good Data</i> =	The SM number in the PID in the rICFSCOP tuple (rICFSCOP.bcid.crpId).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the rICFSCOP tuple.

Table 4 Error Code: NOCFSCOP

Error Code:	NOCFSCOP
Error Description:	A call record produced by call forwarding was found without corresponding tuple in the RLCFSCOP relation. For forwarded calls, the rICRECORD.cfscop_idx field should contain the key of a valid, busy, rICFSCOP tuple.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.cfscop_idx field.
<i>Good Data</i> =	The custom call code (rICRECORD.custcallcode) in the rICRECORD tuple.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 5 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	A call record was found for which there was no corresponding call process. This means the call record is being used for measurements and billing operations, but is not associated with an active call. Every active call record (rICRECORD.crlink.st_ownk owner linkage marked DBCRECBUSY + DBSFDB_HEA) must have an active (live) call process ID (PID marked in rICRECORD.crpId).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.

	NOTE: If there is an active call associated with the call record, but the process marked in the Call Record was corrupt, the call is left up, provided all other call data structures are in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.crpId field (process ID found in the call record).
<i>Good Data</i> =	If the process control block (PCB) for the process marked in the rICRECORD tuple has not been reallocated for another call (where rICRECORD.crpId.procno = rIOSPCB.pcbkey AND rIOSPCB.uniqueness = rICRECORD.crpId.uniq + 1 AND rIOSPCB.state = OSEMPY), then the program ID for the process which allocated the call record will be dumped. Otherwise, 0 will be dumped.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crn timer).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the PCB associated with the rICRECORD tuple. If the <i>Bad Data</i> is marked 0, then the PCB has been reallocated since the call associated with the stuck call record ended. For that case, the data contained in this dump will not be valid for investigating the occurrence of this audit error.

Table 6 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A call record was found for which there was no corresponding call process, or for which the existence of a corresponding call process cannot be verified. Specifically, the SM marked in the rICRECORD tuple where the call process resides is not a valid SM. The SM where the process resides should be greater than or equal to the first SM equipped in the office, but less than or equal to the last SM equipped, such that: rICRECORD.crpId.pcrId ≥ GL1STSM AND rICRECORD.crpId.pcrId ≤ GLLASTSM. This error indicates the call record is being used for measurements and billing operations, but is not associated with an active call, or cannot be properly released when the call ends.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state. NOTE: If there is an active call associated with the call record, but the process marked in the Call Record was corrupt, the call is left up, provided all other call data structures are in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.crpId.pcrId field (the SM where the process resides).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crn timer).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 7 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	A call record contained an index to an rINS_IDX tuple (rICRECORD.ns_idx != 0), but the rINS_IDX tuple did not have a linkage back to the call record tuple (rICRECORD.ns_idx != rINS_IDX.nsidkey OR rICRECORD.crn timer (key) != rINS_IDX.crdId idx).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The call record is recovered to an idle state. Any corrupt data in the RLNS_IDX relation is recovered by the NSIDX audit, which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	Key of the rINS_IDX tuple found in the call record (rICRECORD.ns_idx).

<i>Good Data</i> =	Linkage to the call record found in the associated rINS_IDX tuple (rINS_IDX.crd_idx).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the rINS_IDX tuple.

Table 8 Error Code: SFG_CRECORD

Error Code:	SFG_CRECORD
Error Description:	The call record is marked as being associated with an SFG, (rICRECORD.sfg_call == DBYES) but no corresponding rISFGCOR tuple exists. All call records associated with non-6 Port SFG calls must have corresponding rISFGCOR tuples where rICRECORD.cnum == rISFGCOR.cnum.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The SFG call indicator bit is cleared (marked to DBNO).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	SFG call status bit (rICRECORD.sfg_call).
<i>Good Data</i> =	Correct bit status (DBNO).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

CRECORD-B

Software Release: 5E11 - 5E13

1. CALL RECORD (CRECORD) AUDIT

The call record audit verifies data used to provide customer billing. Each call being processed by the switching system has an associated call record. Information in the call record is used to generate billing entries transmitted to the Automatic Message Accounting and Recording Center (AMARC) and to increment traffic measurements transmitted to the Engineering and Administration Data Acquisition System (EADAS). Call records reside only in the switching modules and are used exclusively by administrative features. A call record is assigned when there is a request for service (line and trunk originations). It remains associated with the call throughout call processing. Call processing events and data are recorded in the call record until the call disconnects. The call record then returns to an idle state for reassignment. Call records which remain assigned to a nonexistent calls contain irrelevant data; these circumstances may result in no billing or erroneous billing for a processed call and invalid traffic measurements. The CRECORD audit guards against these errors by identifying and releasing Call Records in the busy state that are not associated with a call.

The CRECORD audit runs in the post-initialization and segmented modes in all switching modules (SMs).

2. DATA STRUCTURES

The CRECORD audit checks rICRECORD tuples to determine if the tuples are validly busy. The audit makes these checks based on call type, the particular RLDB_HEADC list a tuple is linked to, and the state of various other semantic data such as process ID (PID). When a call record is allocated, one or more of the following fields are used:

<i>cfscop_idx</i>	= For forwarded calls, contains the key of an associated rICFSCOP tuple.
<i>crlink</i>	= Contains the owner-member linkage to the RLDB_HEADC busy or idle list head cell (DBCRECBUSY or DBCRECIDLE).
<i>crnum</i>	= Call record number (tuple key). If the SM contains the OSPS loadable package, this value will correspond to the key of an associated rICRECORD (OSPS call record) tuple.
<i>crpid</i>	= The process ID (PID) of the call associated with the call record.
<i>custcallcode</i>	= Contains information on whether this is a forwarded call (ASCF_FWDLEG, ASRMT_FDLEG, or ASBLDA_FDLEG).
<i>ns_idx</i>	= Contains the index of a rINS_IDX tuple if one is
<i>rlt_cri</i>	= Contains the index of a release link trunk (RLT) record.
<i>sfg_call</i>	= Bit to indicate whether the call is associated with a simulated facility group (SFG) member.

The rICRECORD tuple and related data structures referenced by the CRECORD audit are shown in Figures 1, 2, and 3.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	A rICRECORD (call record) tuple has a one2one linkage to a tuple in the RLNS_BILL relation, however, the tuple does not point back, or the one-to-one linkage in RLCRECORD is corrupted. If the one2one linkage to a tuple in the RLNS_BILL relation is in use (rICRECORD.crlink.st_link != 0), the linkage should point to a rINS_BILL tuple that points back to this rICRECORD tuple (nsidx mem.st link - DBSFCRECORD == rICRECORD.cnum).
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of call record resources.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICRECORD tuple.
<i>Bad Data =</i>	The linkage in RLNS_BILL (rINS_BILL.nsidx mem.st link).
<i>Good Data =</i>	The expected key calculated from the call record linkage to RLNS_BILL (rICRECORD.crlink.st_link - DBSFNS_BILL
<i>Logical Index =</i>	The logical key (cnum) of the rICRECORD tuple.
<i>Dump =</i>	The contents of the rICRECORD tuple.

Table 2 Error Code: BRLTLNK

Error Code:	BRLTLNK
Error Description:	An RLT call record has a bad link to a call record. If rICRECORD.rlt_cri is marked in range (< CRLTMAX AND > CRLTNUL) then CRrltcr[rICRECORD.rlt_cri].cr_idx should equal rICRECORD.cnum (key).
Possible Error Effect:	The RLT call record is lost to the system.
Corrective Action Taken:	The RLT call record is initialized and released, and the rICRECORD.rlt_cri field is corrected to CRLTNUL (idle).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICRECORD tuple.
<i>Bad Data =</i>	RLT call record index (rICRECORD.rlt_cri).
<i>Good Data =</i>	CRLTNUL
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump =</i>	The contents of the CRrltcr[] array indexed by rICRECORD.rlt_cri.

Table 3 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	A rICRECORD tuple was found unlinked to both the busy and idle list RLDB_HEADC tuples. All rICRECORD tuples must be linked to one of these head cells (rICRECORD.crlink.st_ownk == DBCRECIDLE or DBCRECBUSY).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is linked to the RLDB_HEADC idle list head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICRECORD tuple.
<i>Bad Data =</i>	The RLCRECORD owner linkage's logical value (rICRECORD.st_ownk - DBSFDB_HEA)
<i>Good Data =</i>	The busy list head cell for RLDB_HEADC (DBCRECIDLE).
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump =</i>	Contents of the rICRECORD tuple.

Table 4 Error Code: FWDRGSLOT

Error Code:	FWDRGSLOT
Error Description:	A rICFSCOP tuple linked to a rICRECORD tuple does not have the correct PID marked, or the state of the associated rICFSCOP tuple is marked inactive. All rICFSCOP tuples linked to rICRECORD tuples must be associated with the same process, except for queuing and vectoring terminal processes associated with ACD/BRCS (rICFSCOP.bpid == rICRECORD.crpId except when rICFSCOP.bfeatId == FCACD_RERTE). In addition, the state of the rICFSCOP tuple must not be marked inactive (rICFSCOP.state != CFINACTIVE).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is linked to the RLDB_HEADC idle list head cell. The rICFSCOP tuple is recovered by the CFSCOP audit which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICRECORD tuple.
<i>Bad Data =</i>	The state of the rICFSCOP tuple (rICFSCOP.state)
<i>Good Data =</i>	The SM number in the PID in the rICFSCOP tuple (rICFSCOP.bcid.crpId).
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1 =</i>	Contents of the rICRECORD tuple.
<i>Dump 2 =</i>	Contents of the rICFSCOP tuple.

Table 5 Error Code: NOCFSCOP

Error Code:	NOCFSCOP
Error Description:	A call record produced by call forwarding was found without corresponding tuple in the RLCFSCOP relation. For forwarded calls, the rICRECORD.cfscop_idx field should contain the key of a valid, busy, rICFSCOP tuple.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICRECORD tuple.
<i>Bad Data =</i>	The rICRECORD.cfscop_idx field.
<i>Good Data =</i>	The custom call code (rICRECORD.custcallcode) in the rICRECORD tuple.
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump =</i>	Contents of the rICRECORD tuple.

Table 6 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	A call record was found for which there was no corresponding call process. This means the call record is being used for measurements and billing operations, but is not associated with an active call. Every active call record (rICRECORD.crlink.st_ownk owner linkage marked DBCRECBUSY + DBSFDB_HEA) must have an active (live) call process ID (PID marked in rICRECORD.crpId).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state. NOTE: If there is an active call associated with the call record, but the process marked in the call record was corrupt, the call is left up, provided all other call data structures are in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICRECORD tuple.
<i>Bad Data =</i>	The rICRECORD.crpId field (process ID found in the call record).
<i>Good Data =</i>	If the process control block (PCB) for the process marked in the rICRECORD tuple has not been reallocated for another call (where rICRECORD.crpId.procno = rIOSPCB.pcbkey AND rIOSPCB.uniqueness = rICRECORD.crpId.uni + 1

	AND rIOSPCB.state = OSEMPY), then the program ID for the process which allocated the call record will be dumped. Otherwise, 0 will be dumped.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the PCB associated with the rICRECORD tuple. If the <i>Bad Data</i> is marked 0, then the PCB has been reallocated since the call associated with the stuck call record ended. For that case, the data contained in this dump will not be valid for investigating the occurrence of this audit error.

Table 7 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	For an idle call record (rICRECORD.crlink.st_ownc - DBSFDB_HEADC == DBCRECIDLE), the one-to-one linkage to RLNS_BILL is not null (DALNKNUL). This linkage should not be in use for idle rICRECORD tuples.
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of call record resources.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The contents of the one-to-one linkage with RLNS_BILL.
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	The call record number (logical key, rICRECORD.cnum) of the rICRECORD tuple.
<i>Dump</i> =	The contents of the rICRECORD tuple found in error.

Table 8 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A call record was found for which there was no corresponding call process, or for which the existence of a corresponding call process cannot be verified. Specifically, the SM marked in the rICRECORD tuple where the call process resides is not a valid SM. The SM where the process resides should be greater than or equal to the first SM equipped in the office, but less than or equal to the last SM equipped, such that: rICRECORD.crpri.pcri ≥ GL1STSM AND rICRECORD.crpri.pcri ≤ GLLASTSM. This error indicates the call record is being used for measurements and billing operations, but is not associated with an active call, or cannot be properly released when the call ends.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state. NOTE: If there is an active call associated with the call record, but the process marked in the call record was corrupt, the call is left up, provided all other call data structures are in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.crpri.pcri field (the SM where the process resides).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 9 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	A call record contained an index to an rINS_IDX tuple (rICRECORD.ns_idx != 0), but the rINS_IDX tuple did not have a linkage back to the call record tuple (rICRECORD.ns_idx != rINS_IDX.nsidkey OR rICRECORD.cnum (key) != rINS_IDX.crd_idx).

Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The call record is recovered to an idle state. Any corrupt data in the RLNS_IDX relation is recovered by the NSIDX audit, which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICRECORD tuple.
<i>Bad Data =</i>	Key of the rLNS_IDX tuple found in the call record (rICRECORD.ns_idx).
<i>Good Data =</i>	Linkage to the call record found in the associated rLNS_IDX tuple (rLNS_IDX.crd_idx).
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1 =</i>	Contents of the rICRECORD tuple.
<i>Dump 2 =</i>	Contents of the rLNS_IDX tuple.

Table 10 Error Code: SFG_CRECORD

Error Code:	SFG_CRECORD
Error Description:	The call record is marked as being associated with an SFG, (rICRECORD.sfg_call == DBYES) but no corresponding rISFGCOR tuple exists. All call records associated with non-6 Port SFG calls must have corresponding rISFGCOR tuples where rICRECORD.cnum == rISFGCOR.cnum.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The SFG call indicator bit is cleared (marked to DBNO).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rICRECORD tuple.
<i>Bad Data =</i>	SFG call status bit (rICRECORD.sfg_call).
<i>Good Data =</i>	Correct bit status (DBNO).
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump =</i>	Contents of the rICRECORD tuple.

Table 11 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The call record is marked as being an AUTOPLEX [®] mobile phone services (AMPS) call record owned by the AMPS system process (rICRECORD.crpidx.pcrd == CCAPSMID) but there is no longer an owning AMPS process. If the <i>Good Data</i> is 1 or 2, there is a problem getting the associated rIGPAXDB tuple; when the <i>Good Data</i> is 3, none of the call ID's in the rIGPAXDB tuples match this tuple; when the <i>Good Data</i> is 4, this fix was newly installed and is removing the old tuples associated with the system process.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	When <i>Good Data</i> is 1, the key of the rIGPAXDB tuple (rIGPAXDB.gpaxdb). When <i>Good Data</i> is 2, the client tag (rIGPAXDB.client_tag). When <i>Good Data</i> is 3, the call record index from far1 in rIGPAXDB. When <i>Good Data</i> is 4, key of rICRECORD tuple (rICRECORD.cnum).
<i>Bad Data =</i>	When <i>Good Data</i> is 1, the key of the first rIGPAXDB tuple (rIAPXCTBL.gpaxidx1). When <i>Good Data</i> is 2, the key of the second rIGPAXDB tuple (rIAPXCTBL.gpaxidx2). When <i>Good Data</i> is 3, the call record index from far2 in rIGPAXDB. When <i>Good Data</i> is 4, the contents of the pid (rICRECORD.crpidx).
<i>Good Data =</i>	1, 2, 3 or 4.
<i>Logical Index =</i>	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1 =</i>	Contents of the rICRECORD tuple.
<i>Dump 2 =</i>	When <i>Good Data</i> is 1, contents of the rIAPXCTBL tuple. When <i>Good Data</i> is 2

	or 3, contents of the rGPAXDB tuple. When <i>Good Data</i> is 4, NA.
<i>Dump 3</i> =	When <i>Good Data</i> is 1 or 4, NA. When <i>Good Data</i> is 2 or 3, contents of the rAPXCTBL tuple.

Table 12 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The call record is marked as being an AMPS call record owned by the AMPS system process (rICRECORD.crpId.pcrId == CCAPSMID) but there is no longer an owning AMPS process as there is a problem getting the associated rAPXCTBL tuple.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The key of the rAPXCTBL tuple.
<i>Bad Data</i> =	The contents of the pid (rICRECORD.crpId.procno << 16 pcrId << 8 uniq).
<i>Good Data</i> =	The keys to the two rGPAXDB tuples which must both exist (rAPXCTBL.gpaxIdx1 << 8 rAPXCTBL.gpaxIdx2).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crnNum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the rAPXCTBL tuple.

CRECORD-C

Software Release: 5E14 and later

1. CALL RECORD (CRECORD) AUDIT

The call record audit verifies data used to provide customer billing. Each call being processed by the switching system has an associated call record. Information in the call record is used to generate billing entries transmitted to the Automatic Message Accounting and Recording Center (AMARC) and to increment traffic measurements transmitted to the Engineering and Administration Data Acquisition System (EADAS). Call records reside only in the switching modules and are used exclusively by administrative features. A call record is assigned when there is a request for service (line and trunk originations). It remains associated with the call throughout call processing. Call processing events and data are recorded in the call record until the call disconnects. The call record then returns to an idle state for reassignment. Call records which remain assigned to a nonexistent calls contain irrelevant data; these circumstances may result in no billing or erroneous billing for a processed call and invalid traffic measurements. The CRECORD audit guards against these errors by identifying and releasing Call Records in the busy state that are not associated with a call.

The CRECORD audit runs in the post-initialization and segmented modes in all switching modules (SMs).

2. DATA STRUCTURES

The CRECORD audit checks rICRECORD tuples to determine if the tuples are validly busy. The audit makes these checks based on call type, the particular RLDB_HEADC list a tuple is linked to, and the state of various other semantic data such as process ID (PID). When a call record is allocated, one or more of the following fields are used:

<i>cfscop_idx</i>	= For forwarded calls, contains the key of an associated rICFSCOP tuple.
<i>crlink</i>	= Contains the owner-member linkage to the RLDB_HEADC busy or idle list head cell (DBCRECBUSY or DBCRECIDLE).
<i>crnum</i>	= Call record number (tuple key). If the SM contains the OSPA loadable package, this value will correspond to the key of an associated rICRECORD (OSPA call record) tuple.
<i>crpid</i>	= The process ID (PID) of the call associated with the call record.
<i>custcallcode</i>	= Contains information on whether this is a forwarded call (ASCF_FWDLEG, ASRMT_FDLEG, or ASBLDA_FDLEG).
<i>ns_idx</i>	= Contains the index of a rINS_IDX tuple if one is
<i>rlt_cri</i>	= Contains the index of a release link trunk (RLT) record.
<i>sfg_call</i>	= Bit to indicate whether the call is associated with a simulated facility group (SFG) member.

The rICRECORD tuple and related data structures referenced by the CRECORD audit are shown in Figures 1, 2, and 3.

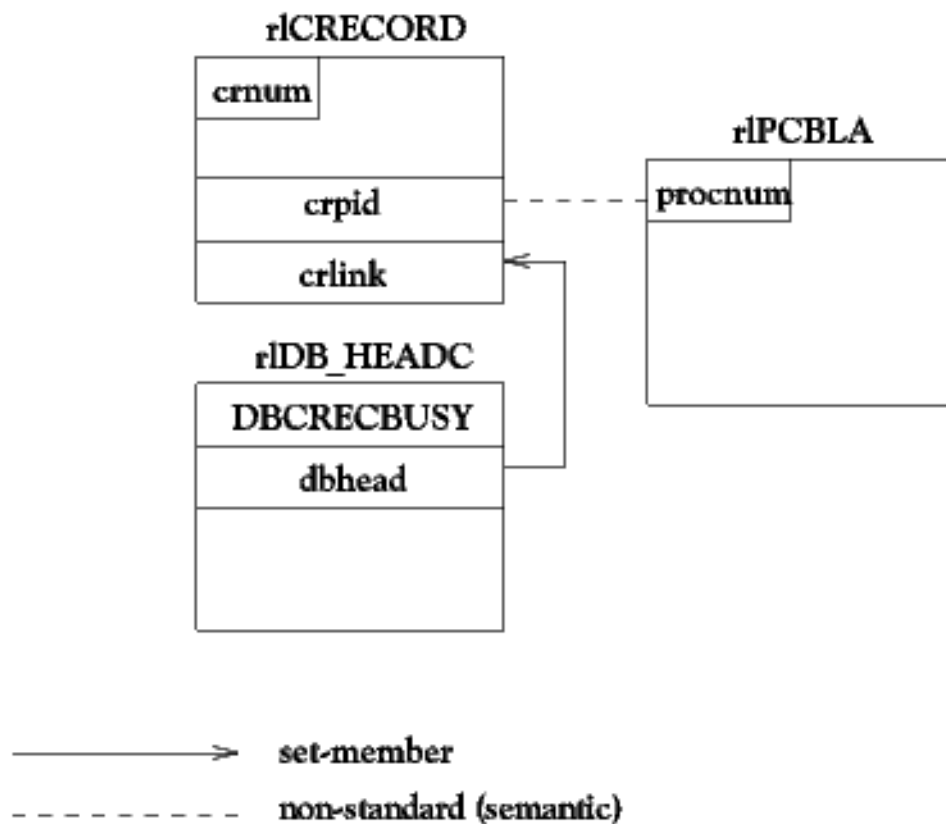


Figure 1 (CRECORD-1) CRECORD Basic Call Scenario

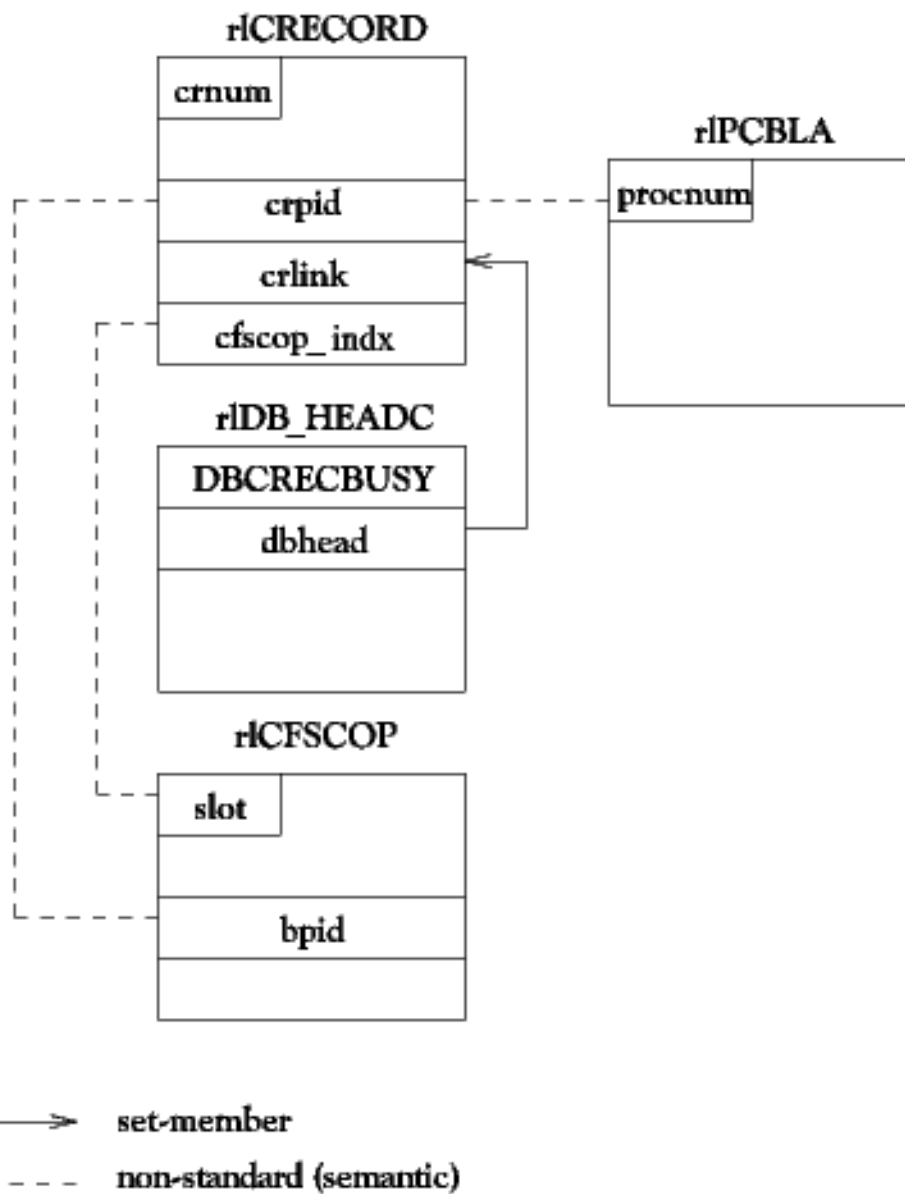


Figure 2 (CRECORD-2) CRECORD Call Forwarding Scenario

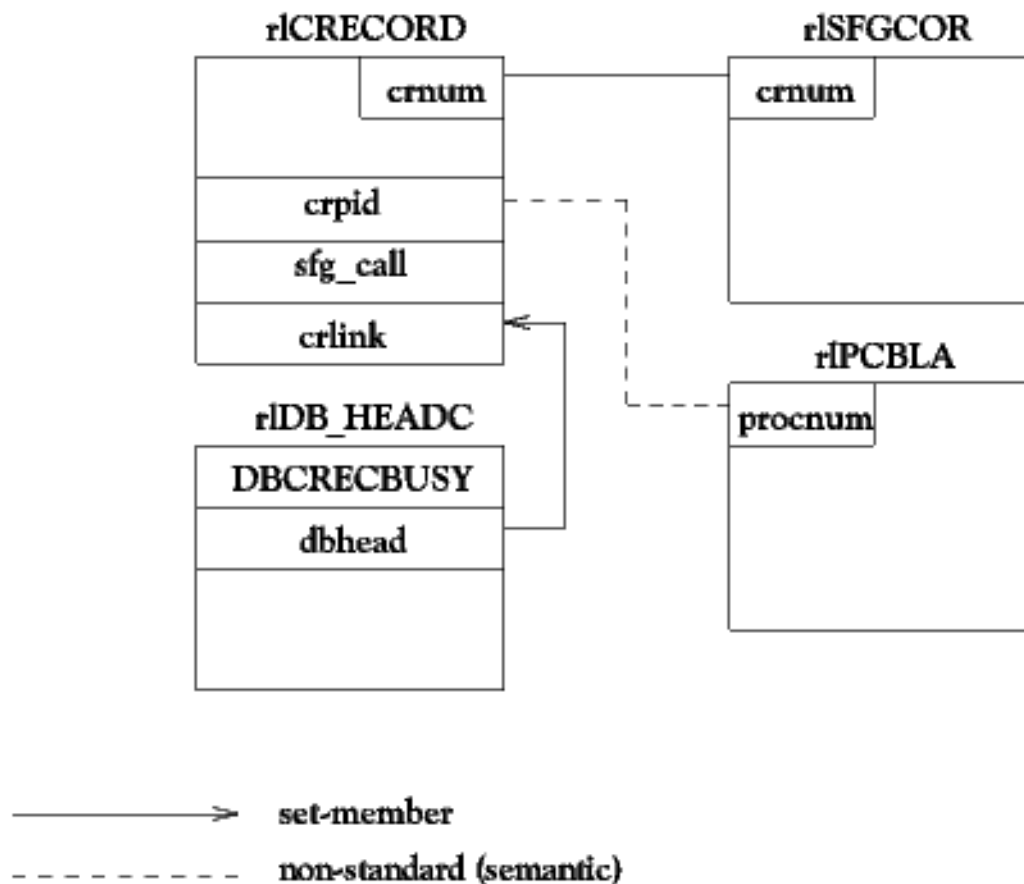


Figure 3 (CRECORD-3) CRECORD SFG Call Scenario

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	A rICRECORD (call record) tuple has a one2one linkage to a tuple in the RLNS_BILL relation, however, the tuple does not point back, or the one-to-one linkage in RLCRECORD is corrupted. If the one2one linkage to a tuple in the RLNS_BILL relation is in use (rICRECORD.crlink.st_link != 0), the linkage should point to a rINS_BILL tuple that points back to this rICRECORD tuple (nsidx mem.st link - DBSFCRECORD == rICRECORD.crnum).
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of call record resources.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The linkage in RLNS_BILL (rlNS_BILL.nsidx_mem.st_link).
<i>Good Data</i> =	The expected key calculated from the call record linkage to RLNS_BILL (rICRECORD.crlink.st_link - DBSFNS_BILL).
<i>Logical Index</i> =	The logical key (crnum) of the rICRECORD tuple.
<i>Dump</i> =	The contents of the rICRECORD tuple.

Table 2 Error Code: BRLTLNK

Error Code:	BRLTLNK
Error Description:	An RLT call record has a bad link to a call record. If rICRECORD.rlt_cri is marked in range (< CRLTMAX AND > CRLTNUL) then CRrltr[rICRECORD.rlt_cri].cr_indx should equal rICRECORD.crnum (key).
Possible Error Effect:	The RLT call record is lost to the system.
Corrective Action Taken:	The RLT call record is initialized and released, and the rICRECORD.rlt_cri field is corrected to CRLTNUL (idle).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICRECORD tuple.
<i>Bad Data</i> =	RLT call record index (rICRECORD.rlt_cri).
<i>Good Data</i> =	CRLTNUL
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crnum).
<i>Dump</i> =	The contents of the CRrltr[] array indexed by rICRECORD.rlt_cri.

Table 3 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	A rICRECORD tuple was found unlinked to both the busy and idle list RLDB_HEADC tuples. All rICRECORD tuples must be linked to one of these head cells (rICRECORD.crlink.st_ownk == DBCRECIDLE or DBCRECBUSY).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is linked to the RLDB_HEADC idle list head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICRECORD tuple.
<i>Bad Data</i> =	The RLCRECORD owner linkage's logical value (rICRECORD.st_ownk - DBSFDB_HEA)
<i>Good Data</i> =	The busy list head cell for RLDB_HEADC (DBCRECIDLE).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 4 Error Code: FWDRGSLOT

Error Code:	FWDRGSLOT
Error Description:	A rICFSCOP tuple linked to a rICRECORD tuple does not have the correct PID marked, or the state of the associated rICFSCOP tuple is marked inactive. All rICFSCOP tuples linked to rICRECORD tuples must be associated with the same process, except for queuing and vectoring terminal processes associated with ACD/BRCS (rICFSCOP.bpid == rICRECORD.crpuid except when rICFSCOP.bfeatid == FCACD_RERTE). In addition, the state of the rICFSCOP tuple must not be marked inactive (rICFSCOP.state != CFINACTIVE).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is linked to the RLDB_HEADC idle list head cell. The rICFSCOP tuple is recovered by the CFSCOP audit which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICRECORD tuple.
<i>Bad Data</i> =	The state of the rICFSCOP tuple (rICFSCOP.state)
<i>Good Data</i> =	The SM number in the PID in the rICFSCOP tuple (rICFSCOP.bcid.crpuid).

<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the rICFSCOP tuple.

Table 5 Error Code: NOCFSCOP

Error Code:	NOCFSCOP
Error Description:	A call record produced by call forwarding was found without corresponding tuple in the RLCFSCOP relation. For forwarded calls, the rICRECORD.cfscop_indx field should contain the key of a valid, busy, rICFSCOP tuple.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.cfscop_indx field.
<i>Good Data</i> =	The custom call code (rICRECORD.custcallcode) in the rICRECORD tuple.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 6 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	A call record was found for which there was no corresponding call process. This means the call record is being used for measurements and billing operations, but is not associated with an active call. Every active call record (rICRECORD.crlink.st_ownk owner linkage marked DBCRECBUSY + DBSFDB_HEA) must have an active (live) call process ID (PID marked in rICRECORD.crpId).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state. NOTE: If there is an active call associated with the call record, but the process marked in the call record was corrupt, the call is left up, provided all other call data structures are in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.crpId field (process ID found in the call record).
<i>Good Data</i> =	If the process control block (PCB) for the process marked in the rICRECORD tuple has not been reallocated for another call (where rICRECORD.crpId.procno = rIOSPCB.pcbkey AND rIOSPCB.uniqueness = rICRECORD.crpId.uniq + 1 AND rIOSPCB.state = OSEMPY), then the program ID for the process which allocated the call record will be dumped. Otherwise, 0 will be dumped.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the PCB associated with the rICRECORD tuple. If the <i>Bad Data</i> is marked 0, then the PCB has been reallocated since the call associated with the stuck call record ended. For that case, the data contained in this dump will not be valid for investigating the occurrence of this audit error.

Table 7 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	For an idle call record (rICRECORD.crlink.st_ownk - DBSFDB_HEADC == DBCRECIDLE), one of the non-idle list linkages is not null (DALNKNUL). This linkage should not be in use for idle rICRECORD tuples.
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of call record resources.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	Contents of the bad linkage field.
<i>Good Data</i> =	DALNKNUL.
<i>Logical Index</i> =	The call record number (logical key, rICRECORD.crnnum) of the rICRECORD tuple.
<i>Dump</i> =	The contents of the rICRECORD tuple found in error.

Table 8 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A call record was found for which there was no corresponding call process, or for which the existence of a corresponding call process cannot be verified. Specifically, the SM marked in the rICRECORD tuple where the call process resides is not a valid SM. The SM where the process resides should be greater than or equal to the first SM equipped in the office, but less than or equal to the last SM equipped, such that: rICRECORD.crpId.pcrId ≥ GL1STSM AND rICRECORD.crpId.pcrId ≤ GLLASTSM. This error indicates the call record is being used for measurements and billing operations, but is not associated with an active call, or cannot be properly released when the call ends.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered to an idle state. NOTE: If there is an active call associated with the call record, but the process marked in the call record was corrupt, the call is left up, provided all other call data structures are in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	The rICRECORD.crpId.pcrId field (the SM where the process resides).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crnnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 9 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	A call record contained an index to an rINS_IDX tuple (rICRECORD.ns_idx != 0), but the rINS_IDX tuple did not have a linkage back to the call record tuple (rICRECORD.ns_idx != rINS_IDX.nsidxkey OR rICRECORD.crnnum (key) != rINS_IDX.crdB_idx).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The call record is recovered to an idle state. Any corrupt data in the RLNS_IDX relation is recovered by the NSIDX audit, which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	Key of the rINS_IDX tuple found in the call record (rICRECORD.ns_idx).
<i>Good Data</i> =	Linkage to the call record found in the associated rINS_IDX tuple (rINS_IDX.crdB_idx).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.crnnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	Contents of the rINS_IDX tuple.

Table 10 Error Code: SFG_CRECORD

Error Code:	SFG_CRECORD
Error Description:	The call record is marked as being associated with an SFG, (rICRECORD.sfg_call == DBYES) but no corresponding rSFGCOR tuple exists.

	All call records associated with non-6 Port SFG calls must have corresponding rISFGCOR tuples where rICRECORD.cnum == rISFGCOR.cnum.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The SFG call indicator bit is cleared (marked to DBNO).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICRECORD tuple.
<i>Bad Data</i> =	SFG call status bit (rICRECORD.sfg_call).
<i>Good Data</i> =	Correct bit status (DBNO).
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump</i> =	Contents of the rICRECORD tuple.

Table 11 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The call record is marked as being an <i>AUTOPLEX</i> [®] mobile phone services (AMPS) call record owned by the AMPS system process (rICRECORD.crpId.pcrId == CCAPSMID) but there is no longer an owning AMPS process. If the <i>Good Data</i> is 1 or 2, there is a problem getting the associated rIGPAXDB tuple; when the <i>Good Data</i> is 3, none of the call ID's in the rIGPAXDB tuples match this tuple; when the <i>Good Data</i> is 4, this fix was newly installed and is removing the old tuples associated with the system process.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	When <i>Good Data</i> is 1, the key of the rIGPAXDB tuple (rIGPAXDB.gpaxdb). When <i>Good Data</i> is 2, the client tag (rIGPAXDB.client_tag). When <i>Good Data</i> is 3, the call record index from far1 in rIGPAXDB. When <i>Good Data</i> is 4, key of rICRECORD tuple (rICRECORD.cnum).
<i>Bad Data</i> =	When <i>Good Data</i> is 1, the key of the first rIGPAXDB tuple (rIAPXCTBL.gpaxidx1). When <i>Good Data</i> is 2, the key of the second rIGPAXDB tuple (rIAPXCTBL.gpaxidx2). When <i>Good Data</i> is 3, the call record index from far2 in rIGPAXDB. When <i>Good Data</i> is 4, the contents of the pid (rICRECORD.crpId).
<i>Good Data</i> =	1, 2, 3 or 4.
<i>Logical Index</i> =	Key of the rICRECORD tuple (rICRECORD.cnum).
<i>Dump 1</i> =	Contents of the rICRECORD tuple.
<i>Dump 2</i> =	When <i>Good Data</i> is 1, contents of the rIAPXCTBL tuple. When <i>Good Data</i> is 2 or 3, contents of the rIGPAXDB tuple. When <i>Good Data</i> is 4, NA.
<i>Dump 3</i> =	When <i>Good Data</i> is 1 or 4, NA. When <i>Good Data</i> is 2 or 3, contents of the rIAPXCTBL tuple.

Table 12 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The call record is marked as being an AMPS call record owned by the AMPS system process (rICRECORD.crpId.pcrId == CCAPSMID) but there is no longer an owning AMPS process as there is a problem getting the associated rIAPXCTBL tuple.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rICRECORD tuple is recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The key of the rIAPXCTBL tuple.
<i>Bad Data</i> =	The contents of the pid (rICRECORD.crpId.procno << 16 pcrId << 8 uniq).
<i>Good Data</i> =	The keys to the two rIGPAXDB tuples which must both exist

	(rlAPXCTBL.gpaxidx1 << 8 rlAPXCTBL.gpaxidx2).
<i>Logical Index</i> =	Key of the rlCRECORD tuple (rlCRECORD.cnum).
<i>Dump 1</i> =	Contents of the rlCRECORD tuple.
<i>Dump 2</i> =	Contents of the rlAPXCTBL tuple.

CRV

Software Release: 5E10 and later

1. CALL REFERENCE VALUE (CRV) AUDIT

The CRV audit is part of the ISDNBASE package and runs routinely in the operating system for distributed switching-switching module (OSDS-SM) environment as part of routine port conditioning (RPC). It is executed under the control of an integrated services port administrator (ISPA) terminal process and is not part of the routine audits system process cycle. In addition to routine execution under RPC, the CRV audit also runs as a part of the switching module (SM) full and selective init, protocol handler (PH) selective init (using ISPA selective restore job), digital subscriber line (DSL) restoration, customer premises equipment (CPE) refresh, and the PH audit recovery interface using RPC. The CRV audit has interfaces with the terminal maintenance (TM) for CRV data collection and error logging, and with PS for CRV recovery. The CRV audit is the complement of the call activity check [part of the D-channel application linkage block (DALB) audit]. The call activity check validates CRVs that the switch knows about in an attempt to synchronize CPEs with lost information. The CRV audit uses the information acquired from the CPE [full initializing terminals (FIT) only], and compares it with the data in the switch.

2. DATA STRUCTURES

The CRV audit is given a terminal endpoint identifier (TEI) linkage area (TEILA) tuple a list of active CRV's as input, and it looks at the owning D-channel port block (DPB) and the list of DALBs owned by this DPB.

3. AUDIT CHECKS

The TEILA tuple, which represents a terminal on a DSL, is passed to the CRV audit. A list of active CRVs, which the terminal thinks is active and is collected by a maintenance information message (MIM), is passed to the audit. The CRV audit looks through the list of DALBs linked to this DSL (from the owning DPB tuple) and constructs a list of CRVs that the switch knows about. The audit compares the results in an effort to find any lost CRVs from the perspective of the switch.

4. ERROR HANDLING

For any CRV that the switch does not know about, a message is sent to the terminal to release the CRV, and the error is logged [not printed on the 3B read-only printer (ROP)]. The user is able to dump the error logs if desired using the Master Control Center (MCC) command.

```
EXEC:ALE, DN = nnnmmmm, per, print,
```

where nnnmmmm = the directory number of the terminal.

Additional Comments

The CRV audit is not requestable via the normal audit input message; it is only inhibitable and allowable, separately, in addition to inhibiting RPC.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

CTINFO

Software Release: 5E10 and later

1. CALL TYPE INFORMATION (CTINFO) AUDIT

Features of the operator services position system (OSPS) include basic automatic call distribution (ACD) and call handling. These features contain the call processing software needed for an operator to log onto a basic services terminal (BST) and handle requests for operator services. The ACD device routes calls to available positions. Audits are needed to maintain the integrity of the data structures needed to perform these OSPS functions. The call type information audit (CTINFO) verifies data used by the head cells for call type queues audit (HCCALLTQ). The CTINFO audit is part of the OSPS AM basic automatic call distribution loadable package (OPAMBACD). It is part of the OSPS audits subgroup. The audit runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The CTINFO audit uses two relations, the static call type information relation (RLCTINFO) and the dynamic head cells for call type queues relation (RLHCCALLTQ).

3. AUDIT CHECKS

The CTINFO audit calls the static driver (DRSTATIC) to perform a static-to-dynamic existence check on the tuples of RLCTINFO and RLHCCALLTQ. It ensures that all tuples in RLCTINFO have a corresponding tuple in the dynamic relation. If there is no corresponding tuple, the audit creates one.

4. ERROR HANDLING

If the audit does not find a corresponding tuple in RLHCCALLTQ, it creates one.

The CTINFO audit schedules the HCCALLTQ audit as a related audit when it detects errors.

The audit produces these error codes.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A rIHCCALLTQ does not exist when it should.
Possible Error Effect:	This condition results in lost resources.
Corrective Action Taken:	The rIHCCALLTQ tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIHCCALLTQ.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIHCCALLTQ.
<i>Dump =</i>	NA.

CUMEM

Software Release: 5E10 and later

1. CONTROL UNIT MEMORY COMPARISON (CUMEM) AUDIT

The control unit memory comparison (CUMEM) audit compares the active and standby main store units in duplex processors. It runs by manual request only and reports a completed termination status after all of the main store has been audited.

NOTE: The CUMEM audit is a manually invoked audit that can take more than 90 minutes to complete in a 16-megabyte main store configuration. It should be requested only when the processor configuration is ACTIVE/STANDBY. Excessive error counts during the running of this audit (in correcting mode) may result in a hardware reconfiguration with one of the control units (CUs) being marked out-of-service (OOS).

The CUMEM audit compares the content of the on-line main store with that of the standby unit. It considers the on-line main store to be correct and reports most errors against the standby when mismatches occur. If the on-line cache is the source of the problem, the errors are reported against the on-line cache unit. If the on-line or off-line control unit (CU) exceeds its error threshold and is taken out of service, the CUMEM audit aborts.

1.1 Requesting the CUMEM Audit

To request the memory comparison (CUMEM) audit, use the following input message.

Enter:

```
AUD:CUMEM=1 [:a|:b|:a,b];
```

where:

a = Correcting or non correcting. Valid values are:

CORR = Specifies that an audit should correct errors (default).

NCG = Specifies that an audit does not correct errors.

b = Type of report. Valid values are:

DETL = Generates a raw data report for each error found plus an audit completion message (default). (A limited number of raw data reports are generated.)

SUM = Generates only an audit completion message when the audit terminates. Raw data reports are not generated for any errors that are found.

To interpret system responses to the AUD:CUMEM input message, refer to the APP:AUD Appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD:CUMEM Output Messages and Error Reports

The AUD:CUMEM output message reports the termination status of the CUMEM audit and/or any errors found while executing the audit. For a detailed explanation of the AUD CUMEM output message, see the Output Messages manual (235-600-750).

Table 1 lists the CUMEM audit error numbers and their associated categories.

Table 1 CUMEM AUDIT ERROR NUMBERS

Category	Error Number	Meaning
0,3	2	Bad data in ECD.
2,3	4	ECD access function failed.
0	6	Maintenance channel failure.
0	8	Reading off-line memory using update circuit failed.
1	10	Kconfig function call failed.
0	12	CU was found to have the wrong major state in ECD.
0	14	Bad online system status register (SSR) value.
0	16	Bad offline SSR value.
0	18	Bad online hardware status register (HSR) value.
0	20	Bad offline HSR value.
0	22	Cache hit/miss ratio is unacceptable.
0	24	PMS database access failure.
0	26	Audit control library call failed.
0	28	Internal PCPAUD failure.
0	30	Off-line store access failure.
1	32	Memory audit internal error.
0	34	Off-line memory mismatch.
0	36	On-line cache memory mismatch.
0	38	CUMEM run when off-line was not standby.

For error numbers 30, 34, and 36, the DATA1 field contains the main store address at which the test failed. For all other error messages, the DATA1 field contains the assert number. Table 2 lists the assert numbers that may be reported and explains their meanings.

Table 2 CUMEM ERROR REPORTS

Assert Number	Meaning
0x2000	Memory audit did not complete its entire memory block comparison
0x2001	Kconfig function failure rpterr
0x2002	Kconfig function failure rpterr
0x2003	Kconfig function failure fltdrvr
0x2004	CUMEM called when off-line CU was not standby

Under some error conditions, the CUMEM audit may produce additional output messages. These reports have the following format:

```
REPT PCPAUD FAULT TYPE a ASSERT NO b
```

where:

a = Fault type.

b = Assert number.

For a detailed explanation of the REPT:PCPAUD output message and a list of fault types and assert numbers, refer to the Output Messages manual (235-600-750).

1.3 Responding to AUD:CUMEM Error Reports

For the listed error numbers, take the following actions. For any other error numbers, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

Error Number:	Action:
4	Run the ECD 1 audit.
6	If the off-line CU was not taken out of service, remove it. Diagnose it, repair it if necessary, and restore it. Then switch CUs and remove, diagnose, repair, and restore the other CU.
8	The off-line CU should have been removed automatically. Diagnose it, repair it if necessary, and restore it. Then switch CUs and remove, diagnose, repair, and restore the other CU.
24	Run the PMS 1 audit.

30	The off-line CU should have been removed automatically. Diagnose it, repair it if necessary, and restore it. Then switch CUs and remove, diagnose, repair, and restore the other CU.
34	If the off-line CU was not taken out of service, remove it. Diagnose it, repair it if necessary, and restore it.
36	If the on-line CU was not taken out of service, switch CUs and remove it. Diagnose it, repair it if necessary, and restore it.

CUSTAT

Software Release: 5E10 and later

1. CONTROL UNIT HARDWARE STATUS (CUSTAT) AUDIT

The control unit hardware status (CUSTAT) audit periodically verifies the control unit (CU) hardware status. Its main function is to guarantee a useful CU hardware configuration. Inconsistent hardware status and unacceptable hardware configurations are corrected when possible. The audit checks:

- The hardware status information in the system status registers (SSRs).
- The consistency and validity of the ECD.
- The backup maintenance channel bits in the hardware status register (HSR) of the on-line processor.

The CUSTAT interprets both a maintenance channel failure and a read off-line store failure as the other processor is being powered down. The audit does not fail any tests because the other processor is powered down as long as the other processor is not marked active or standby.

In addition, CUSTAT performs a memory comparison of active and standby main store units similar to CUMEM. Failures are reported against the on-line cache unit or the off-line CU.

Excessive error counts during the running of this audit (in correcting mode) may result in a hardware reconfiguration with one of the CUs being marked out of service (OOS).

1.1 Requesting the CUSTAT Audit

To request the CUSTAT audit, use the following input message:

```
AUD:CUSTAT=1 [:a|:b|:a,b];
```

where:

a	= Indicates whether audit should correct errors.	
	CORR	= An audit should correct errors (default).
	NCG	= An audit does not correct errors.
b	= Type of report generated. Valid values are:	
	DETL	= Generates a raw data report for each error found plus an audit completion message (default). A limited number of raw data reports are generated.
	SUM	= Generates only an audit completion message when the audit terminates. Raw data reports are not generated for any errors that are found.

To interpret system responses to the AUD:CUSTAT input message, refer to the APP:AUD Appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD:CUSTAT Output Message and Error Reports

The AUD:CUSTAT output message reports the termination status of the CUSTAT audit and/or any errors found while executing the audit. For a detailed explanation of the AUD:CUSTAT output message, see the Output Messages manual (235-600-750).

Table 1 lists the CUSTAT audit error numbers and their associated categories.

Table 1 CUSTAT AUDIT ERROR NUMBERS

Category	Error Number	Meaning
0,3	2	Bad data in the equipment configuration database (ECD) for a CU unit or subunit.
2,3	4	Could not access the ECD. Possibly a bad process identification (PID).
0	6	Maintenance channel failure.
0	8	Reading off-line memory using update circuit failed.
1	10	Kconfig function call failed.
0	12	CU was found to have the wrong major state in ECD.
0,1	14	On-line SSR value was incorrect.
0,1	16	Off-line SSR value was incorrect.
0,1	18	On-line HSR value was incorrect.
0	20	Off-line HSR value was incorrect.
0	22	Cache hit-miss ratio is unacceptable.
0	24	PMS database access failure.
0	26	Audit control library function call failure.
0	28	Internal PCPAUD failure.
0	30	Off-line store access failure.
1	32	Memory audit internal error.
0	34	Off-line memory mismatch.
0	36	On-line cache memory mismatch.
0	42	EAI buffers mismatch.

For error numbers 30, 34, and 36, the DATA1 field contains the main store address at which the test failed. For all other error messages, the DATA1 field contains the assert number. 2 lists the assert numbers that may be reported and explains their meanings.

Table 2 CUSTAT ERROR REPORTS

Assert Number	Meaning
0x1000	RID for on-line CU is IDNULL; stop audit.
0x1001	Could not get on-line CU UCB from ECD; stop audit.
0x1002	On-line CU UCB indicates device type other than DEV CU; stop audit.
0x1003	RID for on-line is same as RID for off-line; stop audit.
0x1004	Could not get off-line CU UCB from ECD; stop audit.
0x1005	Off-line CU UCB indicates device type other than DEV CU; stop audit.
0x1006	Could not get on-line CU minor device chain table from ECD; stop audit.
0x1007	MDCT for the on-line CU has too many m_minxt entries indicates that there were more than two CUs).
0x1008	MDCT for the on-line CU has too many m_minxt entries indicates that there were more than two CUs).
0x1009	m_minxt[0] equals m_minxt[1] indicating that CU0 and CU1 have same RIDs; stop audit.
0x100a	MDCT m_minxt[0] does not equal RID for on-line or off-line CU; stop audit.
0x100b	MDCT m_minxt[1] does not equal RID for on-line or off-line CU; stop audit.
0x1100	Faulty maintenance channel circuit.
0x1101	Off-line memory access failure; update circuit not functional.
0x1102	Kconfig fltdrvr call did not complete successfully.
0x1103	Kconfig fltdrvr call did not complete successfully.
0x1105	Maintenance channel (MCH) and update circuit not functional; but other CU UCB u_stat is standby (STBY).
0x1106	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x1107	Call to kconfig chgstate to change off-line CU UCB u_stat to UNAV unsuccessful.
0x1108	Call to kconfig chgstate to change off-line CU UCB u_stat to UNAV unsuccessful.
0x1109	Call to ECD lla_rdget to get off-line CU UCB unsuccessful.
0x110b	Call to ECD chucb to change off-line CU UCBs u_manrst to ON unsuccessful.
0x110c	Call to kconfig chgstate to change off-line CU UCB u_stat to OOS unsuccessful.
0x110d	Call to ECD chucb to change off-line CU UCBs u_manrst to OFF unsuccessful.
0x110e	Call to ECD lla_rdget to get off-line CU UCB unsuccessful.
0x1200	On-line CU UCB u_stat not active (ACT) or initializing (INIT).
0x1201	Call to kconfig chgstate to change on-line CU UCB u_stat to ACT unsuccessful.
0x1202	Off-line CU SSR (SSR_CC bit 19) indicated it is the on-line CU.
0x1203	Call to PCPAUD initofl to initialize off-line CU unsuccessful.
0x1204	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x1205	Off-line CU UCB u_stat is ACT.
0x1206	Call to kconfig fltdrvr to remove other CU unsuccessful. Other CU UCB u_stat is ACT.

0x1207	Simplex mode on-line CU SSR has force off-line (SSR_FOFL bit 9) asserted.
0x1208	On-line CU HSR bit 8 (backup maintenance channel bit) asserted.
0x1209	On-line CU HSR bit 8 or bit 9 (backup maintenance channel bits) asserted.
0x120a	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x120b	On-line CU SSR IO disable (SSR_DSAB bit 10) asserted.
0x120c	On-line CU SSR isolate DMA from MAS bus (SSR_ISOD bit 20) asserted.
0x120d	Off-line CU SSR initialization sequence control bits (SSR_ISCO or SSR_ISC1 bits 4 and 5) asserted.
0x1300	On-line CU SSR is in enable update mode (SSR_UPD bit 19 asserted) but off-line CU SSR has isolate other (on-line) CU from main store bus (SSR_ISOU bit 21) asserted.
0x1301	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x1302	Off-line CU SSR is in enable update mode (SSR_UPD bit 19 asserted) or off-line CU SSR has isolate other (on-line) CU from main store bus (SSR_ISOU bit 21) asserted.
0x1303	Call to PCPAUD initofl to initialize off-line CU unsuccessful.
0x1304	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x1305	Simplex on-line CU SSR in enable update mode (SSR_UPD bit 19 asserted).
0x1306	On-line CU SSR not isolating other (off-line) CU from its main store bus (SSR_ISOU bit 21 not asserted).
0x1307	3B21D CUs only: On-line CU SSR not isolating its expansion slots from its main store bus (SSR_ISOIE bit 22 not asserted).
0x1308	3B21D CUs only: Off-line CU SSR not isolating its expansion slots from its main store bus (SSR_ISOIE bit 22 not asserted).
0x1309	Call to PCPAUD initofl to initialize off-line CU unsuccessful.
0x1400	On-line CU UCB u_update is OFF (not updated).
0x1401	Could not get on-line CU UCB from ECD; stop audit.
0x1402	Call to ECD lla_uprid to change on-line CU UCB u_update to ON unsuccessful.
0x1403	Off-line CU UCB u_stat is STBY but off-line CU UCB u_update is OFF (not updated) or on-line CU SSR is not in enable update mode (SSR_UPD bit 19 not asserted).
0x1404	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x1405	On-line CU SSR not in enable update mode (SSR_UPD bit no asserted), but off-line CU UCB u_update is ON (updated).
0x1406	Call to ECD chuchb to change off-line CU UCBs u_update to OFF unsuccessful.
0x1500	3B20D CUs only: Duplex configuration, but on-line CU SSR indicating simplex (SSR_SIDU bit 2 not asserted).
0x1501	3B20D CUs only: Same as 0x1500 except off-line CU SSR indicating simplex (SSR_SIDU bit 2 not asserted).
0x1502	Simplex configuration but on-line CU indicating duplex (SSR_SIDU bit 2 asserted).
0x1503	Off-line CU SSR has force on-line (SSR_FONL bit 10) asserted.
0x1504	On-line CU SSR forced on-line (SSR_FONL bit 10) asserted, off-line CU SSR forced off-line (SSR_FONL bit 9) not asserted.
0x1505	On-line CU SSR forced on-line (SSR_FONL bit 10) not asserted, off-line CU SSR forced off-line (SSR_FOFL bit 9) asserted.
0x1506	Both CU SSRs indicate they are CU 1 (SSR_CCID bit 0 asserted).
0x1507	Both on-line and off-line CU think they are CU0 (SSR_CCID bit 0 not asserted).
0x1508	Off-line CU SSR has power clear (SSR_PCLR bit 15) asserted.
0x1509	Call to PCPAUD initofl to initialize off-line CU unsuccessful.
0x150a	HSR bit 8, 9, or 31 asserted in off-line CU.
0x150b	Call to PCPAUD initofl to initialize off-line CU unsuccessful.
0x150c	Call to ECD gnxt unsuccessful in on-line CU UCB chain.
0x150d	Calls to ECD gnxt did not find a CC UCB (u_dtype=DEV_CC) in the on-line CU UCB chain; stop audit.
0x150e	Call to ECD gnxt unsuccessful in off-line CU UCB chain.
0x150f	Calls to ECD gnxt did not find a CC UCB (u_dtype=DEV_CC) in the off-line CU UCB chain; stop audit.
0x1510	On-line CU 1 CC 0 UCB does not have u_opccid (bit 4) asserted in u_equip.
0x1511	On-line CU 0 CC 0 UCB has u_opccid (bit 4) asserted in u_equip.
0x1512	Off-line CU 0 CC 0 UCB has u_opccid (bit 4) asserted in u_equip.
0x1513	Off-line CU 1 CC 0 UCB does not have u_opccid (bit 4) asserted in u_equip.
0x1514	3B20D CUs only: Duplex configuration, but on-line CU CC 0 UCB specifying simplex (u_opsimdup bit 6 not asserted in u_equip).
0x1515	3B20D CUs only: Duplex configuration, but off-line CU CC 0 UCB specifying simplex (u_opsimdup bit 6 not asserted in u_equip).
0x1516	3B20D CUs only: Simplex configuration, but CU CC 0 UCB specifying duplex (u_opsimdup bit 6 asserted in u_equip).
0x1517	On-line CU SSR has power clear (SSR_PCLR bit 15) asserted.
0x1518	Call to ECD lla_rdget to get on-line CU MASC UCB unsuccessful.
0x1519	Call to ECD lla_rdget to get off-line CU MASC UCB unsuccessful.
0x151a	3B20D CUs only: On-line CU memory controller type indicators inconsistent between the SSR (SSR_VLMAS bit 1) and the MASC UCB.
0x151b	3B20D CUs only: Off-line CU memory controller type indicators inconsistent between the SSR (SSR_VLMAS bit 1) and the MASC UCB.

0x151c	Call to kconfig fltdrvr to remove other CU unsuccessful.
0x151d	Call to ECD lla_rdgct to get off-line CU UCB unsuccessful.
0x1600	EAI buffers in CU 0 and CU 1 mismatch.
0x1601	Call to PCPAUD getofflpb to read off-line CU EAI IPB unsuccessful.
0x2000	Memory comparison did not complete its entire memory block comparison.
0x2001	Call to kconfig rpterrr unsuccessful.
0x2002	Call to kconfig rpterrr unsuccessful.
0x2003	Call to kconfig fltdrvr to remove other CU unsuccessful.

Under some error conditions, the CUSTAT audit may produce additional output messages. These reports have the following format:

```
REPT PCPAUD FAULT TYPE a ASSERT NO b
```

where:

a = Fault type.

b = Assert number.

For a detailed explanation of the REPT:PCPAUD output message and a list of fault types and assert numbers, see the Output Messages manual (235-600-750).

1.3 Responding to AUD:CUSTAT Error Reports

For the listed error numbers, take the action designated in the following table. For any other error numbers, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

Error Number:	Action:
0x4	Run the ECD 1 audit.
0x6	Check maintenance channel cable. Run CC diagnostics. The problem could also be in the online maintenance channel. The problem is serious and should be taken care of as soon as possible.
0x8	The off-line CU should have been removed automatically. Diagnose it, repair it if necessary, and restore it. Then switch CUs and remove, diagnose, repair, and restore the other CU.
0xe	Take action on this error only if it occurs repeatedly. Switch CUs, then remove, diagnose, repair, and restore the affected CU.
0x10	Take action on this error only if it occurs repeatedly. Remove, diagnose, repair, and restore the off-line CU.
0x12	Take action on this error only if it occurs repeatedly. Either CU could cause bad value in the audited on-line HSR bits. Diagnose the off-line CU, repair it if necessary, and restore it. Then switch CUs and remove, diagnose, repair, and restore the other CU.
0x14	Take action on this error only if it occurs repeatedly. Either CU could cause bad value in the audited off-line HSR bits. Remove, diagnose, repair, and restore the off-line CU. Then switch CUs and remove, diagnose, repair, and restore the other CU.
0x16	Switch CUs. Remove the faulty CU, replace the cache memory board, diagnose the CU, and restore it. Then switch CUs again (that is, return to the original configuration) and rerun the audit.
0x18	Run the PMS 1 audit.
0x1e	The off-line CU should have been removed automatically. Diagnose it, repair it if necessary, and restore it. Then switch CUs and remove, diagnose, repair, and restore the other CU.
0x22	If the off-line CU was not taken out of service, remove it. Diagnose it, repair it if necessary, and restore it.
0x24	If the on-line CU was not taken out of service, switch CUs and remove it. Diagnose it, repair it if necessary, and restore it.
0x2a	Clear the EAls with a poker 14 on the EAI page of the MCRT.
0x2c	Take action on this error only if it occurs repeatedly. Either CU could cause inconsistent values between the SSRs. Run CU diagnostics, switch CUs, run CU diagnostics.

CVADMGP

Software Release: 5E11 and later

1. CONVERSION ADMINISTRATION GROUP TABLE (CVADMGP) AUDIT

CVADMGP audit steps through the static relation RLCV_ADMGP (SCCP conversion for administration group) and check CCSCCPADM table. Error in the table will result in the table being rebuilt.

2. DATA STRUCTURES

For details on the fields contained in the RLCV_ADMGP relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ADMGRP_ERR

Error Code:	ADMGRP_ERR
Error Description:	A CCSCCPADM table entry has been found to differ from its counterpart in rICV_ADMGP. The error address value can be used to determine which particular field the audit is complaining about. The following matches the error address value with the field in error. 5 = adm_grp
Possible Error Effect:	Either network control point (NCP) queries could fail, or signaling connection control part (SCCP) messages could potentially not be converted to the proper format for the outgoing network.
Corrective Action Taken:	The incorrect field is corrected from static data, and the corresponding update is made on all protocol handlers (PHs).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	See previous error description.
<i>Bad Data =</i>	Dynamic field value.
<i>Good Data =</i>	Static field value.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	Entire dynamic entry.

Table 2 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The number of valid entries in the CCSCCPADM table as counted by the audit does not match the entry count in the table dictionary.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild [with protocol handler (PH) updates], and then the audit stops.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Table ID (CCSCCPADM).
<i>Bad Data</i> =	Dictionary version of the number of entries.
<i>Good Data</i> =	The number of entries found by the audit.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The size of the CCSCCPADM table has grown beyond the maximum size that SCCP audits allow (AUCCGBITS). Change the definition of AUCCGBITS to an appropriate larger value.
Possible Error Effect:	This table is not audited.
Corrective Action Taken:	The audit returns to audit control without performing any checks on the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Maximum table size allowed by SCCP audits.
<i>Good Data</i> =	Maximum table size from the table dictionary.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: INSERT_FAILED

Error Code:	INSERT_FAILED
Error Description:	An attempt to insert an entry into the CCSCCPADM table failed. It exists in rICV_ADMGP, but not in the table.
Possible Error Effect:	NCP query messages could be lost.
Corrective Action Taken:	The audit requests a table rebuild (with PH updates), and then the audit stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 5 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A static RLCV_ADMGP relation tuple has been found to have no counterpart in the CCSCCPADM table.
Possible Error Effect:	NCP query messages could be lost.
Corrective Action Taken:	An attempt is made to insert the entry (with values drawn from the static tuple).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-null entry has been found in the CCSCCPADM table that does not exist in the static relation RLCV_ADMGP; or, the entry exists in the static relation, but the full ct field of the static tuple is greater than 0.
Possible Error Effect:	SCCP recent change may encounter errors or difficulties.
Corrective Action Taken:	The entry is nulled out (deleted), and the PH copies are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the beginning of the non-null entry.
<i>Bad Data</i> =	Offset (into table) of the non-null entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.

<i>Dump</i> =	Entire non-null entry.

CVCDPY

Software Release: 5E11 and later

1. CONVERSION CALLED PARTY (CVCDPY) AUDIT

CVCDPY audit steps through the static relation RLCV_CDPY (SCCP conversion for called party address) and check CCSCCPCLD table. Error in the table will result in the table being rebuilt.

2. DATA STRUCTURES

For details on the fields contained in the RLCV_CDPY relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The number of valid entries in the CCSCCPCLD table as counted by the audit does not match the entry count in the table dictionary.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests a table rebuild [with protocol handler (PH) updates], and then the audit stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Table ID (CCSCCPCLD).
<i>Bad Data =</i>	Dictionary version of the number of entries.
<i>Good Data =</i>	The number of entries found by the audit.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: CLDPTY_ERR

Error Code:	CLDPTY_ERR
Error Description:	A CCSCCPCLD table entry has been found to differ from its counterpart in rICV_CDPY. The error address value can be used to determine which particular field the audit is complaining about. The following list matches the error address value with the field in error. 5 = num_dlt 6 = new_tt 7 = new_nai 8 = new_np

	9 = new_ensch 10 = ssn_map 11 = new_rti 12 = new_gti 13 = src_nai 14 = src_np 15 = src_ensch 16 = adj_ind NOTE: For any occurrence of the audit, no more than two CLDPTY_ERR error codes are issued.
Possible Error Effect:	Either network control point (NCP) queries could fail, or signaling connection control part (SCCP) messages could potentially not be converted to the proper format for the outgoing network.
Corrective Action Taken:	The incorrect field is corrected from static data, and the corresponding update is made on all PHs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	See previous error description.
<i>Bad Data</i> =	Dynamic field value.
<i>Good Data</i> =	Static field value.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire dynamic entry.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The size of the CCSCCPCLD table has grown beyond the maximum size that SCCP audits allow (AUCCGBITS). Change the definition of AUCCGBITS to an appropriate larger value.
Possible Error Effect:	This table is not audited.
Corrective Action Taken:	The audit returns to audit control without performing any checks on the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Maximum table size allowed by SCCP audits
<i>Good Data</i> =	Maximum table size from the table dictionary
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: INSERT_FAILED

Error Code:	INSERT_FAILED
Error Description:	An attempt to insert an entry into the CCSCCPCLD table failed. It exists in rICV CDPY, but not in the table.
Possible Error Effect:	NCP query messages could be lost.
Corrective Action Taken:	The audit requests a table rebuild (with PH updates), and then the audit stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 5 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A static rICV_CDPY relation tuple has been found to have no counterpart in the CCSCCPCLD table.
Possible Error Effect:	NCP query messages could be lost.
Corrective Action Taken:	An attempt is made to insert the entry (with values drawn from the static tuple).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	Copy of the entry buffer with the key fields initialized.

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-null entry has been found in the CCSCCPCLD table that does not exist in the static relation RLCV_CDPY.
Possible Error Effect:	SCCP recent change may encounter errors or difficulties.
Corrective Action Taken:	The entry is nulled out (deleted), and the PH copies are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the beginning of the non-null entry.
<i>Bad Data =</i>	Offset (into table) of the non-null entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	Entire non-null entry.

CVCGPY

Software Release: 5E11 and later

1. CONVERSION CALLING PARTY TABLE (CVCGPY) AUDIT

CVCGPY audit steps through the static relation RLCV_CGPY (SCCP conversion for calling party address) and check CCSCCPCNG table. Error in the table will result in the table being rebuilt.

2. DATA STRUCTURES

For details on the fields contained in the RLCV_CGPY relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The number of valid entries in the CCSCCPCNG table as counted by the audit does not match the entry count in the table dictionary.
Possible Error Effect:	Unknown
Corrective Action Taken:	The audit requests a table rebuild [with protocol handler (PH) updates], and then the audit stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Table ID (CCSCCPCNG).
<i>Bad Data =</i>	Dictionary version of the number of entries.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: CNGPTY_ERR

Error Code:	CNGPTY_ERR
Error Description:	A CCSCCPCNG table entry has been found to differ from its counterpart in RLCV_CGPY. The error address value can help to determine which particular field the audit is complaining about. The following list matches the error address value with the field in error. 3 = src_fpc 4 = src_spc 7 = st_pcs 8 = new_tt

	9 = src_nai 10 = new_nai 11 = new_gti 12 = new_np 13 = src_np 14 = src_enssch 15 = new_enssch 16 = new_rti 17 = adj_ind NOTE: For any occurrence of the audit, no more than two CNGPTY_ERR error codes are issued.
Possible Error Effect:	Either network control point (NCP) queries could fail, or signaling connection control part (SCCP) messages could potentially not be converted to proper format for the outgoing network.
Corrective Action Taken:	The incorrect field is corrected from static data, and the corresponding update is made on all protocol handlers (PHs).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	See previous error description.
<i>Bad Data</i> =	Dynamic field value.
<i>Good Data</i> =	Static field value.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire dynamic entry.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The size of the CCSCCPCNG table has grown beyond the maximum size that SCCP audits allow (AUCCGBITS). Change the definition of AUCCGBITS to an appropriate larger value.
Possible Error Effect:	This table is not audited.
Corrective Action Taken:	The audit returns to audit control without performing any checks on the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Maximum table size allowed by SCCP audits.
<i>Good Data</i> =	Maximum table size from the table dictionary.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: INSERT_FAILED

Error Code:	INSERT_FAILED
Error Description:	An attempt to insert an entry into the CCSCCPCNG table failed. It exists in rICV_CGPY but not in the table.
Possible Error Effect:	NCP query messages could be lost.
Corrective Action Taken:	The audit requests a table rebuild (with PH updates), and then the audit stops.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.

<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 5 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A static RLCV_CGPY relation tuple has been found to have no counterpart in the CCSCCPCNG table.
Possible Error Effect:	NCP query messages could be lost.
Corrective Action Taken:	An attempt is made to insert the entry (with values drawn from the static tuple).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Copy of the entry buffer with the key fields initialized.

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-null entry has been found in the CCSCCPCNG table that does not exist in the static relation RLCV_CGPY.
Possible Error Effect:	SCCP change may encounter errors or difficulties.
Corrective Action Taken:	The entry is nulled out (deleted), and the PH copies are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the beginning of the non-null entry.
<i>Bad Data</i> =	Offset (into table) of the non-null entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entire non-null entry.

CVSTATE

Software Release: 5E10 and later

1. CALL VECTORING STATES (CVSTATE) AUDIT

The new CVSTATE audit recovers any lost rICV_STATE tuples by initializing them and linking them on the idle list. Any error found with the rICV_STATE audit is reported, and the tuple is recovered. The audit also checks the RLCV_STATE relation for correct linkages, and corrects semantic data. This new audit runs in the routine mode in the normal audit cycle, and in the elevated mode when scheduled by a user request.

2. DATA STRUCTURES

The CVSTATE audit is executed using the DRINASEK driver. The driver provides syntactic checks by stepping through the rICV_STATE's tuples by using the relation key and examining the quarantine field and key field of each tuple. If these fields are not mutilated, the driver proceeds to validate the owner-member linkage between rICV_STATE and rIDBHEADC, and the one-to-one linkage between rICV_STATE and rPCBLA (Refer to Figure 1). If any error is detected, the audit driver DRINASEK prints an audit message associated with the error condition, zeros out the rICV_STATE tuple, returns the tuple to the idle list, and requests audit PCBLA as a related audit.

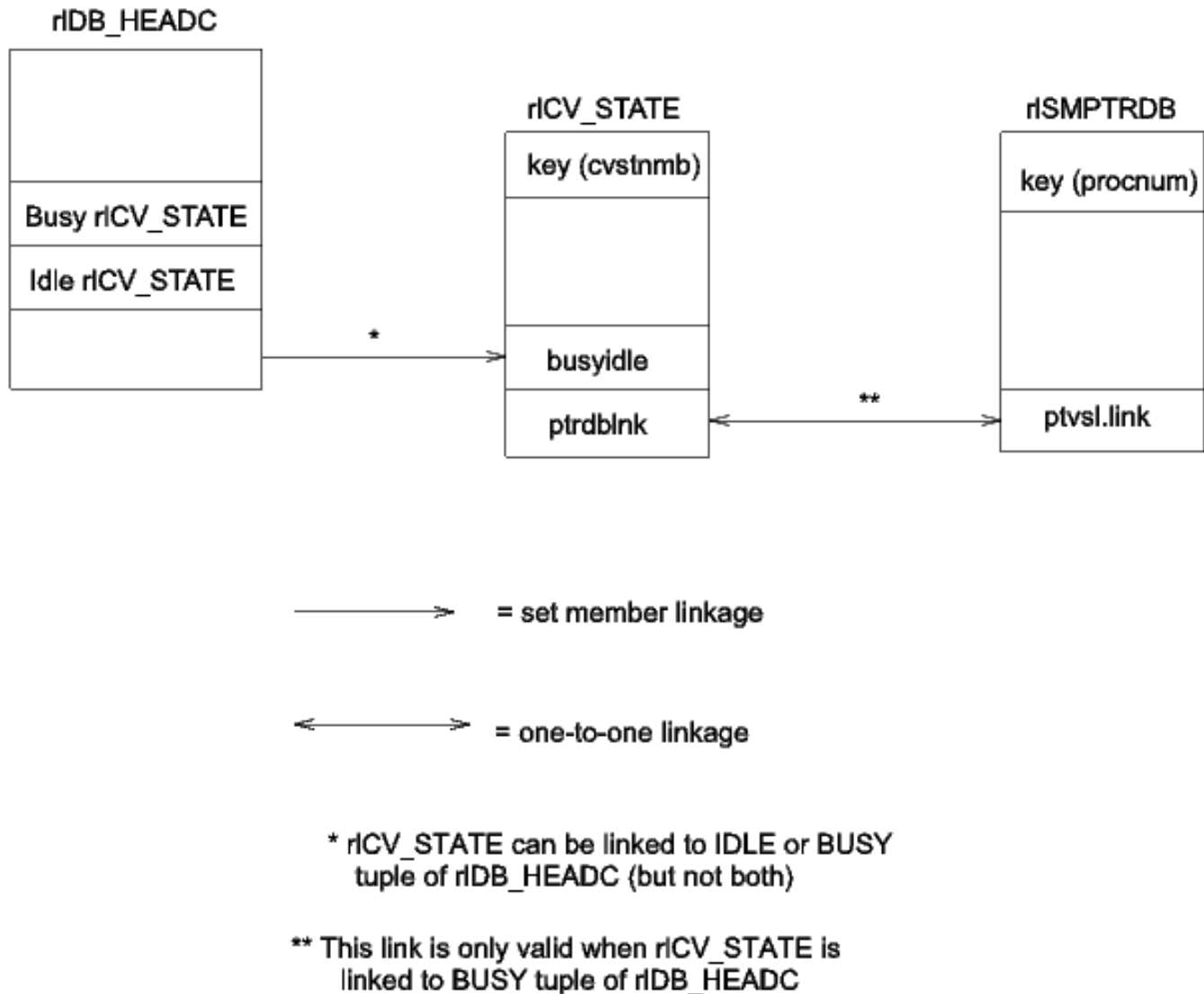


Figure 1 (CVSTATE-1) CVSTATE Relation Linkages

3. AUDIT CHECKS

The CVSTATE audit invokes the DRINASEK driver to execute the syntactic checks and invokes the semantic function. For each tuple, the CVSTATE audit makes the following checks after the driver has verified the linkages. For details regarding each error code, see the comment blocks given.

NOTBUSY = Check if the tuple should be on the busy list.

NOTIDLE = Check if the tuple should be on the idle list.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A tuple, with valid linkages and data to qualify as busy, is not on the busy list.
Possible Error Effect:	If this tuple is on the idle list, the data could be lost when the supposedly idle tuple is made active.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICV_STATE tuple.
<i>Bad Data</i> =	Contents of st_owrk field.
<i>Good Data</i> =	Busy list head cell.
<i>Logical Index</i> =	Key to rICV_STATE tuple.
<i>Dump</i> =	Contents of rICV_STATE tuple.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A tuple that has no link into RLPCBLA is not fully idled.
Possible Error Effect:	When the tuple is made active again, some fields may be carried over.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICV_STATE tuple.
<i>Bad Data</i> =	Contents of st_owrk field.
<i>Good Data</i> =	Idle list head cell.
<i>Logical Index</i> =	Key of rICV_STATE tuple.
<i>Dump</i> =	Contents of rICV_STATE tuple.

13. DA - DH

DALB-A

Software Release: 5E10 - 5E11

1. D-CHANNEL APPLICATION LINKAGE BLOCK (DALB) AUDIT

The D-channel application linkage block (RLDALB) and the D-channel protocol block (RLDPB) are used to handle communications between the integrated services digital network (ISDN) periphery and the processes that control the ISDN ports. The rIDPB tuple corresponds to the D-channel port, and has one rIDALB tuple linked to it for each process that needs to communicate with the ISDN customer premises equipment (CPE). Most (but not all) of these processes control a B-port served by the D-port. The RLDALB relation represents per call and terminal dynamic data. For every ISDN call, whether it is an active call or an associated call, a rIDALB tuple is allocated. It may be used for each ISDN call or used for distributing maintenance message. It has a semantic linkage with the terminal endpoint identifier linkage area (RLTEILA). RLDALB is linked to terminal equipment data block (RLTEDB) and RLDPB for [plain old circuit-switched ISDN call (PODS) or business and residential customer services (BRCS)]. It is also linked to primary rate interface group status (RLPRISTAT) for non-OSPS PRI calls. The *dalb_type* field in DALB indicates the purpose of DALB. The DALB audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment by the initialization, demand, and segmented modes.

2. DATA STRUCTURES

The RLDALB relation is indexed access and the audit uses the DRINASEK driver to step through the tuples. The rIDALB has two separate one-to-one linkages with protocol control block linkage area (RLPCBLA) for:

- (a) ISDN plain old circuit switching ISDN call (PODS) and business and residential customer service (BRCS), OSPS, applications processor (AP), and extended digital subscriber line (EDSL).
- (b) for ISDN Attendant.

The rIDALB shares a set-member linkage (*dadpm*) with:

RLDBHEADC	= The RLDALB idle list.
RLPRISTAT	= PRI calls.
RLDPB	= Active calls and other maintenance.

When the rIDALB tuple is not idle, the attribute *pri_dpb_bit* will identify the linkage as associated with an rIPRISTAT tuple when the attribute is DBYES, or to the rIDPB tuple when the attribute is DBNO. Refer to Figure 2. There may be other data blocks linked to the RLDALB depending on the application, for example the supplementary protocol block (RLSPB) for operating system for distributed switching (OSDS) applications. The RLDALB relation represents per-call dynamic data. For every ISDN/OSPS/AP/EDSL call, whether it is an active call or associated call, a rIDALB tuple is allocated. Refer to Figure 1.

3. AUDIT CHECKS

- The audit first checks the quarantine field in the rIDALB tuple.
- The audit verifies that the key in the rIDALB tuple is correct.
- The audit verifies that the RDL linkages between rIDALB and other relations are correct.
- The audit determines if the rIDALB is currently active on an ISDN call by seeing if it is linked to the RLDBHEADC idle list head cell. If the rIDALB is not active, the audit does no further checking.

- The audit verifies that the process that owns an active rIDALB is a live process.
- The audit performs a transient timing check for rIDALBs that have been previously recovered and await synchronization.
- The audit verifies a possible shared call process ID in rPCBLA tuple.
- The audit checks the tedb_flag and linkages to rBRCSD and rPCBLA tuples.
- The audit verifies the semantic copies of static RLSLEQUIP information.
- The audit verifies semantic data between rIDALB and rTEDB.
- During normal operation (not selective initialization) the audit interfaces with FC to perform a call activity check (CAC). This interface sends a status inquiry message to the terminal to verify the call data.
- The audit verifies the RDL linkage with rTEDB and the tedb_flag.
- The audit performs a point-to-point-back check on the semantic linkage to rTEDB based on dalb_type.
- The audit verifies semantic data between rIDALB and rDPB tuples.
- The audit performs a point-to-point-back check on the semantic linkage to rTEILA tuples.
- There is a special recovery function to reconstruct the rIDALB.
- The audit verifies the consistency of procedures used by user-side primary rate interface (PRI). Either user-side or network-side procedures are allowed. The rIDALB tuple is checked with the rDSLEQUIP tuple and the rIDALB tuple is corrected if an inconsistency is found.
- The audit verifies the key of the rPRISTAT tuple (MEMBAD).
- The audit verifies the PID of the rPRISTAT tuple when one of the call activity requests is set (PID_MISMATCH).

DALB audit is modified to check the semantic linkage with RLTEILA. The linkage with rTEDB tuple is added in DALB audit. For each type of rIDALB, a special check is performed to verify the data consistency. Every DALB error results in single process purge and CPE synchronization function is called to synchronize the call status.

4. ERROR HANDLING

Every DALB error results in a single process purge and the CPE synchronization function is called to synchronize the call status. If any errors are detected in rIDALB, the audit determines if the process corresponding to the rIDALB is alive. If the process is alive, the audit recovers the data by requesting a single process purge of the owning process. If the rIDALB does not have a living process, the audit unlinks the rIDALB from the rDPB and rTEDB, and reinitializes all of the fields in the rIDALB. The CPE synchronization function is called to synchronize the call status.

If the rIDALB is properly linked to the rTEDB tuple, it will be placed in a transient state and an integrated services port administrator (ISPA) job schedules to recover it later, as part of customer premises equipment (CPE) resynchronization.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The first linkage to rIPCBLA tuple (dalb_lnk.lnk) is not null but the second rIPCBLA tuple linkage is zero (dadpm.st link).
Possible Error Effect:	The call is not set up correctly. Call processing messages will not be routed correctly.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Second linkage to rIPCBLA tuple.
<i>Good Data =</i>	First linkage to rIPCBLA tuple.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The tedbflag in the rIDALB tuple does not match the tedbflag in the owning rIDPB tuple.
Possible Error Effect:	Call processing software will access the wrong data and cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	tedbflag in rIDALB tuple.
<i>Good Data =</i>	tedbflag in rIDPB tuple.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	1rIDPB tuple.

Table 3 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The rIDALB tuple is supposed to be in the idle list but the linkage has an error.
Possible Error Effect:	The rIDALB tuple cannot be used for call processing.
Corrective Action Taken:	The rIDALB tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Linkage fields to rIDBHEADC tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 4 Error Code: BADIS_DTE

Error Code:	BADIS_DTE
Error Description:	The user/network bit in the rIDALB tuple does not match the user/network bit in rIDSLEQUIP tuple.
Possible Error Effect:	The rIDALB tuple is not set up properly for user/network PRI's.
Corrective Action Taken:	Correct this rIDALB tuple to agree with the rIDSLEQUIP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	User/network bit in rIDALB tuple.
<i>Good Data =</i>	User/network bit in rIDSLEQUIP tuple.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	Contents of rIDALB tuple.

Table 5 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The dalbty in the rIDALB tuple is out of range.
Possible Error Effect:	The call processing software may not be able to properly tear down this call or

	activate features.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Value in dalbtype field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 6 Error Code: BRCSFLAG

Error Code:	BRCSFLAG
Error Description:	The brcsflag in rIDALB tuple does not match the brcsflag in rITEDB tuple.
Possible Error Effect:	The call processing software will not be able to properly activate BRCS features for this call.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	brcsflag in rIDALB tuple.
<i>Good Data =</i>	brcsflag in rITEDB tuple.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	rITEDB tuple.

Table 7 Error Code: BRCSLINK

Error Code:	BRCSLINK
Error Description:	The rIDALB tuple not used by ISDNBRCS should not link to RLBRCSDB data structure.
Possible Error Effect:	The system could run out of rIBRCSDB tuple data blocks.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Linkage fields to rPCBLA tuple and rIBRCSDB tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 8 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	The rIDALB tuple not used by ISDNBRCS is not owned by a terminal process (PCBLA).
Possible Error Effect:	No terminal process controls the call.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Linkage fields to rPCBLA tuple and rIBRCSDB tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 9 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	This DSL equipage does not exist but the rIDPB tuple does. It probably is an internal ODD problem.
Possible Error Effect:	The call is unable to receive signals from customer equipment.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.

<i>Bad Data</i> =	Linkage fields to rIDPB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 10 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	rIDALB tuple is linked to a rIDPB tuple or rITEDB tuple, but the owning process is not active.
Possible Error Effect:	Message may be sent to inactive process.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Process number of this rIDALB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 11 Error Code: DLOGCHAN

Error Code:	DLOGCHAN
Error Description:	The level 3 logical channel number for this rIDALB tuple is not valid.
Possible Error Effect:	Level 3 messages cannot be properly processed with an invalid L3 logical channel number.
Corrective Action Taken:	The audit purges the process owning rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Invalid logical channel number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 12 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The rIDALB tuple is supposed to link to rIDPB tuple but the linkage is in error.
Possible Error Effect:	The call will not handle D-channel message properly.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDPB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 13 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The logical PH for this rIDALB tuple does not match that of the D-PORT (which is populated in rITEDB tuple).
Possible Error Effect:	Signals from the switch to the customer premise equipment are misrouted or lost.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Logical PH in the rIDALB tuple.
<i>Good Data</i> =	Logical PH in the TEDB.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 14 Error Code: DSL_MEM

Error Code:	DSL_MEM
Error Description:	The member number for this rDALB tuple does not match that of the D-PORT.
Possible Error Effect:	Signals from the switch to the customer premise equipment are misrouted or lost.
Corrective Action Taken:	Process owning rDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDALB tuple.
<i>Bad Data =</i>	Logical MEM in the rDALB tuple.
<i>Good Data =</i>	Logical MEM in the rDSLEQUIP tuple.
<i>Logical Index =</i>	Logical key of rDALB tuple.
<i>Dump =</i>	rDALB tuple.

Table 15 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rDALB tuple is in the idle state but is not linked to the RLDALB idle list.
Possible Error Effect:	The rDALB tuple is unusable by call processing.
Corrective Action Taken:	The audit idles the rDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDALB tuple.
<i>Bad Data =</i>	Owner link of rDALB tuple.
<i>Good Data =</i>	Physical key of idle head cell.
<i>Logical Index =</i>	Logical key of rDALB tuple.
<i>Dump =</i>	rDALB tuple.

Table 16 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	A rDALB tuple is associated with an rIPRISTAT tuple; however, the key to the rIPRISTAT tuple is incorrect.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rDALB or rIPRISTAT tuples.
Corrective Action Taken:	None by this audit; the related rIPRISTAT audit will be run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rDALB tuple.
<i>Bad Data =</i>	Address of the rIPRISTAT tuple (rDALB.dadpm.st_ownk).
<i>Good Data =</i>	The flag indicating whether the rDALB tuple is linked to the RLDPB or RLPRISTAT (rDALB.pri_dpb_bit).
<i>Logical Index =</i>	Logical key of rDALB tuple.
<i>Dump =</i>	The rDALB tuple that is in error.

Table 17 Error Code: MSGPTR

Error Code:	MSGPTR
Error Description:	This error code is used in two places for the following: - The rDALB tuple is linked to rTEDB tuple and the fwd_msgs field in rDALB tuple does not match the fwd_msgs field in rDALB tuple. - The rDALB tuple is not linked to rTEDB tuple but the fwd_msgs field is invalidly set to DBYES.
Possible Error Effect:	- Call processing will not be able to properly forward messages. - The rDALB tuple is improperly set up. No call processing errors should be expected.
Corrective Action Taken:	Recover this rDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	fwd_msgs in rIDALB tuple.
<i>Good Data</i> =	1 = fwd_msgs in TEDB. 2 = DBNO.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rTEDB tuple.

Table 18 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rIDALB tuple has a null value in the semantic linkage to rTEILA tuple but the tei and the (tedbflag or dalbtype) is in an inconsistent state.
Possible Error Effect:	The rIDALB tuple is not set up properly (maintenance rIDALB tuple or call tear down).
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 19 Error Code: PID

Error Code:	PID
Error Description:	The controlling process in the rIDALB tuple is different from that represented in the rPCBLA tuple.
Possible Error Effect:	The call will be handled by the wrong process.
Corrective Action Taken:	The audit idles the rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	PID field in the rIDALB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 20 Error Code: PID_MISMATCH

Error Code:	PID_MISMATCH
Error Description:	A rIDALB tuple is associated with an rIPRISTAT tuple and a call activity has been requested (rIDALB.cac_req or rIDALB.kill_req = DBYES); however, there is no process ID (rIPRISTAT.send_pid) associated with the activity.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIDALB or rIPRISTAT tuples.
Corrective Action Taken:	The call activity request fields (rIDALB.cac_req and rIDALB.kill_req) are set to DBNO.
Manual Action Required:	None.
<i>Error Address</i> =	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Bad Data</i> =	The bit in the rIDALB tuple indicating that the process is being killed (rIDALB.kill_req).
<i>Good Data</i> =	The flag in the rIDALB tuple indicating that the process has a call activity check (rIDALB.cac_req).
<i>Logical Index</i> =	Logical key of rIDALB tuple (rIDALB.dalb).
<i>Dump 1</i> =	The rIDALB tuple.
<i>Dump 2</i> =	The rIPRISTAT tuple.

Table 21 Error Code: TEDBFLAG

Error Code:	TEDBFLAG
Error Description:	This error code is used in two places for the following: - The rIDALB tuple is linked to rITEDB tuple but the tedbflag in rIDALB tuple is set to DBNO. - The rIDALB tuple is not linked to rITEDB tuple but the tedbflag is set to DBYES.
Possible Error Effect:	In all cases the call processing software will use the wrong data causing failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	1 = DBNO. 2 = DBYES.
<i>Good Data =</i>	1 = DBYES. 2 = DBNO.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 22 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The point-to-point-back check with the particular TEDB (indicated by dalbtype) has failed.
Possible Error Effect:	The call processing software will not be able to properly activate features on this call.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	dalbtype.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	rITEDB tuple.

Table 23 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	The rIDALB tuple has a non-null value in the semantic linkage to rTEILA tuple but the point-to-point-back check with rTEILA tuple has failed.
Possible Error Effect:	The terminal may not be bound properly or the maintenance rIDALB tuple is improperly set up and will cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	rTEILA tuple.

Table 24 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The rIDALB tuple has been at transient state for too long, probably longer than audit cycle plus 5 minutes.
Possible Error Effect:	This tuple is in a stuck state. ISPA probably did not perform synchronization and release the rIDALB tuple. dalbtype reflects that it should be linked to rTEDB tuple.
Possible Error Effect:	The rIDALB tuple is not set up properly and the call processing or maintenance software may cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Dalbtype.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	1rIDALB tuple.

Table 25 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rIDALB tuple is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rIDALB tuple on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rIDALB tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Non-zero value of linkage.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

DALB-B

Software Release: 5E12 only

1. D-CHANNEL APPLICATION LINKAGE BLOCK (DALB) AUDIT

The D-channel application linkage block (RLDALB) and the D-channel protocol block (RLDPB) are used to handle communications between the integrated services digital network (ISDN) periphery and the processes that control the ISDN ports. The rIDPB tuple corresponds to the D-channel port, and has one rIDALB tuple linked to it for each process that needs to communicate with the ISDN customer premises equipment (CPE). Most (but not all) of these processes control a B-port served by the D-port.

The RLDALB relation represents per call and terminal dynamic data. For every ISDN call, whether it is an active call or an associated call, a rIDALB tuple is allocated. It may be used for each ISDN call or used for distributing maintenance message. It has a semantic linkage with the terminal endpoint identifier linkage area (RLTEILA). RLDALB is linked to terminal equipment data block (RLTEDB) and RLDPB for [plain old circuit-switched ISDN call (PODS) or business and residential customer services (BRCS)]. It is also linked to primary rate interface group status (RLPRISTAT) for non-OSPS PRI calls. The *dalb_type* field in DALB indicates the purpose of DALB. The DALB audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment by the initialization, demand, and segmented modes.

2. DATA STRUCTURES

The RLDALB relation is indexed access and the audit uses the DRINASEK driver to step through the tuples. The rIDALB has two separate one-to-one linkages with protocol control block linkage area (RLPCBLA) for:

- (1) ISDN plain old circuit switching ISDN call (PODS) and business and residential customer service (BRCS), OSPS, applications processor (AP), and extended digital subscriber line (EDSL).
- (2) for ISDN Attendant.

The rIDALB shares a set-member linkage (*dadpm*) with:

RLDBHEADC	= The RLDALB idle list.
RLPRISTAT	= PRI calls.
RLDPB	= Active calls and other maintenance.

When the rIDALB tuple is not idle, the attribute *pri_dpb_bit* will identify the linkage as associated with an rIPRISTAT tuple when the attribute is DBYES, or to the rIDPB tuple when the attribute is DBNO. Refer to Figure 2.

There may be other data blocks linked to the RLDALB depending on the application, for example the supplementary protocol block (RLSPB) for operating system for distributed switching (OSDS) applications. The RLDALB relation represents per-call dynamic data. For every ISDN/OSPS/AP/EDSL call, whether it is an active call or associated call, a rIDALB tuple is allocated. Refer to Figure 1.

3. AUDIT CHECKS

- The audit first checks the quarantine field in the rIDALB tuple.
- The audit verifies that the key in the rIDALB tuple is correct.
- The audit verifies that the RDL linkages between rIDALB and other relations are correct.

- The audit determines if the rDALB is currently active on an ISDN call by seeing if it is linked to the RLDBHEADC idle list head cell. If the rDALB is not active, the audit does no further checking.
- The audit verifies that the process that owns an active rDALB is a live process.
- The audit performs a transient timing check for rDALBs that have been previously recovered and await synchronization.
- The audit verifies a possible shared call process ID in rPCBLA tuple.
- The audit checks the tedb_flag and linkages to rBRCSD and rPCBLA tuples.
- The audit verifies the semantic copies of static RLDSLEQUIP information.
- The audit verifies semantic data between rDALB and rTEDB.
- During normal operation (not selective initialization) the audit interfaces with FC to perform a call activity check (CAC). This interface sends a status inquiry message to the terminal to verify the call data.
- The audit verifies the RDL linkage with rTEDB and the tedb_flag.
- The audit performs a point-to-point-back check on the semantic linkage to rTEDB based on dalb_type.
- The audit verifies semantic data between rDALB and rDPB tuples.
- The audit performs a point-to-point-back check on the semantic linkage to rTEILA tuples.
- There is a special recovery function to reconstruct the rDALB.
- The audit verifies the consistency of procedures used by user-side primary rate interface (PRI). Either user-side or network-side procedures are allowed. The rDALB tuple is checked with the rDSLEQUIP tuple and the rDALB tuple is corrected if an inconsistency is found.
- The audit verifies the key of the rPRISTAT tuple (MEMBAD).
- The audit verifies the PID of the rPRISTAT tuple when one of the call activity requests is set (PID_MISMATCH).

DALB audit is modified to check the semantic linkage with RLTEILA. The linkage with rTEDB tuple is added in DALB audit. For each type of rDALB, a special check is performed to verify the data consistency. Every DALB error results in single process purge and CPE synchronization function is called to synchronize the call status.

4. ERROR HANDLING

Every DALB error results in a single process purge and the CPE synchronization function is called to synchronize the call status. If any errors are detected in rDALB, the audit determines if the process corresponding to the rDALB is alive. If the process is alive, the audit recovers the data by requesting a single process purge of the owning process. If the rDALB does not have a living process, the audit unlinks the rDALB from the rDPB and rTEDB, and reinitializes all of the fields in the rDALB. The CPE synchronization function is called to synchronize the call status.

If the rDALB is properly linked to the rTEDB tuple, it will be placed in a transient state and an integrated services port administrator (ISPA) job schedules to recover it later, as part of customer premises equipment (CPE) resynchronization.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The first linkage to rIPCBLA tuple (dalb_lnk.lnk) is not null but the second rIPCBLA tuple linkage is zero (dadpm.st link).
Possible Error Effect:	The call is not set up correctly. Call processing messages will not be routed correctly.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Second linkage to rIPCBLA tuple.
<i>Good Data</i> =	First linkage to rIPCBLA tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The tedbflag in the rIDALB tuple does not match the tedbflag in the owning rIDPB tuple.
Possible Error Effect:	Call processing software will access the wrong data and cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	tedbflag in rIDALB tuple.
<i>Good Data</i> =	tedbflag in rIDPB tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rIDPB tuple.

Table 3 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The rIDALB tuple is supposed to be in the idle list but the linkage has an error.
Possible Error Effect:	The rIDALB tuple cannot be used for call processing.
Corrective Action Taken:	The rIDALB tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDBHEADC tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 4 Error Code: BADIS_DTE

Error Code:	BADIS_DTE
Error Description:	The user/network bit in the rIDALB tuple does not match the user/network bit in rIDSLEQUIP tuple.
Possible Error Effect:	The rIDALB tuple is not set up properly for user/network PRI's.
Corrective Action Taken:	Correct this rIDALB tuple to agree with the rIDSLEQUIP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	User/network bit in rIDALB tuple.
<i>Good Data</i> =	User/network bit in rIDSLEQUIP tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	Contents of rIDALB tuple.

Table 5 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The dalbtype in the rIDALB tuple is out of range.
Possible Error Effect:	The call processing software may not be able to properly tear down this call or activate features.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Value in dalbtype field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 6 Error Code: BRCSFLAG

Error Code:	BRCSFLAG
Error Description:	The brcsflag in rIDALB tuple does not match the brcsflag in rITEDB tuple.
Possible Error Effect:	The call processing software will not be able to properly activate BRCS features for this call.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	brcsflag in rIDALB tuple.
<i>Good Data</i> =	brcsflag in rITEDB tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rITEDB tuple.

Table 7 Error Code: BRCSLINK

Error Code:	BRCSLINK
Error Description:	The rIDALB tuple not used by ISDNBRCS should not link to RLBRCSDb data structure.
Possible Error Effect:	The system could run out of rIBRCSDb tuple data blocks.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rPCBLA tuple and rIBRCSDb tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 8 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	The rIDALB tuple not used by ISDNBRCS is not owned by a terminal process (PCBLA).
Possible Error Effect:	No terminal process controls the call.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rPCBLA tuple and rIBRCSDb tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 9 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	This DSL equipage does not exist but the rIDPB tuple does. It probably is an internal ODD problem.
Possible Error Effect:	The call is unable to receive signals from customer equipment.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDPB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 10 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	rIDALB tuple is linked to a rIDPB tuple or rITEDB tuple, but the owning process is not active.
Possible Error Effect:	Message may be sent to inactive process.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Process number of this rIDALB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 11 Error Code: DLOGCHAN

Error Code:	DLOGCHAN
Error Description:	The level 3 logical channel number for this rIDALB tuple is not valid.
Possible Error Effect:	Level 3 messages cannot be properly processed with an invalid L3 logical channel number.
Corrective Action Taken:	The audit purges the process owning rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Invalid logical channel number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 12 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The rIDALB tuple is supposed to link to rIDPB tuple but the linkage is in error.
Possible Error Effect:	The call will not handle D-channel message properly.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDPB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 13 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The logical PH for this rIDALB tuple does not match that of the D-PORT (which is populated in rITEDB tuple).
Possible Error Effect:	Signals from the switch to the customer premise equipment are misrouted or lost.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Logical PH in the rIDALB tuple.
<i>Good Data</i> =	Logical PH in the TEDB.
<i>Logical Index</i> =	Logical key of rIDALB tuple.

<i>Dump</i> =	rIDALB tuple.

Table 14 Error Code: DSL_MEM

Error Code:	DSL_MEM
Error Description:	The member number for this rIDALB tuple does not match that of the D-PORT.
Possible Error Effect:	Signals from the switch to the customer premise equipment are misrouted or lost.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Logical MEM in the rIDALB tuple.
<i>Good Data</i> =	Logical MEM in the rIDSLEQUIP tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 15 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rIDALB tuple is in the idle state but is not linked to the RLDALB idle list.
Possible Error Effect:	The rIDALB tuple is unusable by call processing.
Corrective Action Taken:	The audit idles the rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Owner link of rIDALB tuple.
<i>Good Data</i> =	Physical key of idle head cell.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 16 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	A rIDALB tuple is associated with an rIPRISTAT tuple however, the key to the rIPRISTAT tuple is incorrect.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIDALB or rIPRISTAT tuples.
Corrective Action Taken:	None by this audit the related rIPRISTAT audit will be run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDALB tuple.
<i>Bad Data</i> =	Address of the rIPRISTAT tuple (rIDALB.dadpm.st_ownk).
<i>Good Data</i> =	The flag indicating whether the rIDALB tuple is linked to the RLDPB or RLPRISTAT (rIDALB.pri_dpb_bit).
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	The rIDALB tuple that is in error.

Table 17 Error Code: MSGPTR

Error Code:	MSGPTR
Error Description:	This error code is used in two places for the following: - The rIDALB tuple is linked to rTEDB tuple and the fwd_msgs field in rIDALB tuple does not match the fwd_msgs field in rIDALB tuple. - The rIDALB tuple is not linked to rTEDB tuple but the fwd_msgs field is invalidly set to DBYES.
Possible Error Effect:	- Call processing will not be able to properly forward messages. - The rIDALB tuple is improperly set up. No call processing errors should

	be expected.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	fwd_msgs in rIDALB tuple.
<i>Good Data =</i>	1 = fwd_msgs in TEDB. 2 = DBNO.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	rTEDB tuple.

Table 18 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rIDALB tuple has a null value in the semantic linkage to rTEILA tuple but the tei and the (tedbflag or dalbtype) is in an inconsistent state.
Possible Error Effect:	The rIDALB tuple is not set up properly (maintenance rIDALB tuple or call tear down).
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 19 Error Code: PID

Error Code:	PID
Error Description:	The controlling process in the rIDALB tuple is different from that represented in the rPCBLA tuple.
Possible Error Effect:	The call will be handled by the wrong process.
Corrective Action Taken:	The audit idles the rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	PID field in the rIDALB tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 20 Error Code: PID_MISMATCH

Error Code:	PID_MISMATCH
Error Description:	A rIDALB tuple is associated with an rIPRISTAT tuple and a call activity has been requested (rIDALB.cac_req or rIDALB.kill_req = DBYES) however, there is no process ID (rIPRISTAT.send_pid) associated with the activity.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIDALB or rIPRISTAT tuples.
Corrective Action Taken:	The call activity request fields (rIDALB.cac_req and rIDALB.kill_req) are set to DBNO.
Manual Action Required:	None.
<i>Error Address =</i>	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Bad Data =</i>	The bit in the rIDALB tuple indicating that the process is being killed (rIDALB.kill_req).
<i>Good Data =</i>	The flag in the rIDALB tuple indicating that the process has a call activity check (rIDALB.cac_req).
<i>Logical Index =</i>	Logical key of rIDALB tuple (rIDALB.dalb).

<i>Dump 1 =</i>	The rIDALB tuple.
<i>Dump 2 =</i>	The rIPRISTAT tuple.

Table 21 Error Code: TEDBFLAG

Error Code:	TEDBFLAG
Error Description:	This error code is used in two places for the following: - The rIDALB tuple is linked to rTEDB tuple but the tedbflag in rIDALB tuple is set to DBNO. - The rIDALB tuple is not linked to rTEDB tuple but the tedbflag is set to DBYES.
Possible Error Effect:	In all cases the call processing software will use the wrong data causing failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	1 = DBNO. 2 = DBYES.
<i>Good Data =</i>	1 = DBYES. 2 = DBNO.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 22 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The point-to-point-back check with the particular TEDB (indicated by dalbtype) has failed.
Possible Error Effect:	The call processing software will not be able to properly activate features on this call.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	dalbtype.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	rTEDB tuple.

Table 23 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	The rIDALB tuple has a non-null value in the semantic linkage to rTEILA tuple but the point-to-point-back check with rTEILA tuple has failed.
Possible Error Effect:	The terminal may not be bound properly or the maintenance rIDALB tuple is improperly set up and will cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.

<i>Dump 2 =</i>	rTEILA tuple.

Table 24 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The OSDS ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUCactime, may have occurred.
Possible Error Effect:	The audit's timing has been interrupted. The status inquiry message for the call activity check may not be sent within the allotted time because timing is restarted.
Corrective Action Taken:	AUCactime will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the timing global variable (AUCactime).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Content of the current OSDS ITIME (OSnow).
<i>Dump 2 =</i>	Content of the transient timing variable (AUCactime).

Table 25 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The rIDALB tuple has been at transient state for too long, probably longer than audit cycle plus 5 minutes.
Possible Error Effect:	This tuple is in a stuck state. ISPA probably did not perform synchronization and release the rIDALB tuple. dalbtype reflects that it should be linked to rTEDB tuple.
Possible Error Effect:	The rIDALB tuple is not set up properly and the call processing or maintenance software may cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Dalbtype.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 26 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rIDALB tuple is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rIDALB tuple on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rIDALB tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Non-zero value of linkage.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

DALB-C

Software Release: 5E13 and later

1. D-CHANNEL APPLICATION LINKAGE BLOCK (DALB) AUDIT

The D-channel application linkage block (RLDALB) and the D-channel protocol block (RLDPB) are used to handle communications between the integrated services digital network (ISDN) periphery and the processes that control the ISDN ports. The rIDPB tuple corresponds to the D-channel port, and has one rIDALB tuple linked to it for each process that needs to communicate with the ISDN customer premises equipment (CPE). Most (but not all) of these processes control a B-port served by the D-port.

The RLDALB relation represents per call and terminal dynamic data. For every ISDN call, whether it is an active call or an associated call, a rIDALB tuple is allocated. It may be used for each ISDN call or used for distributing maintenance message. It has a semantic linkage with the terminal endpoint identifier linkage area (RLTEILA). RLDALB is linked to terminal equipment data block (RLTEDB) and RLDPB for [plain old circuit-switched ISDN call (PODS) or business and residential customer services (BRCS)]. It is also linked to primary rate interface group status (RLPRISTAT) for non-OSPS PRI calls. The *dalb_type* field in DALB indicates the purpose of DALB. The DALB audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment by the initialization, demand, and segmented modes.

2. DATA STRUCTURES

The RLDALB relation is indexed access and the audit uses the DRINASEK driver to step through the tuples. The rIDALB has two separate one-to-one linkages with protocol control block linkage area (RLPCBLA) for:

- (1) ISDN plain old circuit switching ISDN call (PODS) and business and residential customer service (BRCS), OSPS, applications processor (AP), and extended digital subscriber line (EDSL).
- (2) for ISDN Attendant.

The rIDALB shares a set-member linkage (*dadpm*) with:

RLDBHEADC	= The RLDALB idle list.
RLPRISTAT	= PRI calls.
RLDPB	= Active calls and other maintenance.

When the rIDALB tuple is not idle, the attribute *pri_dpb_bit* will identify the linkage as associated with an rIPRISTAT tuple when the attribute is DBYES, or to the rIDPB tuple when the attribute is DBNO. Refer to Figure 2.

There may be other data blocks linked to the RLDALB depending on the application, for example the supplementary protocol block (RLSPB) for operating system for distributed switching (OSDS) applications. The RLDALB relation represents per-call dynamic data. For every ISDN/OSPS/AP/EDSL call, whether it is an active call or associated call, a rIDALB tuple is allocated. Refer to Figure 1.

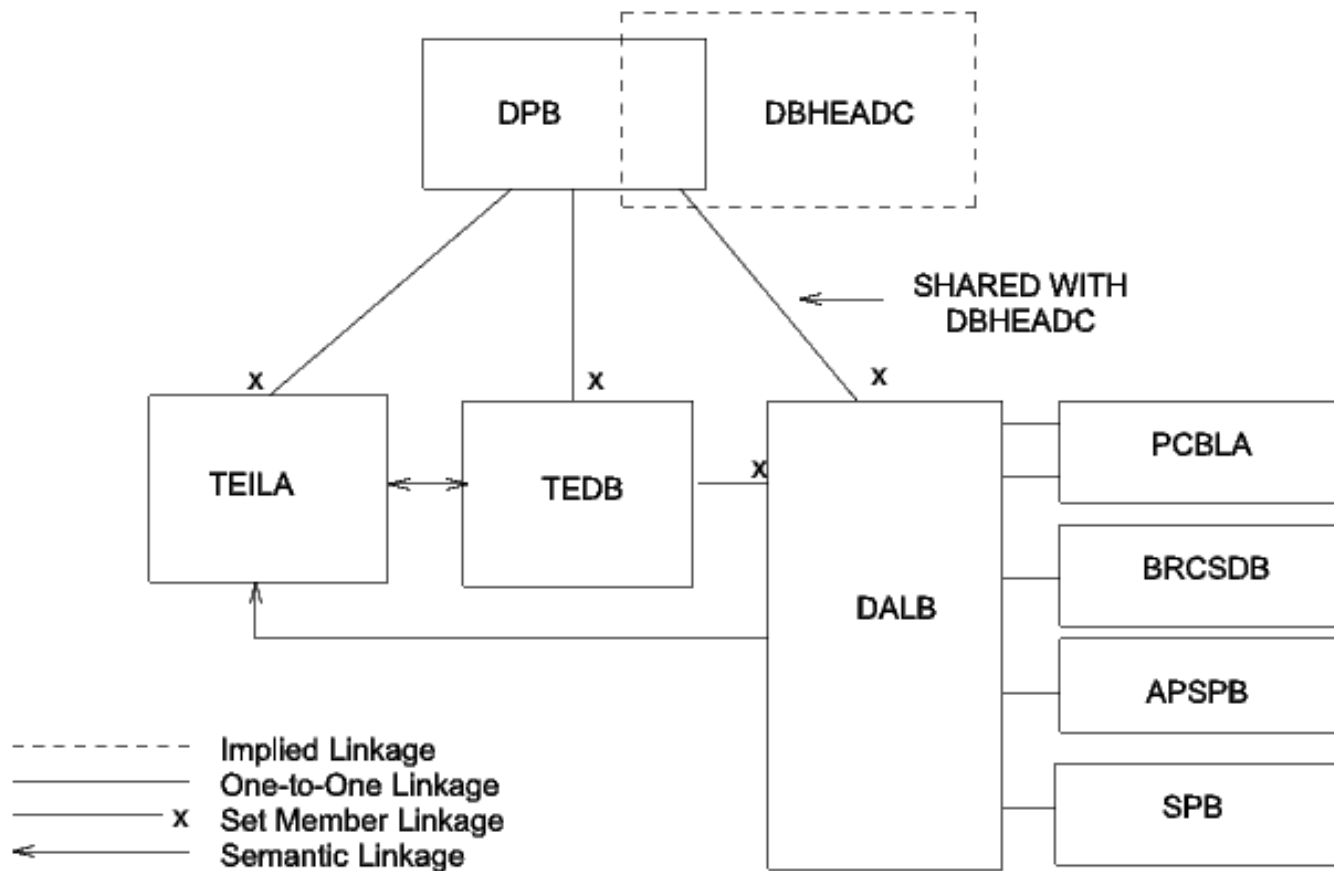


Figure 1 (DALB-1)

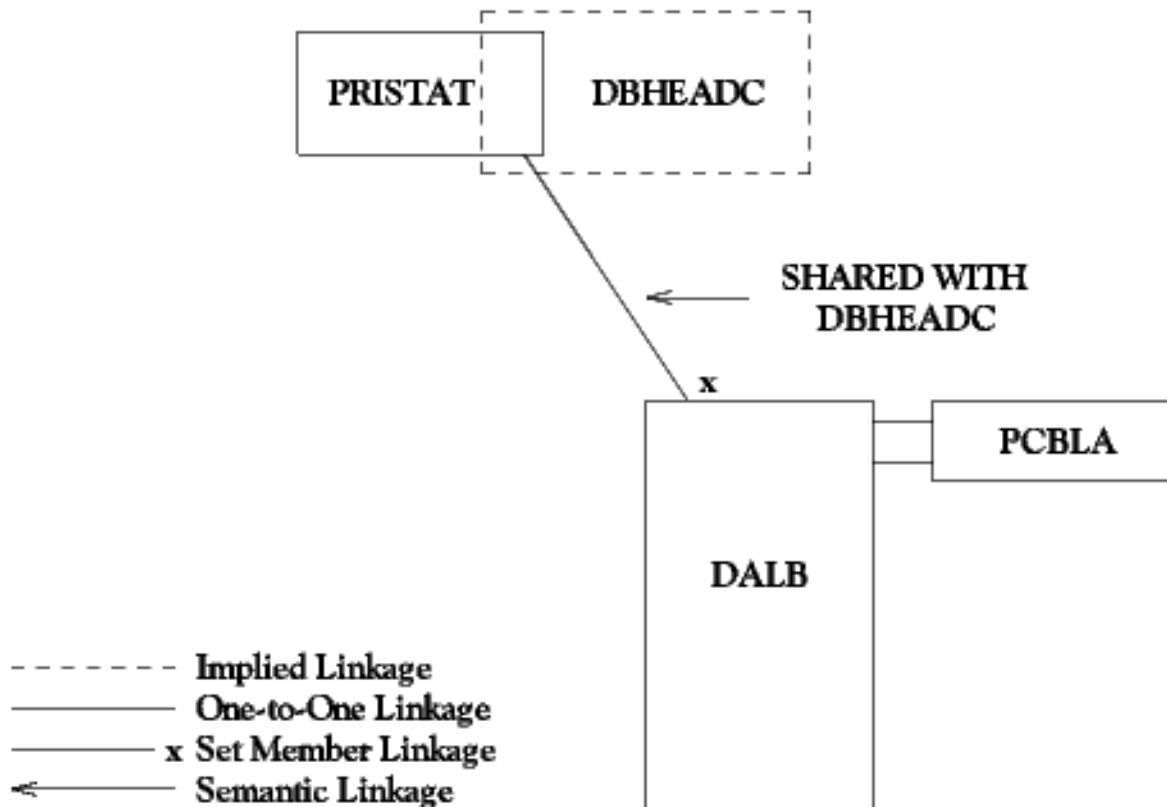


Figure 2 (DALB-2)

3. AUDIT CHECKS

- The audit first checks the quarantine field in the rIDALB tuple.
- The audit verifies that the key in the rIDALB tuple is correct.
- The audit verifies that the RDL linkages between rIDALB and other relations are correct.
- The audit determines if the rIDALB is currently active on an ISDN call by seeing if it is linked to the RLDBHEADC idle list head cell. If the rIDALB is not active, the audit does no further checking.
- The audit verifies that the process that owns an active rIDALB is a live process.
- The audit performs a transient timing check for rIDALBs that have been previously recovered and await synchronization.
- The audit verifies a possible shared call process ID in rPCBLA tuple.
- The audit checks the tedb_flag and linkages to rIBRCSDB and rPCBLA tuples.
- The audit verifies the semantic copies of static RLDSLEQUIP information.
- The audit verifies semantic data between rIDALB and rITEDB.
- During normal operation (not selective initialization) the audit interfaces with FC to perform a call activity check (CAC). This interface sends a status inquiry message to the terminal to verify the call data.

- The audit verifies the RDL linkage with rITEDB and the tedb_flag.
- The audit performs a point-to-point-back check on the semantic linkage to rITEDB based on dalb_type.
- The audit verifies semantic data between rIDALB and rIDPB tuples.
- The audit performs a point-to-point-back check on the semantic linkage to rITEILA tuples.
- There is a special recovery function to reconstruct the rIDALB.
- The audit verifies the consistency of procedures used by user-side primary rate interface (PRI). Either user-side or network-side procedures are allowed. The rIDALB tuple is checked with the rIDSLEQUIP tuple and the rIDALB tuple is corrected if an inconsistency is found.
- The audit verifies the key of the rIPRISTAT tuple (MEMBAD).
- The audit verifies the PID of the rIPRISTAT tuple when one of the call activity requests is set (PID_MISMATCH).

DALB audit is modified to check the semantic linkage with RLTEILA. The linkage with rITEDB tuple is added in DALB audit. For each type of rIDALB, a special check is performed to verify the data consistency. Every DALB error results in single process purge and CPE synchronization function is called to synchronize the call status.

4. ERROR HANDLING

Every DALB error results in a single process purge and the CPE synchronization function is called to synchronize the call status. If any errors are detected in rIDALB, the audit determines if the process corresponding to the rIDALB is alive. If the process is alive, the audit recovers the data by requesting a single process purge of the owning process. If the rIDALB does not have a living process, the audit unlinks the rIDALB from the rIDPB and rITEDB, and reinitializes all of the fields in the rIDALB. The CPE synchronization function is called to synchronize the call status.

If the rIDALB is properly linked to the rITEDB tuple, it will be placed in a transient state and an integrated services port administrator (ISPA) job schedules to recover it later, as part of customer premises equipment (CPE) resynchronization.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The first linkage to rPCBLA tuple (dalb_lnk.lnk) is not null but the second rPCBLA tuple linkage is zero (dadpm.st link).
Possible Error Effect:	The call is not set up correctly. Call processing messages will not be routed correctly.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Second linkage to rPCBLA tuple.
<i>Good Data =</i>	First linkage to rPCBLA tuple.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The tedbflag in the rIDALB tuple does not match the tedbflag in the owning rIDPB tuple.
Possible Error Effect:	Call processing software will access the wrong data and cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	tedbflag in rIDALB tuple.
<i>Good Data</i> =	tedbflag in rIDPB tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rIDPB tuple.

Table 3 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The rIDALB tuple is supposed to be in the idle list but the linkage has an error.
Possible Error Effect:	The rIDALB tuple cannot be used for call processing.
Corrective Action Taken:	The rIDALB tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDBHEADC tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 4 Error Code: BADIS_DTE

Error Code:	BADIS_DTE
Error Description:	The user/network bit in the rIDALB tuple does not match the user/network bit in rIDSLEQUIP tuple.
Possible Error Effect:	The rIDALB tuple is not set up properly for user/network PRI's.
Corrective Action Taken:	Correct this rIDALB tuple to agree with the rIDSLEQUIP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	User/network bit in rIDALB tuple.
<i>Good Data</i> =	User/network bit in rIDSLEQUIP tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	Contents of rIDALB tuple.

Table 5 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The dalbtype in the rIDALB tuple is out of range.
Possible Error Effect:	The call processing software may not be able to properly tear down this call or activate features.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Value in dalbtype field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 6 Error Code: BRCSFLAG

Error Code:	BRCSFLAG
Error Description:	The brcsflag in rIDALB tuple does not match the brcsflag in rITEDB tuple.
Possible Error Effect:	The call processing software will not be able to properly activate BRCS features for this call.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	brcsflag in rIDALB tuple.
<i>Good Data</i> =	brcsflag in rITEDB tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rITEDB tuple.

Table 7 Error Code: BRCSLINK

Error Code:	BRCSLINK
Error Description:	The rIDALB tuple not used by ISDNBRCS should not link to RLBRCSDb data structure.
Possible Error Effect:	The system could run out of rIBRCSDb tuple data blocks.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rPCBLA tuple and rIBRCSDb tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 8 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	The rIDALB tuple not used by ISDNBRCS is not owned by a terminal process (PCBLA).
Possible Error Effect:	No terminal process controls the call.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rPCBLA tuple and rIBRCSDb tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 9 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	This DSL equipage does not exist but the rIDPB tuple does. It probably is an internal ODD problem.
Possible Error Effect:	The call is unable to receive signals from customer equipment.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDPB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 10 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	rIDALB tuple is linked to a rIDPB tuple or rITEDB tuple, but the owning process is not active.
Possible Error Effect:	Message may be sent to inactive process.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Process number of this rIDALB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.

<i>Dump</i> =	rIDALB tuple.

Table 11 Error Code: DLOGCHAN

Error Code:	DLOGCHAN
Error Description:	The level 3 logical channel number for this rIDALB tuple is not valid.
Possible Error Effect:	Level 3 messages cannot be properly processed with an invalid L3 logical channel number.
Corrective Action Taken:	The audit purges the process owning rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Invalid logical channel number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 12 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The rIDALB tuple is supposed to link to rIDPB tuple but the linkage is in error.
Possible Error Effect:	The call will not handle D-channel message properly.
Corrective Action Taken:	The audit idles the rIDALB tuple and purges the associated process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Linkage fields to rIDPB tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 13 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The logical PH for this rIDALB tuple does not match that of the D-PORT (which is populated in rITEDB tuple).
Possible Error Effect:	Signals from the switch to the customer premise equipment are misrouted or lost.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Logical PH in the rIDALB tuple.
<i>Good Data</i> =	Logical PH in the TEDB.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 14 Error Code: DSL_MEM

Error Code:	DSL_MEM
Error Description:	The member number for this rIDALB tuple does not match that of the D-PORT.
Possible Error Effect:	Signals from the switch to the customer premise equipment are misrouted or lost.
Corrective Action Taken:	Process owning rIDALB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Logical MEM in the rIDALB tuple.
<i>Good Data</i> =	Logical MEM in the rDSLEQUIP tuple.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 15 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rIDALB tuple is in the idle state but is not linked to the RLDALB idle list.

Possible Error Effect:	The rIDALB tuple is unusable by call processing.
Corrective Action Taken:	The audit idles the rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	Owner link of rIDALB tuple.
<i>Good Data =</i>	Physical key of idle head cell.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 16 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	A rIDALB tuple is associated with an rIPRISTAT tuple however, the key to the rIPRISTAT tuple is incorrect.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIDALB or rIPRISTAT tuples.
Corrective Action Taken:	None by this audit the related rIPRISTAT audit will be run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDALB tuple.
<i>Bad Data =</i>	Address of the rIPRISTAT tuple (rIDALB.dadpm.st_ownk).
<i>Good Data =</i>	The flag indicating whether the rIDALB tuple is linked to the RLDPB or RLPRISTAT (rIDALB.pri_dpb_bit).
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	The rIDALB tuple that is in error.

Table 17 Error Code: MSGPTR

Error Code:	MSGPTR
Error Description:	This error code is used in two places for the following: - The rIDALB tuple is linked to rTEDB tuple and the fwd_msgs field in rIDALB tuple does not match the fwd_msgs field in rIDALB tuple. - The rIDALB tuple is not linked to rTEDB tuple but the fwd_msgs field is invalidly set to DBYES.
Possible Error Effect:	- Call processing will not be able to properly forward messages. - The rIDALB tuple is improperly set up. No call processing errors should be expected.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	fwd_msgs in rIDALB tuple.
<i>Good Data =</i>	1 = fwd_msgs in TEDB. 2 = DBNO.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump 1 =</i>	rIDALB tuple.
<i>Dump 2 =</i>	rTEDB tuple.

Table 18 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rIDALB tuple has a null value in the semantic linkage to rTEILA tuple but the tei and the (tedbflag or dalbtype) is in an inconsistent state.

Possible Error Effect:	The rIDALB tuple is not set up properly (maintenance rIDALB tuple or call tear down).
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 19 Error Code: PID

Error Code:	PID
Error Description:	The controlling process in the rIDALB tuple is different from that represented in the rPCBLA tuple.
Possible Error Effect:	The call will be handled by the wrong process.
Corrective Action Taken:	The audit idles the rIDALB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	PID field in the rIDALB tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rIDALB tuple.
<i>Dump =</i>	rIDALB tuple.

Table 20 Error Code: PID_MISMATCH

Error Code:	PID_MISMATCH
Error Description:	A rIDALB tuple is associated with an rIPRISTAT tuple and a call activity has been requested (rIDALB.cac_req or rIDALB.kill_req = DBYES) however, there is no process ID (rIPRISTAT.send_pid) associated with the activity.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIDALB or rIPRISTAT tuples.
Corrective Action Taken:	The call activity request fields (rIDALB.cac_req and rIDALB.kill_req) are set to DBNO.
Manual Action Required:	None.
<i>Error Address =</i>	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Bad Data =</i>	The bit in the rIDALB tuple indicating that the process is being killed (rIDALB.kill_req).
<i>Good Data =</i>	The flag in the rIDALB tuple indicating that the process has a call activity check (rIDALB.cac_req).
<i>Logical Index =</i>	Logical key of rIDALB tuple (rIDALB.dalb).
<i>Dump 1 =</i>	The rIDALB tuple.
<i>Dump 2 =</i>	The rIPRISTAT tuple.

Table 21 Error Code: TEDBFLAG

Error Code:	TEDBFLAG
Error Description:	This error code is used in two places for the following: - The rIDALB tuple is linked to rTEDB tuple but the tedbflag in rIDALB tuple is set to DBNO. - The rIDALB tuple is not linked to rTEDB tuple but the tedbflag is set to DBYES.
Possible Error Effect:	In all cases the call processing software will use the wrong data causing failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDALB tuple.
<i>Bad Data =</i>	1 = DBNO.

	2 = DBYES.
<i>Good Data</i> =	1 = DBYES. 2 = DBNO.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 22 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The point-to-point-back check with the particular TEDB (indicated by dalbtype) has failed.
Possible Error Effect:	The call processing software will not be able to properly activate features on this call.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	dalbtype.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rTEDB tuple.

Table 23 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	The rIDALB tuple has a non-null value in the semantic linkage to rTEILA tuple but the point-to-point-back check with rTEILA tuple has failed.
Possible Error Effect:	The terminal may not be bound properly or the maintenance rIDALB tuple is improperly set up and will cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump 1</i> =	rIDALB tuple.
<i>Dump 2</i> =	rTEILA tuple.

Table 24 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The OSDS ITIME (OSnow) seems to have moved backwards. Possible corruption of the timer, rIDALB.cacaud_time, may have occurred.
Possible Error Effect:	The audit's timing has been interrupted. The status inquiry message for the call activity check may not be sent within the allotted time because timing is restarted.
Corrective Action Taken:	The timer, rIDALB.cacaud_time, will be updated with the current OSDS ITIME (OSnow), and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	High-order value in the timestamp (rIDALB.cacaud_time.hi_time).
<i>Good Data</i> =	Low-order value in the timestamp (rIDALB.cacaud_time.lo_time).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Content of the current OSDS ITIME (OSnow).

Table 25 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The rIDALB tuple has been at transient state for too long, probably longer than audit cycle plus 5 minutes.
Possible Error Effect:	This tuple is in a stuck state. ISPA probably did not perform synchronization and release the rIDALB tuple. dalbtype reflects that it should be linked to rTEDB tuple.
Possible Error Effect:	The rIDALB tuple is not set up properly and the call processing or maintenance software may cause failures.
Corrective Action Taken:	Recover this rIDALB tuple to the idle list and resync the terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Dalbtype.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

Table 26 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rIDALB tuple is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rIDALB tuple on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rIDALB tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDALB tuple.
<i>Bad Data</i> =	Non-zero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIDALB tuple.
<i>Dump</i> =	rIDALB tuple.

DASC

Software Release: 5E10 and later

1. DIRECTORY ASSISTANCE SYSTEM/COMPUTER (DASC) AUDIT

Design alternatives for operator actions (OA) data included using the stack or creating a new area or areas for them and linking them from the position data block. The directory assistance system/computer relation (RLDASC) checks the mapping between the Operator Services Position System (OSPS) DAS/C call ID and the process number of the running position terminal process (PTP) or announcement terminal process (ATP). Verifying the validity of the mapping will ensure the accuracy of message routing. The DASC audit is part of the OSPS audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It runs in the OSDS-M environment in the segmented mode only.

2. DATA STRUCTURES

The DASC audit uses three dynamic relations, RLDASC, RLLOOPDB, and RLPOSDB. The RLPOSDB relation links DASC to the dynamic relation RLLOOPDB. The DASCs call ID is verified against the data from the LOOPDB. This relation contains process IDs and call IDs, as Figure 1 shows.

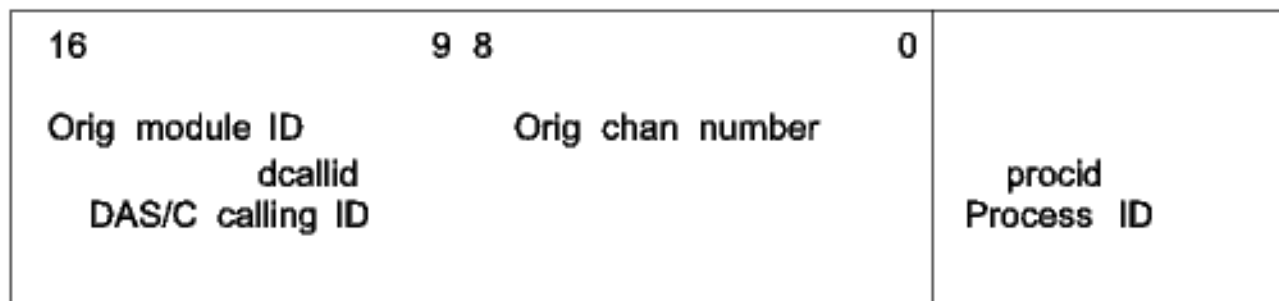


Figure 1 (DASC-1) RLDASC Relation

3. AUDIT CHECKS

The DASC audit invokes the static driver (DRSTATIC) to retrieve tuples of RLDASC. It verifies the process number and call ID by comparing them to information retrieved by the call ID process.

4. ERROR HANDLING

The audit deletes erroneous tuples from RLDASC. The DASC audits schedules no related audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	The process is not valid.
Possible Error Effect:	Possible problems caused by a resource failure.
Corrective Action Taken:	The rIDASC tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDASC.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Key of rIDASC.
<i>Dump</i> =	rIDASC tuple.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	The call ID is not valid.
Possible Error Effect:	Possible problems caused by resource failure.
Corrective Action Taken:	The rIDASC tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDASC.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDASC.
<i>Dump</i> =	rIDASC tuple.

DBCND

Software Release: 5E10 and later

1. INDIVIDUAL CALLING LINE IDENTIFICATION (ICLID) CALLING NUMBER DELIVERY TEMPORARY STORAGE BLOCK (DBCND) AUDIT

The ICLID calling number delivery temporary storage data block is used as a temporary storage structure to store the last incoming calling directory number (DN) line interface circuit directory number (LICDN). By storing the LICDN, the terminal that has the ICLID CND and performs a pickup is able to display the number of the calling party. The DBCND audit verifies internal consistency of the RLDBCND relation by performing linkage and semantic checks. This audit runs segmented in the operating system for distributed switching-switching module (OSDS-SM) environment.

2. DATA STRUCTURES

The RLDBCND relation is a dynamic indirect access sequential keyed relation. When a tuple of the RLDBCND relation is idle, it is owned by a tuple in the RLDB_HEADC relation. When a tuple is busy, it is linked to an active RLPCBLA tuple. Refer to Figure 1.

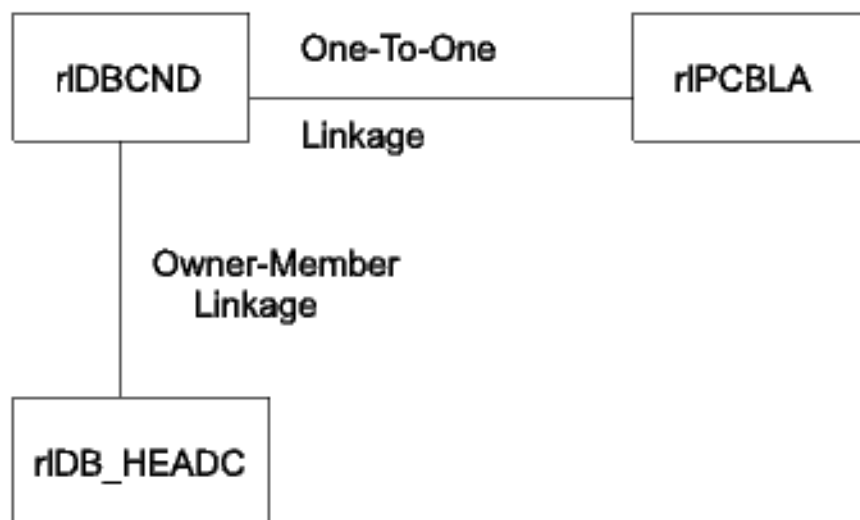


Figure 1 (DBCND-1) DBCND Relations

3. AUDIT CHECKS

The DBCND audit steps through the RLDBCND relation and uses the audit driver to verify the key and linkage of a rDBCND tuple. If those tests pass, the audit's semantic function is called to verify internal data.

4. ERROR HANDLING

If a rDBCND tuple is found in error, it is initialized and linked back to the appropriate idle list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rIDBCND tuple being audited should be on a busy list (linked to rIPCBLA). This error will fire when the owner portion of the linkage field indicates this tuple should be linked to the busy list while another portion of the linkage field indicates that it should be linked to a rIDBHEADC (on a idle list).
Possible Error Effect:	LASS incoming line identification data may be incorrect for a call in the ringing state.
Corrective Action Taken:	The rIDBCND tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDBCND tuple.
<i>Bad Data =</i>	t link portion of linkage field; used to show linkage to a rIPCBLA.
<i>Good Data =</i>	Owner portion of the linkage field of the rIDBCND tuple being audited.
<i>Logical Index =</i>	Key of rIDBCND tuple.
<i>Dump =</i>	rIDBCND tuple being audited.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rIDBCND tuple being audited is indicating that it is not owned by the idle list or a valid busy list.
Possible Error Effect:	LASS incoming line identification data may be incorrect for a call in the ringing state.
Corrective Action Taken:	The rIDBCND tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDBCND tuple.
<i>Bad Data =</i>	st_ownd portion of linkage field; used to show the owner of the rIDBCND.
<i>Good Data =</i>	Value of the idle list owner.
<i>Logical Index =</i>	Key of rIDBCND tuple.
<i>Dump =</i>	rIDBCND tuple being audited.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIDBCND tuple being audited should be on an idle list (linked to rIDBHEADC). This error will fire when the owner portion of the linkage field indicates this tuple should be linked to the idle list while another portion of the linkage field indicates that it should be linked to a rIPCBLA (on a busy list).
Possible Error Effect:	LASS incoming line identification data may be incorrect for a call in the ringing state.
Corrective Action Taken:	The rIDBCND tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDBCND tuple.
<i>Bad Data =</i>	Return from linkage check function 1 indicates the linkage fields are incorrect.
<i>Good Data =</i>	Owner portion of the linkage field of the rIDBCND tuple being audited.
<i>Logical Index =</i>	Key of rIDBCND tuple.
<i>Dump =</i>	rIDBCND tuple being audited.

DBCSTMSG

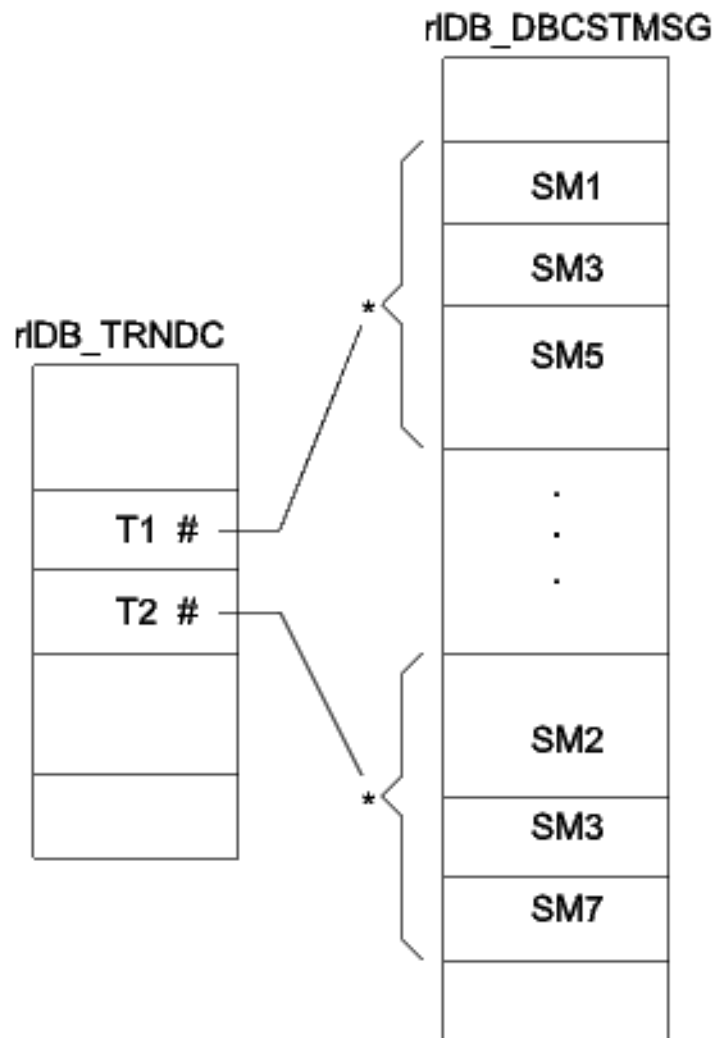
Software Release: 5E10 and later

1. DATABASE BROADCAST MESSAGE DICTIONARY (DBCSTMSG) AUDIT

The database manager program ensures that the slave processes associated with partitioned transactions update data in the SMs properly. If a transaction processes data that exists in more than one switching module, all the related SM slave process IDs are linked together in the OSDS process ID (OSPID) list dictionary (RLDB_PIDL). The list dictionary resides in the administrative module and contains entries available for use by each switching module and transaction. The OSPID list dictionary audit (PIDL) checks the linkages of the list dictionary array and ensures that no OSPID list dictionary entries are lost. This audit is part of the database audits subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The OSPID list dictionary consists of tuples that are used for different transactions. There is a set-member linkage between each tuple of the list dictionary and the transaction dictionary. There are forward and backward pointers between the tuples of the OSPID list dictionary. Refer to Figure 1 .



————— * = Owner-member Linkage

Figure 1 (DBCSTMSG-1) Relations of the DBCSTMSG Audit

3. AUDIT CHECKS

The DBCSTMSG audit calls the linkage driver (DRLINK) to check the linkages between the OSPID list dictionary array and the transaction dictionary. The audit performs no semantic checks.

4. ERROR HANDLING

When the audit detects an error, the lost OSPID list dictionary entries are recovered by aborting the owning transaction and reclaiming the related resources (open dictionaries, head tables, and data pages). See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

DBHEADC

Software Release: 5E10 and later

1. DATABASE HEAD CELLS (DBHEADC) AUDIT

The database head cells relation (RLDB_HEADC) consists solely of head cells used as owners of tuples in various other relations. The relations owned by RLDB_HEADC are listed in Table 1.

Table 1 Relations Owned by RLDB_HEADC

Relation ID	Relation Name
RLADB	Analog data block
RLAPSPB	Applications processor supplementary protocol block
RLATPDB	ATP call and announcement information
RLATTH_BLF	Attendant 1005 group busy lamp field
RLAUTOQ	Automatic queue
RLBRCSDB	BRCS data block
RLCBDB	Call buildup data block
RLCOINOT	OSPS COIN originating treatment
RLCQB	Circuit queuing block
RLCV_STATE	Call vectoring state
RLDALB	D-channel application linkage block
RLDB_TRNDC	Data base transaction dictionary
RLDBCND	ICLID calling number delivery
RLDISPDB	Display data block
RLDLTSBDB	Line time slot bridging data block
RLDOC	Information per individual trunk group DOC control
RLEXRR	Extended reroute
RLGRP_TGC	Group trunk group controls
RLINS_TN	Number services termination notification data
RLISPABUF	Integrated services port administrator buffer
RLLOOPDB	Call and display information at an operator position
RLMODH_BLF	Switching module link header busy lamp field
RLMSQ	Output message queue
RLNS_IDX	Network services indexing information
RLNSDB	Network services data block
RLOSBCB	Broadcast control buffer
RLOSBIT	Broadcast identification table
RLPHISTORY	Pending history
RLPSUUI	Extended digital subscriber line user to user information
RLREPLYBLF	Busy lamp field report
RLSCMDB	Shared call model data block
RLSMQLIST	List of circuits in queue
RLSMXRF	Switch maintenance cross-reference table
RLSPB	Supplemental protocol block
RLSRVOT	Listing services call data for OT and EIS interface
RLTALoopDB	Call and display information at a operator position loop
RLTEILA	Terminal endpoint identifier linkage area
RLTGC	Trunk group controls
RLTR	Individual trunk information

The database head cells audit (DBHEADC) verifies the linkages between rldb_headc head cells and the tuples owned by them. This audit is part of the database audits subgroup.

This audit runs in the operational kernel process (OKP) in the administrative module (AM), in the operational system for distributed switching-switching module (SM), and in the communications module processor (CMP) environments in the segmented mode.

2. DATA STRUCTURES

The DBHEADC audit insures the integrity of the database headcell relation (RLDB_HEADC). The RLDB_HEADC relation owns tuples in the relations listed in Table 1.

3. AUDIT CHECKS

The DBHEADC audit calls the linkage driver (DRLINK) to perform the key and quarantine bit checks and then call the semantic function to check the linkages between the head cell and each of the relations listed in Table 1. The linkage checking is done in the DBHEADC audit's semantic check function because of the many shared mappings to other relations. This function gets mapping information from the DAhctbl[] array. DAhctbl[] is indexed with the logical key of the rIDB_HEADC tuple and contains the relation identification of the member tuple, whether the member tuple is in the administrative module (AM), switching module (SM), communication module processor (CMP), or any combination of modules, and the dynamic access (DA) subsystem set mapping. The primitive AUowner() is invoked, passing on the parameters of tuple, dope, and per mapping information. AUowner() performs point-to-point-back checking between rIDB_HEADC (owner) and the member tuple.

4. ERROR HANDLING

When the DBHEADC audit detects an error, the rIDB_HEADC tuple is zeroed and reconstructed to an idle state. The rIDB_HEADC tuple linkage is later restored by audits operating on the owned tuples. Errors are reported by means of output messages.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 2 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	The relation ID in the rIDB_HEADC tuple does not match the DAhctbl entry.
Possible Error Effect:	SM or TM code loses access to relation.
Corrective Action Taken:	The audit recovers and initializes the rIDB_HEADC tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_HEADC tuple.
<i>Bad Data</i> =	Relation ID in the rIDB_HEADC tuple.
<i>Good Data</i> =	Relation ID from DAhctbl entry.
<i>Logical Index</i> =	Logical key of rIDB_HEADC tuple.
<i>Dump</i> =	rIDB_HEADC tuple.

Table 3 Error Code: NOTIDLE2

Error Code:	NOTIDLE2
Error Description:	An unused rIDB_HEADC tuple did not have the head and tail pointers set to null.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers and initializes the rIDB_HEADC tuple. The head and tail fields will be reset to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_HEADC tuple.
<i>Bad Data</i> =	Head pointer.
<i>Good Data</i> =	Tail pointer.
<i>Logical Index</i> =	Logical key of the rIDB_HEADC tuple.
<i>Dump</i> =	rIDB_HEADC tuple.

Table 4 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Point-to-point-back checking failed between one rIDB_HEADC tuple and one of its member relations. The relation identification of the member relation in error is found in the bad data dump. If available the head and tail linkages will be dumped. <i>Dump 2</i> will be the head or the tail. <i>Dump 3</i> will always be the tail.
Possible Error Effect:	Loss of resources to the system.
Corrective Action Taken:	The head cell will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rIDB_HEADC tuple.
<i>Bad Data</i> =	Relation identification of member relation (DAhctbl[].relid).
<i>Good Data</i> =	Mapping identification (DAhctbl[].mapid).
<i>Logical Index</i> =	Logical key of the db_headc.
<i>Dump 1</i> =	rIDB_HEADC tuple.
<i>Dump 2</i> =	Member tuple at head or tail of list.
<i>Dump 3</i> =	Member tuple at tail of list.

DBLOCK

Software Release: 5E10 and later

1. DATABASE LOCK DICTIONARY (DBLOCK) AUDIT

The database lock dictionary, a data structure used by the database manager, provides control for relations needed by a transaction. For additional information see the audit descriptions for the TRNDC, OPNDC, and RELQ audits. The database manager program provides concurrency among transactions through multi-level locking (MLL) which includes relation level locking (RLL) and tuple level locking (TLL). Read-only transactions are not queued. Update transactions are queued if they have to access the same relations as other update transactions in RLL; or, the same tuples in TLL. When a transaction starts, the database manager creates an open dictionary entry for each relation needed by that transaction under RLL. The database manager also creates a database lock dictionary entry for each relation needed by that transaction. The open dictionary entry and the database lock dictionary entry, associated with the same relation and the same transaction, are linked together with a one-to-one link. While the transaction is in progress, if it needs another relation, the database manager creates a database lock dictionary entry for that relation under TLL. In this case, there is no open dictionary entry created. The database lock dictionary entries are stored in a list for the needed relation. All database lock dictionary lists are owned by the relation queue dictionary. The database lock dictionary audit (DBLOCK) ensures that the database lock dictionary entries are not lost when errors occur. The DBLOCK audit is part of the database audits subgroup. This audit runs in the operational kernel process (OKP) in the administrative module (AM), in the operational system for distributed switching-switching module (SM), and in the communications module processor (CMP) environments in the segmented modes.

2. DATA STRUCTURES

The system allocates one database lock dictionary block for each relation (under RLL or TLL) associated with a database transaction. When two or more transactions require access to the same relation, the database lock dictionary blocks are put in a list. The owner of the list is a tuple in RIDB_RELQ. Figure 1 shows that blocks 1 and 4 need access to rl2; blocks 2, 5, and 7 need access to rl8; blocks 3 and 6 need access to rl15. It also shows that blocks 2, 4, and 7 are under RLL, which have associated open dictionary entries (rlDB_OPNDC); and, blocks 1, 3, 5 and 6 are under TLL.

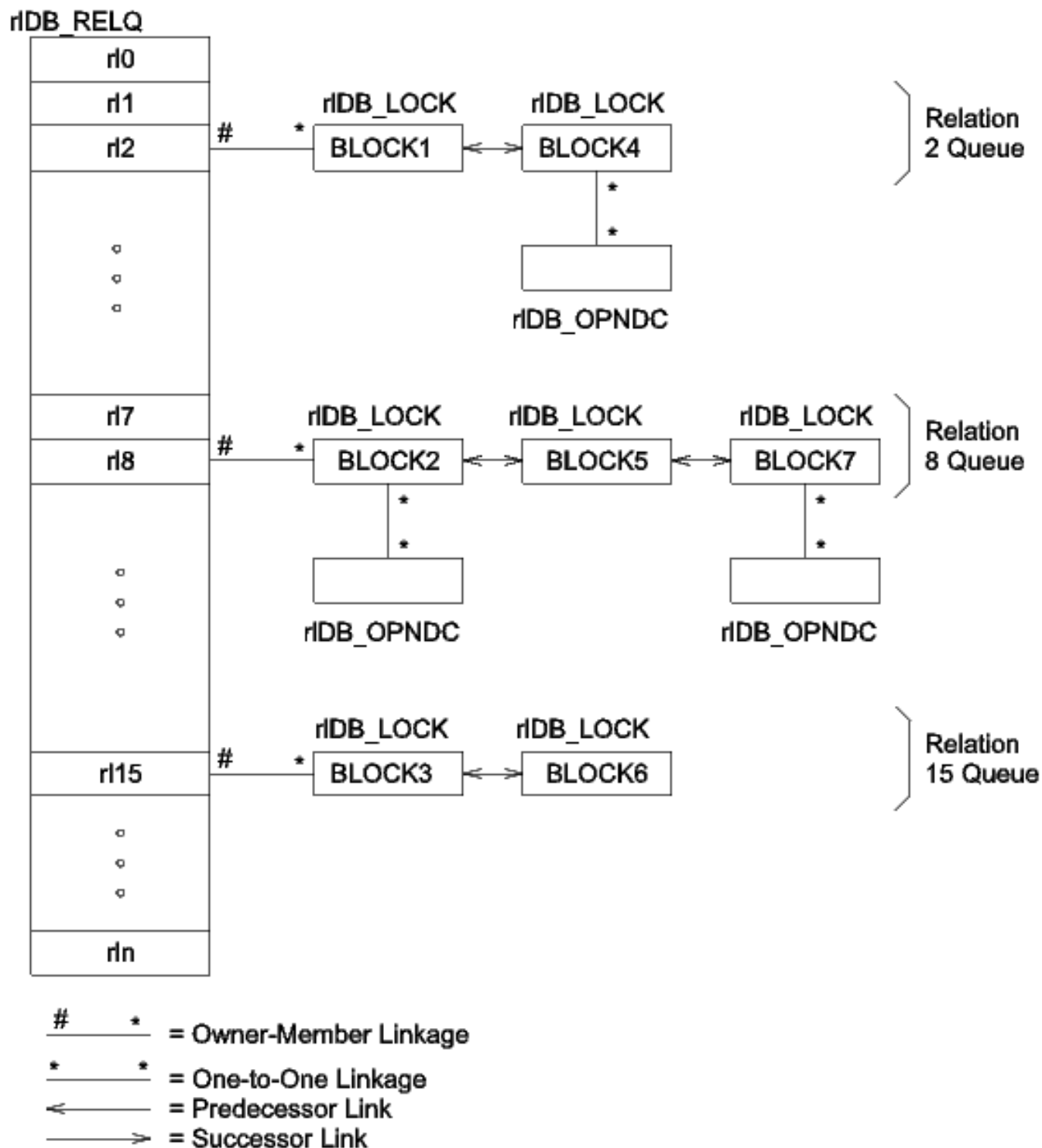


Figure 1 (DBLOCK-1) Relation Linkages

3. AUDIT CHECKS

The DBLOCK audit calls the linkage driver (DRLINK) to check the linkages between the tuples of the relation queue dictionary and the data base lock dictionary entries that are owned by them. The audit then performs the following semantic checks on database lock dictionary entries.

- (a) If a database lock dictionary entry is active (such as, being used by a transaction), it checks that the transaction must be either in a busy queue or in a cleanup queue; it also checks that the data base lock dictionary is owned by a relation queue entry.
- (b) If a database lock dictionary entry is idle, it checks that the database lock dictionary entry is in the idle queue; it also checks that the link field with the open dictionary is zero.

4. ERROR HANDLING

The audit takes one of the following actions when it detects an error. If it detects a broken linkage, it aborts all the transactions and reclaims the lost resources. If the semantic check fails, only the owning transaction is aborted and the related resources are reclaimed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: DBLOCKIDLE

Error Code:	DBLOCKIDLE
Error Description:	The non-zero transaction ID indicated in the rIDB_LOCK tuple corresponds to an active transaction. However this tuple is linked to an idle list.
Possible Error Effect:	Another transaction could access this tuple and corrupt the transaction.
Corrective Action Taken:	The rIDB_LOCK tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_LOCK tuple.
<i>Bad Data =</i>	The transaction ID.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIDB_LOCK.
<i>Dump =</i>	The rIDB_LOCK tuple.

Table 2 Error Code: DBLOCKZCHK

Error Code:	DBLOCKZCHK
Error Description:	A rIDB_LOCK tuple, which is not used by any transaction, is still linked to some transaction.
Possible Error Effect:	Transaction may not complete correctly or the rIDB_LOCK tuple has been corrupted.
Corrective Action Taken:	The rIDB_LOCK tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_LOCK tuple.
<i>Bad Data =</i>	The key to the data structure still linked to this rIDB_LOCK tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIDB_LOCK.
<i>Dump =</i>	The rIDB_LOCK tuple.

Table 3 Error Code: DBOWNERBUSY

Error Code:	DBOWNERBUSY
Error Description:	A rIDB_LOCK entry, which is not used by any transaction, is not in the idle list.
Possible Error Effect:	Transaction did not complete correctly or the rIDB_LOCK tuple has been corrupted.
Corrective Action Taken:	The rIDB_LOCK tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_LOCK tuple.
<i>Bad Data =</i>	The rIDB_LOCK owner.

<i>Good Data</i> =	The correct idle rIDB_LOCK owner.
<i>Logical Index</i> =	Key of rIDB_LOCK tuple.
<i>Dump</i> =	The rIDB_LOCK tuple.

Table 4 Error Code: DBOWNERIDLE

Error Code:	DBOWNERIDLE
Error Description:	A rIDB_LOCK entry is being used by a transaction but it is not linked to the relation queue. Therefore, this transaction is not locked to any relation.
Possible Error Effect:	Transaction did not complete correctly or the rIDB_LOCK tuple has been corrupted.
Corrective Action Taken:	The rIDB_LOCK tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_LOCK tuple.
<i>Bad Data</i> =	The rRELQ key.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDB_LOCK tuple.
<i>Dump</i> =	The rIDB_LOCK tuple.

DBSMSTAT

Software Release: 5E10 and later

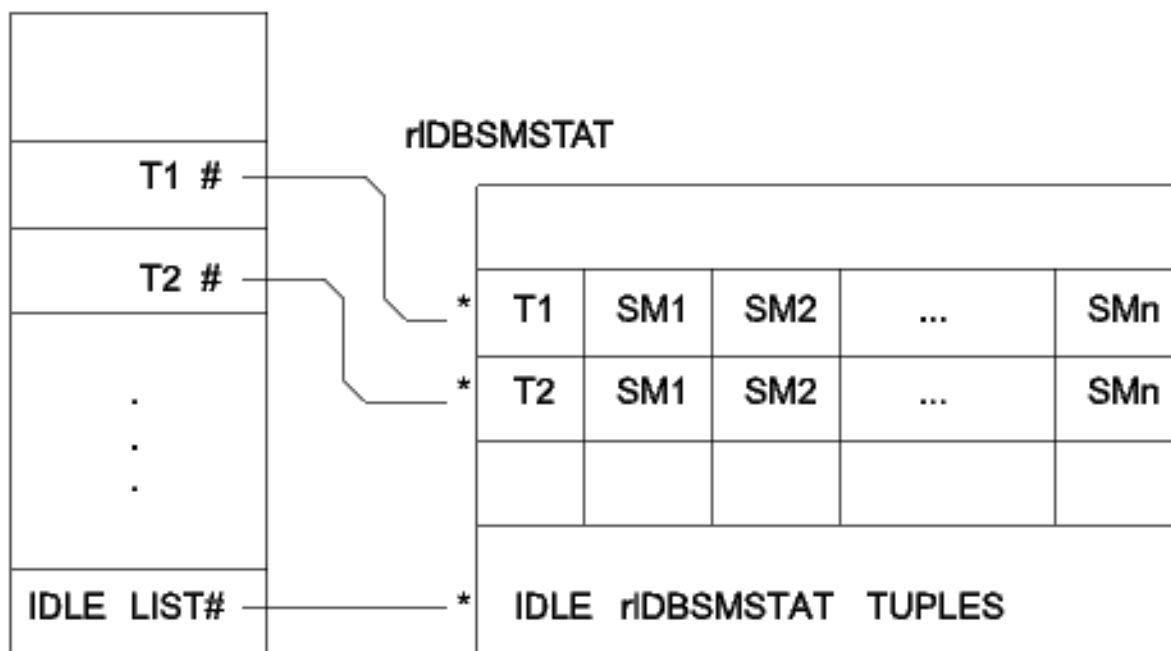
1. DATABASE SM STATUS ARRAY (DBSMSTAT) AUDIT

The database manager program ensures that each transaction involving redundant data has the same effect on all the SMs. If the information in one copy of an SMredundant relation is updated, the same information in all the SMs must be updated. To accomplish this, the database subsystem contains a dictionary of all the process IDs of the SM slave processes associated with each redundant transaction. The database SM status array audit (DBSMSTAT) checks this array to reclaim lost entries. This audit is part of the database audits subgroup. It runs in the CMP and operating system for distributed switching-switching module (OSDS-SM) environments in the segmented modes.

2. DATA STRUCTURES

The array (RLDBSMSTAT) consists of tuples that are used by redundant transactions. For each active, redundant transaction, each tuple of RLDBSMSTAT is a singleton in an owner-member linkage to the transaction dictionary (RLDB_TRNDC). One redundant transaction therefore owns only one array entry. As Figure 1 shows, each tuple contains a field designating the processor to be read when updating the redundant data for the transaction. Unused tuples are stored on an idle list that is owned by the transaction dictionary.

rlDB_TRNDC



T1 = TRANSACTION 1

T2 = TRANSACTION 2

————— * = Owner-Member Linkage

Figure 1 (DBSMSTAT-1) Relations of the DBSMSTAT Audit

3. AUDIT CHECKS

The DBSMSTAT audit calls the linkage driver (DRLINK) to check the linkages between the array and the transaction dictionary. The audit performs no semantic checks.

4. ERROR HANDLING

When the audit detects an error, the owning transaction is aborted and the related resources (open dictionaries, head tables, and data pages) are reclaimed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

DCCB-A

Software Release: 5E10 - 5E11

1. D-CHANNEL CONTROL BLOCK (DCCB) AUDIT

The DCCB audit verifies data associated with D-channel controls resident in the PH. The audit verifies information about each of the fields of the DCCB, and about each terminal end point identifier (TEIB) in it during idle and busy (in use) DCCB. If there is an error in one of the TEIBs, only that TEIB is corrected. Calls on other TEIBs are not affected. If there is a problem with any other field in the DCCB, the whole DCCB is cleared, including all 8 TEIBs. Head cell checks for LCCB queue for conditional notification are also added here. The flow control control block (FCCB) logical link control blocks (LLCB) are used to control the flood of embedded operational channel (EOC) maintenance messages that may result when one or more remote terminals (RT) are powered up. The RTs referred here are the TR303 RTs associated with the integrated digital carrier unit (IDCU). Checks related to FCCB LLCBs are also included in this audit. The DCCB audit runs in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DCCB audit steps through the PSdccb[] array. It interfaces with the D-channel data block (DCDB), the LLCB, the TEIB and the access line control block (ALCB) data structures. If the DCCB index number is divisible by four and if the PH channel type is PS_EOC, an FCCB LLCB is also used that can be associated with this DCCB.

3. AUDIT CHECKS

The audit checks the linkages in PSdccb with the logical link control block (PSllcb).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_VC_BAD

Error Code:	ACT_VC_BAD
Error Description:	The value in the act_vc field of DCCB does not match the computer value of the same by audits.
Possible Error Effect:	No immediate effect on the operation of the switch. But this count is used for camp-on and for "blocking the channel if it is idle". It can give different results if this count is not correct.
Corrective Action Taken:	The audit will report the error and correct the value of the act_vc field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCCB in which this error was found.
<i>Bad Data =</i>	Value in act_vc field in DCCB.
<i>Good Data =</i>	Value in active_vc field as calculated by audit.
<i>Logical Index =</i>	DCCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: ALCBIND

Error Code:	ALCBIND
Error Description:	The alcb_index field of the management LAPD LLCB is not idle which it should be. You cannot have an ALCB for a management call.
Possible Error Effect:	The alcb_index value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.

<i>Bad Data</i> =	Value in alcb_index field of LLCB.
<i>Good Data</i> =	PSALCBNULL - null value of ALCB index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 3 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	When checking a NULL TEIB, the I2_I3inh field was found to be not null.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of I2_I3inh in TEIB.
<i>Good Data</i> =	DBNO - null value of I2_I3inh .
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	For a LAPB LLCB we found the I2_I3inh field in TEIB to be non null.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of I2_I3inh in TEIB.
<i>Good Data</i> =	DBNO - null value of I2_I3inh .
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 5 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	This error code involves pl2alw field of TEIB in DCCB. This field is used to determine whether the terminal endpoint identifier (TEI) of the TEIB is allowed service access point identifier (SAPI) 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The audit expected the value of pl2alw field of TEIB in DCCB to be DBYES, but it found the value to be other than DBYES. The value of pl2alw field of TEIB should be DBYES in the following conditions: - When isdata field in DCDB (DCDB[dcdbind].isdata) says that this channel is not in service (DBNO). - In all TEIB's pl2alw field except TEIB zero, for a LAPB logical link (LAPB packet service, determined from DCDB's pctype field) in DCCB.
Possible Error Effect:	Improper binding of the packet service. All the calls on that channel will be lost.
Corrective Action Taken:	The audit will report the error and recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	pl2alw field of the TEIB (teibptr→pl2alw).
<i>Good Data</i> =	DBYES - Expected value of pl2alw field.
<i>Logical Index</i> =	DCCB index (dcdbind).

<i>Dump 1 =</i>	The TEIB block where bad value of pl2alw field was found.
<i>Dump 2 =</i>	The index of the TEIB in TEIB array in DCCB (PSdccb[dccbind].teib[teibindx]).

Table 6 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The audit has found null or out of range pix value for LAPB channel when we have a valid tei value.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	0. The value chosen here is zero but any value between 0 and 7 will be a good value.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 7 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). The value of type field in the LLCB, found at the index derived from DCCB index in the PSfccb[] array, was not equal to PSL FLWCTRL, which is an error.
Possible Error Effect:	The FCCB LLCB has been initialized incorrectly and may be lost.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Bad Data =</i>	The value of type field of LLCB.
<i>Good Data =</i>	The index of DCCB we are auditing.
<i>Logical Index =</i>	The index derived from DCCB index in the PSfccb[] array.
<i>Dump =</i>	The LLCB whose type field has incorrect value.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	When performing linkage checks, for a NULL TEIB, the I2_I3inh field was found to be not null.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value of I2_I3inh in TEIB.
<i>Good Data =</i>	DBNO - null value of I2_I3inh .
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	"BAD COUNT". The audit has found some value, other than null, of act_vc field in DCCB when isdata field in DCBD says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will receive the DCCB.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	act_vc value of the DCCB.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	"BAD LINK". For a valid tei value, no valid sapi2llcb array index value was found, that is, no logical link was found.
Possible Error Effect:	The TEIB has no logical links attached to it.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in tei in this TEIB.
<i>Good Data</i> =	PSTEINULL - null value of tei.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 11 Error Code: BDPID

Error Code:	BDPID
Error Description:	Bad PID. The audit has found some value, other than null, of pid filed in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	0. The bad value chosen here is zero because you cannot dump a structure. The bad value is any field of pid structure.
<i>Good Data</i> =	0. The good data value is chosen 0 here because you cannot dump a structure. The good value here should be the null value of pid which is OSnullpid.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	PID field of this DCCB.

Table 12 Error Code: BLINK

Error Code:	BLINK
Error Description:	Bad link. At the ALCB index value, found in packet switched LAPD LLCB, the value of llcb_index in that ALCB does not correspond to the LAPD LLCB.
Possible Error Effect:	The linkage between ALCB and LLCB is bad.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the ALCB in which this error was found.
<i>Bad Data</i> =	Value in llcb_index field of ALCB found at alcb_index of LAPB LLCB.
<i>Good Data</i> =	llcbindex - the value of LAPB LLCB.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 13 Error Code: BPKT

Error Code:	BPKT
Error Description:	B-channel packet. The value of alcb_index field of the LLCB should not be null or out of range for LAPB LLCB. You should have a valid ALCB index value here.
Possible Error Effect:	The alcb_index in LAPB LLCB index has bad value or no ALCB is given to this call!!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in alcb index field of LLCB.
<i>Good Data</i> =	The value put in error code is zero. But any valid ALCB index value is good.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 14 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has found some value, other than null, in any of escnt array entry in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost?
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	Value of any of the index of escnt array in the DCCB.
<i>Good Data</i> =	0 - Null value of escnt array.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 15 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The value of dchan field of LAPB LLCB does not correspond to the DCCB index (implied linkage between channel number and DCCB index).
Possible Error Effect:	The channel value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in dchan field of LLCB.
<i>Good Data</i> =	dccbind - DCCB index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	1TEIB block where non-null tei value was found.
<i>Dump 2</i> =	1The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	llcb index.

Table 16 Error Code: HEAD

Error Code:	HEAD
Error Description:	An error was found in the headcell of the DCCB queue. The linkage between the headcell and the list is corrupted.
Possible Error Effect:	Condition notification calls for this channel will fail.
Corrective Action Taken:	The queue will be reconstructed. The audit will step through the PSLccb[] array and link to the queue any LCCB that belongs to this DCCB queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSLCCBNULL.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	AULNK structure.

Table 17 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	Idle channel. The value of second entry of sapi2llcb array in TEIB in LAPB LLCB case is not null. It should be null.
Possible Error Effect:	This TEIB should not have a management LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB and will relink LLCB and ALCB for this LAPB LLCB.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in the second entry of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

Table 18 Error Code: INDEX

Error Code:	INDEX
Error Description:	This error code involves pl2alw and pix fields of TEIB in DCCB. The pl2alw field is used to determine whether the terminal endpoint identifier (TEI) of the TEIB is allowed service access point identifier (SAPI) 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The pix field represents the tei party member. The value of pl2alw field in TEIB is not DBYES when the value of pix field in the same TEIB was not null (not equal to FCP NULL).
Possible Error Effect:	Improper binding of the packet service.
Corrective Action Taken:	The audit will report the error and will recover this TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pix field of the TEIB (teibptr→pix).
<i>Bad Data</i> =	Value of pl2alw field of the TEIB (teibptr→pl2alw).
<i>Good Data</i> =	DBYES, the value expected in pl2alw field.
<i>Logical Index</i> =	DCCB index (dccbind).
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdccb[dccbind].teib[teibindx]).

Table 19 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	Inhibit state. The audit has found some value, other than null, of chan_inh field in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	chan_inh value of the DCCB.
<i>Good Data</i> =	DBNO - null value of chan-inh.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 20 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	This error code involves pl2alw and tei fields of TEIB in DCCB. The pl2alw field is used to determine whether the TEI of this TEIB is allowed SAPI 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The tei field represents the terminal endpoint (equipment) identifier. The value of the pl2alw field of TEIB was not DBYES when the value of tei field of the TEIB was null (PSTEINULL).
Possible Error Effect:	Improper binding of the packet service.
Corrective Action Taken:	The audit will report the error and will recover this TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of tei field of TEIB (teibptr→tei).
<i>Bad Data</i> =	Value of pl2alw field of the TEIB (teibptr→pl2alw).
<i>Good Data</i> =	DBYES, the expected value of pl2alw field.
<i>Logical Index</i> =	DCCB (dccbind) index.

<i>Dump 1</i> =	The TEIB block where null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdccb[dcdbind].teib[teibindx]).

Table 21 Error Code: LAPD

Error Code:	LAPD
Error Description:	The value of llcbindex is out of range here, that is, it should have been a LAPD LLCB index, which it is not. This index could be circuit switched LAPD LLCB index, or management LAPD index or packet switched index. The way to determine which type it is, is by looking at the dump value of the last dump which is sapi_index. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	This TEIB should have a LAPD LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type field of the LLCB attached to this DCCB.
<i>Bad Data</i> =	Value in the first index of sapi2llcb array, that is, llcbindex.
<i>Good Data</i> =	The index of sapi2llcb array in the TEIB being audited.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The array index of the TEIB in TEIB array in DCCB.
<i>Dump 3</i> =	The LLCB attached to this DCCB.

Table 22 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). The value of dchan field in the FCCB LLCB, found at the index in the PSfccb[] array derived from DCCB index, does not match the DCCB index, which is an error.
Possible Error Effect:	The FCCB LLCB has not been initialized incorrectly and can be lost.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Bad Data</i> =	The value of dchan field of LLCB.
<i>Good Data</i> =	The index of DCCB we are auditing.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB whose type field has incorrect value.

Table 23 Error Code: NIDLE

Error Code:	NIDLE
Error Description:	Not idle. The value of first entry of sapi2llcb array in TEIB in LAPB LLCB case is not null. It should be null.
Possible Error Effect:	This TEIB should not have a circuit switched LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB and will relink LLCB and ALCB for this LAPB LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in the first entry of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.

<i>Dump =</i>	The index of the TEIB in teib array in DCCB.

Table 24 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	Non-null pix value. The audit has found some value, other than null, for pix value of this TEIB when the value of tei was null.
Possible Error Effect:	The TEIB has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of this TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 25 Error Code: NONULL

Error Code:	NONULL
Error Description:	Non-null sapi value. The audit has found some value, other than null, at any one of the index in sapi2llcb array in TEIB when its tei value is null.
Possible Error Effect:	The TEIB has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in any one of sapi2llcb array index in TEIB.
<i>Good Data =</i>	PSLLCBNULL - null value of sapi2llcb array indexes.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array where non-null value was found.

Table 26 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The alcb_index field of this circuit switched LAPD LLCB is not idle which it should be. You cannot have an ALCB for a circuit switched call.
Possible Error Effect:	The alcb_index value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in alcb_index field of LLCB.
<i>Good Data =</i>	PSALCBNULL - null value of ALCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcbindex.

Table 27 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is not in service (DBNO). The value of LLCB index, found at the index derived from DCCB index in the PSfcb[] array, was not null. It should have been null.
Possible Error Effect:	Either the FCCB LLCB has been incorrectly initialized or it is an unused/lost resource.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The index of DCCB being audited.
<i>Bad Data</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data</i> =	The null value for the LLCB index, PSLLCBNULL.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB found in the PSfccb[] array.

Table 28 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). Also, the PH channel type field, phctype , in this DCDB does not indicate this to be an EOC channel (PSEOC). Under this case, the value of LLCB index, found at the index derived from DCCB index in the PSfccb[] array, was not equal to the null value of LLCB, PSLLCBNULL, which is an error.
Possible Error Effect:	The PSfccb[] array index entry has been initialized incorrectly or an incorrect type of LLCB is attached to it which is a lost resource.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The phctype field of DCDB associated with this DCCB.
<i>Bad Data</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data</i> =	The index of DCCB being audited.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB found in PSfccb[] array.

Table 29 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	At the ALCB index value, found in LAPB LLCB, the value of llcb_index in that ALCB does not correspond to the LAPB LLCB.
Possible Error Effect:	The linkage between ALCB and LLCB is bad.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in llcb_index field of ALCB found at alcb_index of LAPB LLCB.
<i>Good Data</i> =	llcbindex - the value of LAPB LLCB.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 30 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of llcbindex is out of range here, that is, it should have been a LAPB LLCB index which it is not.
Possible Error Effect:	This TEIB should have a LAPB LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the TEIB in TEIB[] array in DCCB.
<i>Bad Data</i> =	Value in the third index of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB where the non-null tei value was found.
<i>Dump 2</i> =	The LLCB involved.

Table 31 Error Code: PHCTYPE

Error Code:	PHCTYPE

Error Description:	An invalid value of PH channel type (phctype) field in DCDB, associated with current DCCB being audited (PSdccb[dcdbind].phctype), was found by the audit.
Possible Error Effect:	This channel has wrong data from the SM (may be after recent change) and will need a refresh of data in DCDB from the SM.
Corrective Action Taken:	The audit will report the error and will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	Value of the phctype field of DCDB (PSdccb[dcdbind].phctype).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DCCB index (dcdbind).
<i>Dump</i> =	NA.

Table 32 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	At the index value equal to pix in p2aldb array in DCDB, an ALDB index value was found. The value of pix in the TEIB does not match the value of pix found in this ALDB.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the teib.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in pix field of TEIB.
<i>Good Data</i> =	FCP_NULL - null value of pix.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 33 Error Code: PIX2DCDB

Error Code:	PIX2DCDB
Error Description:	At the index value equal to pix in p2aldb array in the DCCB audit, a null value was found. There should have been a valid ALDB index value in this index of this array.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in pix field of TEIB.
<i>Good Data</i> =	FCP_NULL - null value of pix.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 34 Error Code: PIXBAD

Error Code:	PIXBAD
Error Description:	The audit has found some value, other than null, of pix field in DCCB. This error can occur in two different conditions. - When isdata field in DCDB says that this channel is not in service. - When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	- The DCCB has bad linkages or it is lost! - LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Pix value of the TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null pix value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 35 Error Code: PIXOUTOFRNG

Error Code:	PIXOUTOFRNG
Error Description:	The audit has found non-null, out of range pix value when we have a valid tei value. Note that pix value can be null for a valid tei value in LAPD channels.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix. The good data value chosen here is null value but any value between 0 and 7 can also be a good value.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 36 Error Code: PNT2PNT

Error Code:	PNT2PNT
Error Description:	Point to point. The value of sapi_i field of LAPD LLCB does not match the sapi_index value. To determine which type of LAPD LLCB it is, check the value of good data. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The sapi_i value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in sapi_i field of LLCB.
<i>Good Data =</i>	sapi_index - the valid index value of sapi2llcb array for LAPD LLCB.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 37 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The value of tix field of LAPD LLCB does not match the TEIB index of DCCB. To determine which type of LAPD LLCB it is, check the value of last dump variable. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.

Possible Error Effect:	The TEIB value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in tix field of LLCB.
<i>Good Data =</i>	teibind - the TEIB currently being audited.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in the TEIB being audited.

Table 38 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	Point to point. The value of dchan field of LAPD LLCB does not match the DCCB index (implied linkage between channel number and DCCB index). To determine which type of LAPD LLCB it is, check the value of last dump variable. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The channel value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in dchan field of LLCB.
<i>Good Data =</i>	dccbind - DCCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in the TEIB being audited.

Table 39 Error Code: RANGE

Error Code:	RANGE
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). Also the phctype field in DCDB indicates this to be an PSEOC channel. The value of FCCB LLCB index, found at the index derived from DCCB index in the PSfccb[] array, was found to be out of range, that is greater than null value, PSLLCBNULL.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule the routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of DCCB we are auditing.
<i>Bad Data =</i>	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data =</i>	The null value for the LLCB index, PSLLCBNULL.
<i>Logical Index =</i>	The index derived from DCCB index in the PSfccb[] array.
<i>Dump =</i>	NA.

Table 40 Error Code: SAPI

Error Code:	SAPI
Error Description:	The value of sapi_i field of the LLCB does not correspond to the PSKT value.

	For LAPB LLCB this value should be PSPKT.
Possible Error Effect:	The sapi i value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in sapi i field of LLCB.
<i>Good Data =</i>	PSPKT - the valid index value of sapi2llcb array for LAPB LLCB.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 41 Error Code: SAPIBAD

Error Code:	SAPIBAD
Error Description:	The audit has found some value, other than null, at any one of the index in sapi2llcb array in TEIB in DCCB. This error can occur in two different conditions. - When isdata field in DCBD says that this channel is not in service. - When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	- The DCCB has bad linkages or it is lost! - LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in sapi2llcb array index in TEIB.
<i>Good Data =</i>	PSLLCBNULL - null value of sapi2llcb array indexes.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in which non-null value was found.

Table 42 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The audit has found some value, other than null, of tei field in DCCB. This error can occur in two different conditions. - When isdata field in DCBD says that this channel is not service. - When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	- The DCCB has bad linkages or it is lost! - LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	TEI value of the TEIB.
<i>Good Data =</i>	PSTEINULL - null value of tei.
<i>Logical Index =</i>	DCCB index.

<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 43 Error Code: TEIOUTOFRNG

Error Code:	TEIOUTOFRNG
Error Description:	The audit found an out of range TEI value in the tei field of this TEIB.
Possible Error Effect:	This TEIB had bad TEI value/binding.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of tei in TEIB.
<i>Good Data</i> =	PSTEINULL - null value of tei.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 44 Error Code: TIX

Error Code:	TIX
Error Description:	The value of tix field of the LLCB does not correspond to this TEIB index of DCCB.
Possible Error Effect:	The TEIB value of the LLCB attached to DCCB in bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in tix field of LLCB.
<i>Good Data</i> =	teibind - the TEIB currently being audited.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 45 Error Code: TYPE

Error Code:	TYPE
Error Description:	For a LAPB LLCB, the I2_I3inh field in TEIB was set to DBYES for one of the following channel types when it should have been set to DBNO. The channel types include PSDS_LINK, PSDP_LINK, PSDSP_LINK, and PSDP_MP.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of I2_I3inh in TEIB.
<i>Good Data</i> =	DBNO - null value of I2_I3inh.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

DCCB-B

Software Release: 5E12 only

1. D-CHANNEL CONTROL BLOCK (DCCB) AUDIT

The DCCB audit verifies data associated with D-channel controls resident in the PH. The audit verifies information about each of the fields of the DCCB, and about each terminal end point identifier (TEIB) in it during idle and busy (in use) DCCB. If there is an error in one of the TEIBs, only that TEIB is corrected. Calls on other TEIBs are not affected. If there is a problem with any other field in the DCCB, the whole DCCB is cleared, including all 8 TEIBs. Head cell checks for LCCB queue for conditional notification are also added here. The flow control control block (FCCB) logical link control blocks (LLCB) are used to control the flood of embedded operational channel (EOC) maintenance messages that may result when one or more remote terminals (RT) are powered up. The RTs referred here are the TR303 RTs associated with the integrated digital carrier unit (IDCU). Checks related to FCCB LLCBs are also included in this audit. The DCCB audit runs in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DCCB audit steps through the PSdccb[] array. It interfaces with the D-channel data block (DCDB), the LLCB, the TEIB and the access line control block (ALCB) data structures. If the DCCB index number is divisible by four and if the PH channel type is PS_EOC, an FCCB LLCB is also used that can be associated with this DCCB.

3. AUDIT CHECKS

The audit checks the linkages in PSdccb with the logical link control block (PSllcb).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_VC_BAD

Error Code:	ACT_VC_BAD
Error Description:	The value in the act_vc field of DCCB does not match the computer value of the same by audits.
Possible Error Effect:	No immediate effect on the operation of the switch. But this count is used for camp-on and for "blocking the channel if it is idle". It can give different results if this count is not correct.
Corrective Action Taken:	The audit will report the error and correct the value of the act_vc field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCCB in which this error was found.
<i>Bad Data =</i>	Value in act_vc field in DCCB.
<i>Good Data =</i>	Value in active_vc field as calculated by audit.
<i>Logical Index =</i>	DCCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: ALCBIND

Error Code:	ALCBIND
Error Description:	The alcb_index field of the management LAPD LLCB is not idle which it should be. You cannot have an ALCB for a management call.
Possible Error Effect:	The alcb_index value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.

<i>Bad Data</i> =	Value in alcb_index field of LLCB.
<i>Good Data</i> =	PSALCBNULL - null value of ALCB index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 3 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	When checking a NULL TEIB, the I2_I3inh field was found to be not null.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of I2_I3inh in TEIB.
<i>Good Data</i> =	DBNO - null value of I2_I3inh .
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	For a LAPB LLCB we found the I2_I3inh field in TEIB to be non null.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of I2_I3inh in TEIB.
<i>Good Data</i> =	DBNO - null value of I2_I3inh .
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 5 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	This error code involves pl2alw field of TEIB in DCCB. This field is used to determine whether the tei (terminal endpoint identifier) of the TEIB is allowed service access point identifier (SAPI) 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The audit expected the value of pl2alw field of TEIB in DCCB to be DBYES, but it found the value to be other than DBYES. The value of pl2alw field of TEIB should be DBYES in the following conditions: - When isdata field in DCDB (DCDB[dcdbind].isdata) says that this channel is not in service (DBNO). - In all TEIB's pl2alw field except TEIB zero, for a LAPB logical link (LAPB packet service, determined from DCDB's pctype field) in DCCB.
Possible Error Effect:	Improper binding of the packet service. All the calls on that channel will be lost.
Corrective Action Taken:	The audit will report the error and recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	pl2alw field of the TEIB (teibptr→pl2alw).
<i>Good Data</i> =	DBYES - Expected value of pl2alw field.
<i>Logical Index</i> =	DCCB index (dcdbind).

<i>Dump 1 =</i>	The TEIB block where bad value of pl2alw field was found.
<i>Dump 2 =</i>	The index of the TEIB in TEIB array in DCCB (PSdccb[dcdbind].teib[teibindx]).

Table 6 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The audit has found null or out of range pix value for LAPB channel when we have a valid tei value.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	0. The value chosen here is zero but any value between 0 and 7 will be a good value.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 7 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). The value of type field in the LLCB, found at the index derived from DCCB index in the PSfccb[] array, was not equal to PSL_FWLCTRL, which is an error.
Possible Error Effect:	The FCCB LLCB has been initialized incorrectly and may be lost.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Bad Data =</i>	The value of type field of LLCB.
<i>Good Data =</i>	The index of DCCB we are auditing.
<i>Logical Index =</i>	The index derived from DCCB index in the PSfccb[] array.
<i>Dump =</i>	The LLCB whose type field has incorrect value.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	When performing linkage checks, for a NULL TEIB, the I2_I3inh field was found to be not null.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value of I2_I3inh in TEIB.
<i>Good Data =</i>	DBNO - null value of I2_I3inh .
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	Bad count. The audit has found some value, other than null, of act_vc field in DCCB when isdata field in DCBD says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will receive the DCCB.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	act_vc value of the DCCB.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	Bad link. For a valid tei value, no valid sapi2llcb array index value was found, that is, no logical link was found.
Possible Error Effect:	The TEIB has no logical links attached to it.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in tei in this TEIB.
<i>Good Data</i> =	PSTEINULL - null value of tei.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 11 Error Code: BDPID

Error Code:	BDPID
Error Description:	Bad processor ID (PID). The audit has found some value, other than null, of pid filed in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	0. The bad value chosen here is zero because you cannot dump a structure. The bad value is any field of pid structure.
<i>Good Data</i> =	0. The good data value is chosen 0 here because you cannot dump a structure. The good value here should be the null value of pid which is OSnullpid.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	PID field of this DCCB.

Table 12 Error Code: BLINK

Error Code:	BLINK
Error Description:	Bad link. At the ALCB index value, found in packet switched LAPD LLCB, the value of llcb_index in that ALCB does not correspond to the LAPD LLCB.
Possible Error Effect:	The linkage between ALCB and LLCB is bad.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the ALCB in which this error was found.
<i>Bad Data</i> =	Value in llcb_index field of ALCB found at alcb_index of LAPB LLCB.
<i>Good Data</i> =	llcbindex - the value of LAPB LLCB.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 13 Error Code: BPKT

Error Code:	BPKT
Error Description:	B-channel packet. The value of alcb_index field of the LLCB should not be null or out of range for LAPB LLCB. You should have a valid ALCB index value here.
Possible Error Effect:	The alcb_index in LAPB LLCB index has bad value or no ALCB is given to this call!!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in alcb index field of LLCB.
<i>Good Data</i> =	The value put in error code is zero. But any valid ALCB index value is good.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 14 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has found some value, other than null, in any of escnt array entry in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost?
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	Value of any of the index of escnt array in the DCCB.
<i>Good Data</i> =	0 - Null value of escnt array.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 15 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The value of dchan field of LAPB LLCB does not correspond to the DCCB index (implied linkage between channel number and DCCB index).
Possible Error Effect:	The channel value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in dchan field of LLCB.
<i>Good Data</i> =	dccbind - DCCB index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	lcb index.

Table 16 Error Code: HEAD

Error Code:	HEAD
Error Description:	An error was found in the headcell of the DCCB queue. The linkage between the headcell and the list is corrupted.
Possible Error Effect:	Condition notification calls for this channel will fail.
Corrective Action Taken:	The queue will be reconstructed. The audit will step through the PSLccb[] array and link to the queue any LCCB that belongs to this DCCB queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSLCCBNULL.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	AULNK structure.

Table 17 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	Idle channel. The value of second entry of sapi2llcb array in TEIB in LAPB LLCB case is not null. It should be null.
Possible Error Effect:	This TEIB should not have a management LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB and will relink LLCB and ALCB for this LAPB LLCB.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in the second entry of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

Table 18 Error Code: INDEX

Error Code:	INDEX
Error Description:	This error code involves pl2alw and pix fields of TEIB in DCCB. The pl2alw field is used to determine whether the terminal endpoint identifier (TEI) of the TEIB is allowed service access point identifier (SAPI) 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The pix field represents the tei party member. The value of pl2alw field in TEIB is not DBYES when the value of pix field in the same TEIB was not null (not equal to FCP NULL).
Possible Error Effect:	Improper binding of the packet service.
Corrective Action Taken:	The audit will report the error and will recover this TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pix field of the TEIB (teibptr→pix).
<i>Bad Data</i> =	Value of pl2alw field of the TEIB (teibptr→pl2alw).
<i>Good Data</i> =	DBYES, the value expected in pl2alw field.
<i>Logical Index</i> =	DCCB index (dccbind).
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdccb[dccbind].teib[teibindx]).

Table 19 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	Inhibit state. The audit has found some value, other than null, of chan_inh field in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	chan_inh value of the DCCB.
<i>Good Data</i> =	DBNO - null value of chan-inh.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 20 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	This error code involves pl2alw and tei fields of TEIB in DCCB. The pl2alw field is used to determine whether the TEI of this TEIB is allowed SAPI 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The tei field represents the terminal endpoint (equipment) identifier. The value of the pl2alw field of TEIB was not DBYES when the value of tei field of the TEIB was null (PSTEINULL).
Possible Error Effect:	Improper binding of the packet service.
Corrective Action Taken:	The audit will report the error and will recover this TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of tei field of TEIB (teibptr→tei).
<i>Bad Data</i> =	Value of pl2alw field of the TEIB (teibptr→pl2alw).
<i>Good Data</i> =	DBYES, the expected value of pl2alw field.
<i>Logical Index</i> =	DCCB (dccbind) index.

<i>Dump 1</i> =	The TEIB block where null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdccb[dcdbind].teib[teibindx]).

Table 21 Error Code: LAPD

Error Code:	LAPD
Error Description:	The value of llcbindex is out of range here, that is, it should have been a LAPD LLCB index, which it is not. This index could be circuit switched LAPD LLCB index, or management LAPD index or packet switched index. The way to determine which type it is, is by looking at the dump value of the last dump which is sapi_index. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	This TEIB should have a LAPD LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type field of the LLCB attached to this DCCB.
<i>Bad Data</i> =	Value in the first index of sapi2llcb array, that is, llcbindex.
<i>Good Data</i> =	The index of sapi2llcb array in the TEIB being audited.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The array index of the TEIB in TEIB array in DCCB.
<i>Dump 3</i> =	The LLCB attached to this DCCB.

Table 22 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). The value of dchan field in the FCCB LLCB, found at the index in the PSfccb[] array derived from DCCB index, does not match the DCCB index, which is an error.
Possible Error Effect:	The FCCB LLCB has not been initialized incorrectly and can be lost.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Bad Data</i> =	The value of dchan field of LLCB.
<i>Good Data</i> =	The index of DCCB we are auditing.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB whose type field has incorrect value.

Table 23 Error Code: NIDLE

Error Code:	NIDLE
Error Description:	Not idle. The value of first entry of sapi2llcb array in TEIB in LAPB LLCB case is not null. It should be null.
Possible Error Effect:	This TEIB should not have a circuit switched LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB and will relink LLCB and ALCB for this LAPB LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in the first entry of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.

<i>Dump =</i>	The index of the TEIB in teib array in DCCB.

Table 24 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	Non-null pix value. The audit has found some value, other than null, for pix value of this TEIB when the value of tei was null.
Possible Error Effect:	The TEIB has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of this TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 25 Error Code: NONULL

Error Code:	NONULL
Error Description:	Nonnull sapi value. The audit has found some value, other than null, at any one of the index in sapi2llcb array in TEIB when its tei value is null.
Possible Error Effect:	The TEIB has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in any one of sapi2llcb array index in TEIB.
<i>Good Data =</i>	PSLLCBNULL - null value of sapi2llcb array indexes.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array where non-null value was found.

Table 26 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The alcb_index field of this circuit switched LAPD LLCB is not idle which it should be. You cannot have an ALCB for a circuit switched call.
Possible Error Effect:	The alcb_index value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in alcb_index field of LLCB.
<i>Good Data =</i>	PSALCBNULL - null value of ALCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcbindex.

Table 27 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is not in service (DBNO). The value of LLCB index, found at the index derived from DCCB index in the PSfcb[] array, was not null. It should have been null.
Possible Error Effect:	Either the FCCB LLCB has been incorrectly initialized or it is an unused/lost resource.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The index of DCCB being audited.
<i>Bad Data</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data</i> =	The null value for the LLCB index, PSLLCBNULL.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB found in the PSfccb[] array.

Table 28 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). Also, the PH channel type field, phctype , in this DCDB does not indicate this to be an EOC channel (PSEOC). Under this case, the value of LLCB index, found at the index derived from DCCB index in the PSfccb[] array, was not equal to the null value of LLCB, PSLLCBNULL, which is an error.
Possible Error Effect:	The PSfccb[] array index entry has been initialized incorrectly or an incorrect type of LLCB is attached to it which is a lost resource.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The phctype field of DCDB associated with this DCCB.
<i>Bad Data</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data</i> =	The index of DCCB being audited.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB found in PSfccb[] array.

Table 29 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	At the ALCB index value, found in LAPB LLCB, the value of llcb_index in that ALCB does not correspond to the LAPB LLCB.
Possible Error Effect:	The linkage between ALCB and LLCB is bad.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in llcb_index field of ALCB found at alcb_index of LAPB LLCB.
<i>Good Data</i> =	llcbindex - the value of LAPB LLCB.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 30 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of llcbindex is out of range here, that is, it should have been a LAPB LLCB index which it is not.
Possible Error Effect:	This TEIB should have a LAPB LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the TEIB in TEIB[] array in DCCB.
<i>Bad Data</i> =	Value in the third index of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB where the non-null tei value was found.
<i>Dump 2</i> =	The LLCB involved.

Table 31 Error Code: PHCTYPE

Error Code:	PHCTYPE

Error Description:	An invalid value of PH channel type (phctype) field in DCDB, associated with current DCCB being audited (PSdccb[dccbind].phctype), was found by the audit.
Possible Error Effect:	This channel has wrong data from the SM (may be after recent change) and will need a refresh of data in DCDB from the SM.
Corrective Action Taken:	The audit will report the error and will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	Value of the phctype field of DCDB (PSdccb[dccbind].phctype).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DCCB index (dccbind).
<i>Dump</i> =	NA.

Table 32 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	At the index value equal to pix in p2aldb array in DCBD, an ALDB index value was found. The value of pix in the TEIB does not match the value of pix found in this ALDB.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the teib.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in pix field of TEIB.
<i>Good Data</i> =	FCP_NULL - null value of pix.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 33 Error Code: PIX2DCDB

Error Code:	PIX2DCDB
Error Description:	At the index value equal to pix in p2aldb array in the DCCB audit, a null value was found. There should have been a valid ALDB index value in this index of this array.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in pix field of TEIB.
<i>Good Data</i> =	FCP_NULL - null value of pix.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 34 Error Code: PIXBAD

Error Code:	PIXBAD
Error Description:	The audit has found some value, other than null, of pix field in DCCB. This error can occur in two different conditions. 1 = When isdata field in DCDB says that this channel is not in service. 2 = When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	Corresponds to Error Description as follows: 1 = The DCCB has bad linkages or it is lost! 2 = LAPB logical link has bad linkages.

Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Pix value of the TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null pix value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 35 Error Code: PIXOUTOFRNG

Error Code:	PIXOUTOFRNG
Error Description:	The audit has found non-null, out of range pix value when we have a valid tei value. Note that pix value can be null for a valid tei value in LAPD channels.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix. The good data value chosen here is null value but any value between 0 and 7 can also be a good value.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 36 Error Code: PNT2PNT

Error Code:	PNT2PNT
Error Description:	Point to point. The value of sapi_i field of LAPD LLCB does not match the sapi_index value. To determine which type of LAPD LLCB it is, check the value of good data. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The sapi_i value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in sapi_i field of LLCB.
<i>Good Data =</i>	sapi_index - the valid index value of sapi2llcb array for LAPD LLCB.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 37 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The value of tix field of LAPD LLCB does not match the TEIB index of DCCB. To determine which type of LAPD LLCB it is, check the value of last dump variable. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.

Possible Error Effect:	The TEIB value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in tix field of LLCB.
<i>Good Data =</i>	teibind - the TEIB currently being audited.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in the TEIB being audited.

Table 38 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	Point to point. The value of dchan field of LAPD LLCB does not match the DCCB index (implied linkage between channel number and DCCB index). To determine which type of LAPD LLCB it is, check the value of last dump variable. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The channel value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in dchan field of LLCB.
<i>Good Data =</i>	dccbind - DCCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in the TEIB being audited.

Table 39 Error Code: RANGE

Error Code:	RANGE
Error Description:	The DCCB index is divisible by four and the isdata bit in DCDB associated with this DCCB says that this channel is in service (DBYES). Also the phctype field in DCDB indicates this to be an PSEOC channel. The value of FCCB LLCB index, found at the index derived from DCCB index in the PSfccb[] array, was found to be out of range, that is greater than null value, PSLLCBNULL.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule the routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of DCCB we are auditing.
<i>Bad Data =</i>	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data =</i>	The null value for the LLCB index, PSLLCBNULL.
<i>Logical Index =</i>	The index derived from DCCB index in the PSfccb[] array.
<i>Dump =</i>	NA.

Table 40 Error Code: SAPI

Error Code:	SAPI
Error Description:	The value of sapi_i field of the LLCB does not correspond to the PSKT value.

	For LAPB LLCB this value should be PSPKT.
Possible Error Effect:	The sapi i value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in sapi i field of LLCB.
<i>Good Data =</i>	PSPKT - the valid index value of sapi2llcb array for LAPB LLCB.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 41 Error Code: SAPIBAD

Error Code:	SAPIBAD
Error Description:	The audit has found some value, other than null, at any one of the index in sapi2llcb array in TEIB in DCCB. This error can occur in two different conditions. 1 = When isdata field in DCBD says that this channel is not in service. 2 = When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	Corresponds to Error Description as follows: 1 = The DCCB has bad linkages or it is lost! 2 = LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in sapi2llcb array index in TEIB.
<i>Good Data =</i>	PSLLCBNULL - null value of sapi2llcb array indexes.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in which non-null value was found.

Table 42 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The audit has found some value, other than null, of tei field in DCCB. This error can occur in two different conditions. 1 = When isdata field in DCBD says that this channel is not service. 2 = When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	Corresponds to Error Description as follows: 1 = The DCCB has bad linkages or it is lost! 2 = LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	TEI value of the TEIB.
<i>Good Data =</i>	PSTEINULL - null value of tei.

<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 43 Error Code: TEIOUTOFRNG

Error Code:	TEIOUTOFRNG
Error Description:	The audit found an out of range TEI value in the tei field of this TEIB.
Possible Error Effect:	This TEIB had bad TEI value/binding.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of tei in TEIB.
<i>Good Data</i> =	PSTEINULL - null value of tei.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 44 Error Code: TIX

Error Code:	TIX
Error Description:	The value of tix field of the LLCB does not correspond to this TEIB index of DCCB.
Possible Error Effect:	The TEIB value of the LLCB attached to DCCB in bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in tix field of LLCB.
<i>Good Data</i> =	teibind - the TEIB currently being audited.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 45 Error Code: TYPE

Error Code:	TYPE
Error Description:	For a LAPB LLCB, the I2_I3inh field in TEIB was set to DBYES for one of the following channel types when it should have been set to DBNO. The channel types include PSDS LINK, PSDP LINK, PSDSP LINK, and PSDP MP.
Possible Error Effect:	This TEIB has I2_I3inh set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of I2_I3inh in TEIB.
<i>Good Data</i> =	DBNO - null value of I2_I3inh.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

DCCB-C

Software Release: 5E13 and later

1. D-CHANNEL CONTROL BLOCK (DCCB) AUDIT

The DCCB audit verifies data associated with D-channel controls resident in the PH. The audit verifies information about each of the fields of the DCCB, and about each terminal end point identifier (TEIB) in it during idle and busy (in use) DCCB. If there is an error in one of the TEIBs, only that TEIB is corrected. Calls on other TEIBs are not affected. If there is a problem with any other field in the DCCB, the whole DCCB is cleared, including all 8 TEIBs. Head cell checks for LCCB queue for conditional notification are also added here. The flow control control block (FCCB) logical link control blocks (LLCB) are used to control the flood of embedded operational channel (EOC) maintenance messages that may result when one or more remote terminals (RT) are powered up. The RTs referred here are the TR303 RTs associated with the integrated digital carrier unit (IDCU). Checks related to FCCB LLCBs are also included in this audit.

The DCCB audit runs in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DCCB audit steps through the PSdccb[] array. It interfaces with the D-channel data block (DCDB), the LLCB, the TEIB and the access line control block (ALCB) data structures. If the DCCB index number is divisible by four and if the PH channel type is PS_EOC, an FCCB LLCB is also used that can be associated with this DCCB.

3. AUDIT CHECKS

The audit checks the linkages in PSdccb with the logical link control block (PSllcb).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_VC_BAD

Error Code:	ACT_VC_BAD
Error Description:	The value in the act_vc field of DCCB does not match the computer value of the same by audits.
Possible Error Effect:	No immediate effect on the operation of the switch. But this count is used for camp-on and for "blocking the channel if it is idle". It can give different results if this count is not correct.
Corrective Action Taken:	The audit will report the error and correct the value of the act_vc field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCCB in which this error was found.
<i>Bad Data =</i>	Value in act_vc field in DCCB.
<i>Good Data =</i>	Value in active_vc field as calculated by audit.
<i>Logical Index =</i>	DCCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: ALCBIND

Error Code:	ALCBIND
Error Description:	The alcb_index field of the management LAPD LLCB is not idle which it should be. You cannot have an ALCB for a management call.
Possible Error Effect:	The alcb_index value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in alcb_index field of LLCB.
<i>Good Data =</i>	PSALCBNULL - null value of ALCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 3 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	When checking a NULL TEIB, the <i>I2_I3inh</i> field was found to be not null.
Possible Error Effect:	This TEIB has <i>I2_I3inh</i> set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value of <i>I2_I3inh</i> in TEIB.
<i>Good Data =</i>	DBNO - null value of <i>I2_I3inh</i> .
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	For a LAPB LLCB we found the <i>I2_I3inh</i> field in TEIB to be non null.
Possible Error Effect:	This TEIB has <i>I2_I3inh</i> set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value of <i>I2_I3inh</i> in TEIB.
<i>Good Data =</i>	DBNO - null value of <i>I2_I3inh</i> .
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 5 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	This error code involves pl2alw field of TEIB in DCCB. This field is used to determine whether the tei (terminal endpoint identifier) of the TEIB is allowed service access point identifier (SAPI) 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The audit expected the value of pl2alw field of TEIB in DCCB to be DBYES, but it found the value to be other than DBYES. The value of pl2alw field of TEIB should be DBYES in the following conditions: - When isdata field in DCDB (DCDB[dcdbind].isdata) says that this channel is not in service (DBNO). - In all TEIB's pl2alw field except TEIB zero, for a LAPB logical link (LAPB packet service, determined from DCDB's phctype field) in DCCB.
Possible Error Effect:	Improper binding of the packet service. All the calls on that channel will be lost.
Corrective Action Taken:	The audit will report the error and recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.

<i>Bad Data</i> =	pl2alw field of the TEIB (teibptr->pl2alw).
<i>Good Data</i> =	DBYES - Expected value of pl2alw field.
<i>Logical Index</i> =	DCCB index (dccbind).
<i>Dump 1</i> =	The TEIB block where bad value of pl2alw field was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdcccb[dccbind].teib[teibindx]).

Table 6 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The audit has found null or out of range pix value for LAPB channel when we have a valid tei value.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the TEIB in teib array in DCCB.
<i>Bad Data</i> =	Value in pix field of TEIB.
<i>Good Data</i> =	0. The value chosen here is zero but any value between 0 and 7 will be a good value.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	TEIB block where non-null tei value was found.

Table 7 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The DCCB index is divisible by four and the <i>isdata</i> bit in DCDB associated with this DCCB says that this channel is in service (DBYES). The value of <i>type</i> field in the LLCB, found at the index derived from DCCB index in the PSfccb[] array, was not equal to PSL_FLWCTRL, which is an error.
Possible Error Effect:	The FCCB LLCB has been initialized incorrectly and may be lost.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Bad Data</i> =	The value of <i>type</i> field of LLCB.
<i>Good Data</i> =	The index of DCCB we are auditing.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB whose <i>type</i> field has incorrect value.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	When performing linkage checks, for a NULL TEIB, the <i>I2_I3inh</i> field was found to be not null.
Possible Error Effect:	This TEIB has <i>I2_I3inh</i> set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of <i>I2_I3inh</i> in TEIB.
<i>Good Data</i> =	DBNO - null value of <i>I2_I3inh</i> .
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	Bad count. The audit has found some value, other than null, of act_vc field in DCCB when <i>isdata</i> field in DCBD says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will receive the DCCB.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	act_vc value of the DCCB.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	Bad link. For a valid tei value, no valid sapi2llcb array index value was found, that is, no logical link was found.
Possible Error Effect:	The TEIB has no logical links attached to it.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in tei in this TEIB.
<i>Good Data</i> =	PSTEINULL - null value of tei.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 11 Error Code: BDPID

Error Code:	BDPID
Error Description:	Bad processor ID (PID). The audit has found some value, other than null, of pid filed in DCCB when <i>isdata</i> field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	0. The bad value chosen here is zero because you cannot dump a structure. The bad value is any field of pid structure.
<i>Good Data</i> =	0. The good data value is chosen 0 here because you cannot dump a structure. The good value here should be the null value of pid which is OSnullpid.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	PID field of this DCCB.

Table 12 Error Code: BLINK

Error Code:	BLINK
Error Description:	Bad link. At the ALCB index value, found in packet switched LAPD LLCB, the value of llcb_index in that ALCB does not correspond to the LAPD LLCB.
Possible Error Effect:	The linkage between ALCB and LLCB is bad.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the ALCB in which this error was found.
<i>Bad Data</i> =	Value in llcb_index field of ALCB found at alcb_index of LAPB LLCB.
<i>Good Data</i> =	llcbindex - the value of LAPB LLCB.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 13 Error Code: BPKT

Error Code:	BPKT
Error Description:	B-channel packet. The value of alcb_index field of the LLCB should not be null or out of range for LAPB LLCB. You should have a valid ALCB index value here.
Possible Error Effect:	The alcb_index in LAPB LLCB index has bad value or no ALCB is given to this call!!!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in alcb_index field of LLCB.
<i>Good Data</i> =	The value put in error code is zero. But any valid ALCB index value is good.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 14 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has found some value, other than null, in any of escnt array entry in DCCB when <i>isdata</i> field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost?
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	Value of any of the index of escnt array in the DCCB.
<i>Good Data</i> =	0 - Null value of escnt array.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 15 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The value of dchan field of LAPB LLCB does not correspond to the DCCB index (implied linkage between channel number and DCCB index).
Possible Error Effect:	The channel value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in dchan field of LLCB.
<i>Good Data</i> =	dccbind - DCCB index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	lcb index.

Table 16 Error Code: HEAD

Error Code:	HEAD
Error Description:	An error was found in the headcell of the DCCB queue. The linkage between the headcell and the list is corrupted.
Possible Error Effect:	Condition notification calls for this channel will fail.
Corrective Action Taken:	The queue will be reconstructed. The audit will step through the PSlcccb[] array and link to the queue any LCCB that belongs to this DCCB queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSLCCBNULL.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	AULNK structure.

Table 17 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	Idle channel. The value of second entry of sapi2llcb array in TEIB in LAPB LLCB case is not null. It should be null.
Possible Error Effect:	This TEIB should not have a management LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB and will relink LLCB and ALCB for this LAPB LLCB.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in the second entry of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

Table 18 Error Code: INDEX

Error Code:	INDEX
Error Description:	This error code involves pl2alw and pix fields of TEIB in DCCB. The pl2alw field is used to determine whether the terminal endpoint identifier (TEI) of the TEIB is allowed service access point identifier (SAPI) 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The pix field represents the tei party member. The value of pl2alw field in TEIB is not DBYES when the value of pix field in the same TEIB was not null (not equal to FCP NULL).
Possible Error Effect:	Improper binding of the packet service.
Corrective Action Taken:	The audit will report the error and will recover this TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pix field of the TEIB (teibptr->pix).
<i>Bad Data</i> =	Value of pl2alw field of the TEIB (teibptr->pl2alw).
<i>Good Data</i> =	DBYES, the value expected in pl2alw field.
<i>Logical Index</i> =	DCCB index (dccbind).
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdcccb[dccbind].teib[teibndx]).

Table 19 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	Inhibit state. The audit has found some value, other than null, of chan_inh field in DCCB when isdata field in DCDB says that this channel is not in service.
Possible Error Effect:	The DCCB has bad linkages or it is lost!
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCCB in which this error was found.
<i>Bad Data</i> =	chan_inh value of the DCCB.
<i>Good Data</i> =	DBNO - null value of chan_inh.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	NA.

Table 20 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	This error code involves pl2alw and tei fields of TEIB in DCCB. The pl2alw field is used to determine whether the TEI of this TEIB is allowed SAPI 16 or not. SAPI 16 pertains to packet switched calls. They can carry either X.25 packet (from a DSL or modem pool) or an IP packet from a nail-up inter-SM link. The tei field represents the terminal endpoint (equipment) identifier. The value of the pl2alw field of TEIB was not DBYES when the value of tei field of the TEIB was null (PSTEINULL).
Possible Error Effect:	Improper binding of the packet service.
Corrective Action Taken:	The audit will report the error and will recover this TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of tei field of TEIB (teibptr->tei).
<i>Bad Data</i> =	Value of pl2alw field of the TEIB (teibptr->pl2alw).
<i>Good Data</i> =	DBYES, the expected value of pl2alw field.
<i>Logical Index</i> =	DCCB (dccbind) index.

<i>Dump 1</i> =	The TEIB block where null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in TEIB array in DCCB (PSdccb[dccbind].teib[teibindx]).

Table 21 Error Code: LAPD

Error Code:	LAPD
Error Description:	The value of llcbindex is out of range here, that is, it should have been a LAPD LLCB index, which it is not. This index could be circuit switched LAPD LLCB index, or management LAPD index or packet switched index. The way to determine which type it is, is by looking at the dump value of the last dump which is sapi_index. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	This TEIB should have a LAPD LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>type</i> field of the LLCB attached to this DCCB.
<i>Bad Data</i> =	Value in the first index of sapi2llcb array, that is, llcbindex.
<i>Good Data</i> =	The index of sapi2llcb array in the TEIB being audited.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The array index of the TEIB in TEIB array in DCCB.
<i>Dump 3</i> =	The LLCB attached to this DCCB.

Table 22 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The DCCB index is divisible by four and the <i>isdata</i> bit in DCDB associated with this DCCB says that this channel is in service (DBYES). The value of <i>dchan</i> field in the FCCB LLCB, found at the <i>index</i> in the PSfcb[] array derived from DCCB index, does not match the DCCB index, which is an error.
Possible Error Effect:	The FCCB LLCB has not been initialized incorrectly and can be lost.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfcb[] array.
<i>Bad Data</i> =	The value of <i>dchan</i> field of LLCB.
<i>Good Data</i> =	The index of DCCB we are auditing.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfcb[] array.
<i>Dump</i> =	The LLCB whose <i>type</i> field has incorrect value.

Table 23 Error Code: NIDLE

Error Code:	NIDLE
Error Description:	Not idle. The value of first entry of sapi2llcb array in TEIB in LAPB LLCB case is not null. It should be null.
Possible Error Effect:	This TEIB should not have a circuit switched LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB and will relink LLCB and ALCB for this LAPB LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value in the first entry of sapi2llcb array.
<i>Good Data</i> =	PSLLCBNULL - null value of sapi2llcb array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

Table 24 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	Non-null pix value. The audit has found some value, other than null, for pix value of this TEIB when the value of tei was null.
Possible Error Effect:	The TEIB has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of this TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 25 Error Code: NONULL

Error Code:	NONULL
Error Description:	Nonnull sapi value. The audit has found some value, other than null, at any one of the index in sapi2llcb array in TEIB when its tei value is null.
Possible Error Effect:	The TEIB has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in any one of sapi2llcb array index in TEIB.
<i>Good Data =</i>	PSLLCBNULL - null value of sapi2llcb array indexes.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array where non-null value was found.

Table 26 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The alcb_index field of this circuit switched LAPD LLCB is not idle which it should be. You cannot have an ALCB for a circuit switched call.
Possible Error Effect:	The alcb_index value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in alcb_index field of LLCB.
<i>Good Data =</i>	PSALCBNULL - null value of ALCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcbindex.

Table 27 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The DCCB index is divisible by four and the <i>isdata</i> bit in DCDB associated with this DCCB says that this channel is not in service (DBNO). The value of LLCB index, found at the index derived from DCCB index in the PSfccb[] array, was not null. It should have been null.
Possible Error Effect:	Either the FCCB LLCB has been incorrectly initialized or it is an unused/lost resource.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of DCCB being audited.

<i>Bad Data</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data</i> =	The null value for the LLCB index, PSLLCBNULL.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB found in the PSfccb[] array.

Table 28 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The DCCB index is divisible by four and the <i>isdata</i> bit in DCDB associated with this DCCB says that this channel is in service (DBYES). Also, the PH channel type field, <i>phctype</i> , in this DCDB does not indicate this to be an EOC channel (PSEOC). Under this case, the value of LLCB index, found at the index derived from DCCB index in the PSfccb[] array, was not equal to the null value of LLCB, PSLLCBNULL, which is an error.
Possible Error Effect:	The PSfccb[] array index entry has been initialized incorrectly or an incorrect type of LLCB is attached to it which is a lost resource.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>phctype</i> field of DCDB associated with this DCCB.
<i>Bad Data</i> =	The value of LLCB index found at the index derived from DCCB index in the PSfccb[] array.
<i>Good Data</i> =	The index of DCCB being audited.
<i>Logical Index</i> =	The index derived from DCCB index in the PSfccb[] array.
<i>Dump</i> =	The LLCB found in PSfccb[] array.

Table 29 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	At the ALCB index value, found in LAPB LLCB, the value of <i>llcb_index</i> in that ALCB does not correspond to the LAPB LLCB.
Possible Error Effect:	The linkage between ALCB and LLCB is bad.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in <i>llcb_index</i> field of ALCB found at <i>alcb_index</i> of LAPB LLCB.
<i>Good Data</i> =	<i>llcbindex</i> - the value of LAPB LLCB.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null <i>tei</i> value was found.
<i>Dump 2</i> =	The index of the TEIB in <i>teib</i> array in DCCB.

Table 30 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of <i>llcbindex</i> is out of range here, that is, it should have been a LAPB LLCB index which it is not.
Possible Error Effect:	This TEIB should have a LAPB LLCB logical link. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the TEIB in <i>TEIB[]</i> array in DCCB.
<i>Bad Data</i> =	Value in the third index of <i>sapi2llcb</i> array.
<i>Good Data</i> =	PSLLCBNULL - null value of <i>sapi2llcb</i> array index.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB where the non-null <i>tei</i> value was found.
<i>Dump 2</i> =	The LLCB involved.

Table 31 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	An invalid value of PH channel type (<i>phctype</i>) field in DCDB, associated with

	current DCCB being audited (PSdccb[dcdbind].phctype), was found by the audit.
Possible Error Effect:	This channel has wrong data from the SM (may be after recent change) and will need a refresh of data in DCDB from the SM.
Corrective Action Taken:	The audit will report the error and will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCCB in which this error was found.
<i>Bad Data =</i>	Value of the phctype field of DCDB (PSdccb[dcdbind].phctype).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	DCCB index (dcdbind).
<i>Dump =</i>	NA.

Table 32 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	At the index value equal to pix in p2aldb array in DCDB, an ALDB index value was found. The value of pix in the TEIB does not match the value of pix found in this ALDB.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the teib.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 33 Error Code: PIX2DCDB

Error Code:	PIX2DCDB
Error Description:	At the index value equal to pix in p2aldb array in the DCCB audit, a null value was found. There should have been a valid ALDB index value in this index of this array.
Possible Error Effect:	The party index value is bad. It must be set correctly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 34 Error Code: PIXBAD

Error Code:	PIXBAD
Error Description:	The audit has found some value, other than null, of pix field in DCCB. This error can occur in two different conditions. (1) When <i>isdata</i> field in DCDB says that this channel is not in service. (2) When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	Corresponds to Error Description as follows: (1) The DCCB has bad linkages or it is lost! (2) LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Pix value of the TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null pix value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.

Table 35 Error Code: PIXOUTOFRNG

Error Code:	PIXOUTOFRNG
Error Description:	The audit has found non-null, out of range pix value when we have a valid tei value. Note that pix value can be null for a valid tei value in LAPD channels. The party index value is bad. It must be set correctly.
Possible Error Effect:	The audit will report the error and return AUFALSE which will recover the TEIB.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of the TEIB in teib array in DCCB.
<i>Bad Data =</i>	Value in pix field of TEIB.
<i>Good Data =</i>	FCP_NULL - null value of pix. The good data value chosen here is null value but any value between 0 and 7 can also be a good value.
<i>Logical Index =</i>	DCCB index.
<i>Dump =</i>	TEIB block where non-null tei value was found.

Table 36 Error Code: PNT2PNT

Error Code:	PNT2PNT
Error Description:	Point to point. The value of sapi_i field of LAPD LLCB does not match the sapi_index value. To determine which type of LAPD LLCB it is, check the value of good data. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The sapi_i value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in sapi_i field of LLCB.
<i>Good Data =</i>	sapi_index - the valid index value of sapi2llcb array for LAPD LLCB.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 37 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The value of tix field of LAPD LLCB does not match the TEIB index of DCCB. To determine which type of LAPD LLCB it is, check the value of last dump variable. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The TEIB value of the LLCB attached to DCCB is bad. Bad linkage.

Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in tix field of LLCB.
<i>Good Data =</i>	teibind - the TEIB currently being audited.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in the TEIB being audited.

Table 38 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	Point to point. The value of dchan field of LAPD LLCB does not match the DCCB index (implied linkage between channel number and DCCB index). To determine which type of LAPD LLCB it is, check the value of last dump variable. Valid values are: 1 = A circuit switched call. 2 = A management call. 3 = A packet switched call.
Possible Error Effect:	The channel value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in dchan field of LLCB.
<i>Good Data =</i>	dccbind - DCCB index.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in the TEIB being audited.

Table 39 Error Code: RANGE

Error Code:	RANGE
Error Description:	The DCCB index is divisible by four and the <i>isdata</i> bit in DCDB associated with this DCCB says that this channel is in service (DBYES). Also the <i>phctype</i> field in DCDB indicates this to be an PSEOC channel. The value of FCCB LLCB index, found at the index derived from DCCB index in the PSfccbf[] array, was found to be out of range, that is greater than null value, PSLLCBNULL.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and correct the LLCB index value to its null value, PSLLCBNULL. It will also schedule the routine port conditioning of the PH involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of DCCB we are auditing.
<i>Bad Data =</i>	The value of LLCB index found at the index derived from DCCB index in the PSfccbf[] array.
<i>Good Data =</i>	The null value for the LLCB index, PSLLCBNULL.
<i>Logical Index =</i>	The index derived from DCCB index in the PSfccbf[] array.
<i>Dump =</i>	NA.

Table 40 Error Code: SAPI

Error Code:	SAPI
Error Description:	The value of sapi_i field of the LLCB does not correspond to the PSKT value. For LAPB LLCB this value should be PSPKT.

Possible Error Effect:	The sapi_i value of the LLCB attached to DCCB is bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the LLCB in which this error was found.
<i>Bad Data =</i>	Value in sapi_i field of LLCB.
<i>Good Data =</i>	PSPKT - the valid index value of sapi2llcb array for LAPB LLCB.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The llcb index.

Table 41 Error Code: SAPIBAD

Error Code:	SAPIBAD
Error Description:	The audit has found some value, other than null, at any one of the index in sapi2llcb array in TEIB in DCCB. This error can occur in two different conditions. (1) When <i>isdata</i> field in DCBD says that this channel is not in service. (2) When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	Corresponds to Error Description as follows: (1) The DCCB has bad linkages or it is lost! (2) LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	Value in sapi2llcb array index in TEIB.
<i>Good Data =</i>	PSLLCBNULL - null value of sapi2llcb array indexes.
<i>Logical Index =</i>	DCCB index.
<i>Dump 1 =</i>	TEIB block where non-null tei value was found.
<i>Dump 2 =</i>	The index of the TEIB in teib array in DCCB.
<i>Dump 3 =</i>	The index of sapi2llcb array in which non-null value was found.

Table 42 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The audit has found some value, other than null, of tei field in DCCB. This error can occur in two different conditions. (1) When <i>isdata</i> field in DCBD says that this channel is not service. (2) When it was a LAPB logical link and we found some value in tei field of TEIB whose in index (entry) in DCCB is other than zero.
Possible Error Effect:	Corresponds to Error Description as follows: (1) The DCCB has bad linkages or it is lost! (2) LAPB logical link has bad linkages.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the DCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TEIB in which this error was found.
<i>Bad Data =</i>	TEI value of the TEIB.
<i>Good Data =</i>	PSTEINULL - null value of tei.
<i>Logical Index =</i>	DCCB index.

<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 43 Error Code: TEIOUTOFRNG

Error Code:	TEIOUTOFRNG
Error Description:	The audit found an out of range TEI value in the tei field of this TEIB.
Possible Error Effect:	This TEIB had bad TEI value/binding.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of tei in TEIB.
<i>Good Data</i> =	PSTEINULL - null value of tei.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.

Table 44 Error Code: TIX

Error Code:	TIX
Error Description:	The value of tix field of the LLCB does not correspond to this TEIB index of DCCB.
Possible Error Effect:	The TEIB value of the LLCB attached to DCCB in bad. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the LLCB in which this error was found.
<i>Bad Data</i> =	Value in tix field of LLCB.
<i>Good Data</i> =	teibind - the TEIB currently being audited.
<i>Logical Index</i> =	DCCB index.
<i>Dump 1</i> =	TEIB block where non-null tei value was found.
<i>Dump 2</i> =	The index of the TEIB in teib array in DCCB.
<i>Dump 3</i> =	The llcb index.

Table 45 Error Code: TYPE

Error Code:	TYPE
Error Description:	For a LAPB LLCB, the <i>I2_I3inh</i> field in TEIB was set to DBYES for one of the following channel types when it should have been set to DBNO. The channel types include PSDS LINK, PSDP LINK, PSDSP LINK, and PSDP MP.
Possible Error Effect:	This TEIB has <i>I2_I3inh</i> set incorrectly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the TEIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the TEIB in which this error was found.
<i>Bad Data</i> =	Value of <i>I2_I3inh</i> in TEIB.
<i>Good Data</i> =	DBNO - null value of <i>I2_I3inh</i> .
<i>Logical Index</i> =	DCCB index.
<i>Dump</i> =	The index of the TEIB in teib array in DCCB.

DCDB-A

Software Release: 5E10 - 5E11

1. D-CHANNEL DATA BLOCK (DCDB) AUDIT

The D-channel data block audit is assigned to each channel when it is turned on. The audit has linkages with access line data block (ALDB) and packet pipe data block (PPDB). This audit checks the linkages with the ALDB and PPDB data structures only. The audit does not make any semantic checks, since DCDB is a write protected data structure. The linkage checks are made to guard against any errors occurred during the assignment; and the release of ALDB from and to the free list, during which the write protection is removed temporarily.

2. DATA STRUCTURES

PSDCDB, PSALDB, and PSPPDB.

3. AUDIT CHECKS

The DCDB audit checks the linkages with ALDB and PPDB.

4. ERROR HANDLING

If the audit finds an error with an ALDB linkage, that link is cleared and routine port conditioning is called. If the audit finds an error with the semantic linkage to PPDB (PSdcdB[].member), or if the PPDB linkage back to the DCDB (PSppdb[].channel) or the PSppdb[].isdata is "no", the DCDB's semantic linkage to the PPDB is nulled and it allows PPDB recovery to call the routine port conditioning.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	Either the channel value in PPDB (PSppdb[PSdcdB[].member].channel) does not match the value of dcdB index or the PPDB is not in use (PSppdb[].isdata = DBNO) and the DCDB is in use (PSdcdB[].isdata = DBYES).
Possible Error Effect:	Because the PPDB and DCDB array entries are inconsistent, packets will not be received by the packet pipe.
Corrective Action Taken:	The audit will report the error and null out the member . DCDB recovery will not be called but the PPDB related audit will recover all associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	The channel value in the PPDB (PSppdb[].channel).
<i>Good Data =</i>	The index into the PPDB (member).
<i>Logical Index =</i>	The index into the DCDB (channel).
<i>Dump 1 =</i>	The contents of the DCDB array entry.
<i>Dump 2 =</i>	The contents of the PPDB array entry.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	When we went to ALDB from the aldb index value found in the 0th entry of the p2aldb array in LAPB DCDB, the channel value in ALDB does not match the value of dcdB index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	Value in 0th entry of p2aldb array of DCDB.
<i>Good Data</i> =	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB (channel) is good data.
<i>Logical Index</i> =	DCDB index (channel).
<i>Dump</i> =	NA.

Table 3 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	For LAPD DCDB, the value in any of the entry of the p2aldb array has out of range ALDB index (greater than PSALDBNULL) value.
Possible Error Effect:	The p2aldb array entry has bad value. This could have been either wild write or because the DCDB data structure was not cleared properly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in removing and restoring this channel.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	Value in p2aldb array entry of DCDB.
<i>Good Data</i> =	PSALDBNULL - null value for ALDB or any valid ALDB index value.
<i>Logical Index</i> =	DCDB index (channel).
<i>Dump</i> =	NA.

Table 4 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The index into PPDB (PSdcbd[].member) is out of range.
Possible Error Effect:	Because of the bad linkage between the PPDB and DCDB array entries, packets will not be received by the packet pipe.
Corrective Action Taken:	The audit will report the error and null out the member . DCDB recovery will not be called but the PPDB related audit will recover all associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	The index into the PSppdb[] (PSdcbd[].member).
<i>Good Data</i> =	The in-use data in the DCDB (PSdcbd[].isdata).
<i>Logical Index</i> =	The index into the DCDB (channel).
<i>Dump</i> =	The contents of the DCDB array entry.

Table 5 Error Code: MKNULL

Error Code:	MKNUL
Error Description:	For packet pipes, all ALDB indices in the DCDB (PSdcbd[].p2aldb[]) should be null, but they are not all null in this DCDB.
Possible Error Effect:	PSdcbd[].p2aldb[] should not have any linkages to the ALDB. If the DCDB owns ALDB's, it is tying up resources that could be used by other users.
Corrective Action Taken:	The audit will report the error and null out all entries in PSdcbd[].p2aldb[]. The channel is left intact.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	The ALDB index in the DCDB (PSdcbd[].p2aldb[]).
<i>Good Data</i> =	The array index of p2aldb array in the DCDB where the bad value of ALDB index was found.
<i>Logical Index</i> =	The index into the DCDB (channel).
<i>Dump</i> =	The contents of the DCDB array entry.

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The "isdata" field of DCDB says that this channel is turned off (DBNO) and we found a value other than null value in any of the p2aldb array entry.
Possible Error Effect:	The p2aldb array entry has bad value. This could have been either wild write or

	because the DCDB data structure was not cleared properly.
Corrective Action Taken:	The audit will report the error and set the entry to PSALDBNULL value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in p2aldb array entry of DCDB.
<i>Good Data =</i>	PSALDBNULL - null value for ALDB.
<i>Logical Index =</i>	1DCDB index (channel).
<i>Dump =</i>	NA.

Table 7 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	For LAPB DCDB, we should have a valid ALDB index only at the 0th entry of the p2aldb array in DCDB. All other entries, 1 through 7, should have a null value of ALDB index, PSALDBNULL. The audit has found a non null entry in any of these indexes.
Possible Error Effect:	The p2aldb array entry has bad value. This could have been either wild write or because the DCDB data structure was not cleared properly.
Corrective Action Taken:	The audit will report the error and set the entry to PSALDBNULL value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in p2aldb array entry of DCDB.
<i>Good Data =</i>	PSALDBNULL - null value for ALDB.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 8 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of ALDB index at 0th entry of p2aldb array for a LAPB DCDB is either null or out of range.
Possible Error Effect:	The p2aldb array entry has bad value. For LAPB channel, you should have a valid ALDB index value at 0th slot of the p2aldb array. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will remove and restore this channel.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in 0th entry of p2aldb array for LAPB DCDB.
<i>Good Data =</i>	0. The value chosen here is 0 but any value between 0 and 127 will be a good value.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 9 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	We have a bad "phctype" value in DCDB. This value is not defined as phctype.
Possible Error Effect:	The channel had bad phctype value.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value of phctype in DCDB.
<i>Good Data =</i>	0. The value chosen here is zero but any valid value of phctype is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 10 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	When we went to ALDB from the aldb index value found in the 0th entry of the

	p2aldb array in LAPB DCDB, the pix value in ALDB does not match the value of p2aldb array index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in 0th entry of p2aldb array of DCDB.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 11 Error Code: PNT2PNT

Error Code:	PNT2PNT
Error Description:	When we went to ALDB from a valid aldb index value found in any entry of the p2aldb array in LAPD DCDB, the pix value in ALDB does not match the value of the index of the p2aldb array where we found this ALDB index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This value is usually an address but it is used here to dump the value of aldb index which is bad.
<i>Bad Data =</i>	The array index of p2aldb array in DCDB where we found the bad value of ALDB index.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 12 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	When we went to ALDB from a valid aldb index value found in any entry of the p2aldb array in LAPD DCDB, the channel value in ALDB does not match the value of dcd index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value of ALDB index in p2aldb array entry.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB (channel) is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

DCDB-B

Software Release: 5E12 and later

1. D-CHANNEL DATA BLOCK (DCDB) AUDIT

The D-channel data block audit is assigned to each channel when it is turned on. The audit has linkages with access line data block (ALDB) and packet pipe data block (PPDB). This audit checks the linkages with the ALDB and PPDB data structures only. The audit does not make any semantic checks, since DCDB is a write protected data structure. The linkage checks are made to guard against any errors occurred during the assignment; and the release of ALDB from and to the free list, during which the write protection is removed temporarily.

2. DATA STRUCTURES

PSDCDB, PSALDB, and PSPPDB.

3. AUDIT CHECKS

The DCDB audit checks the linkages with ALDB and PPDB.

4. ERROR HANDLING

If the audit finds an error with an ALDB linkage, that link is cleared and routine port conditioning is called. If the audit finds an error with the semantic linkage to PPDB (PSdcdB[].member), or if the PPDB linkage back to the DCDB (PSppdb[].channel) or the PSppdb[].isdata is "no", the DCDB's semantic linkage to the PPDB is nulled and it allows PPDB recovery to call the routine port conditioning.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	Either the channel value in PPDB (PSppdb[PSdcdB[].member].channel) does not match the value of dcdB index or the PPDB is not in use (PSppdb[].isdata = DBNO) and the DCDB is in use (PSdcdB[].isdata = DBYES).
Possible Error Effect:	Because the PPDB and DCDB array entries are inconsistent, packets will not be received by the packet pipe.
Corrective Action Taken:	The audit will report the error and null out the member . DCDB recovery will not be called but the PPDB related audit will recover all associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	The channel value in the PPDB (PSppdb[].channel).
<i>Good Data =</i>	The index into the PPDB (member).
<i>Logical Index =</i>	The index into the DCDB (channel).
<i>Dump 1 =</i>	The contents of the DCDB array entry.
<i>Dump 2 =</i>	The contents of the PPDB array entry.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	When we went to ALDB from the aldb index value found in the 0th entry of the p2aldb array in LAPB DCDB, the channel value in ALDB does not match the value of dcdB index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	Value in 0th entry of p2aldb array of DCDB.
<i>Good Data</i> =	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB (channel) is good data.
<i>Logical Index</i> =	DCDB index (channel).
<i>Dump</i> =	NA.

Table 3 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	For LAPD DCDB, the value in any of the entry of the p2aldb array has out of range ALDB index (greater than PSALDBNULL) value.
Possible Error Effect:	The p2aldb array entry has bad value. This could have been either wild write or because the DCDB data structure was not cleared properly.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in removing and restoring this channel.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	Value in p2aldb array entry of DCDB.
<i>Good Data</i> =	PSALDBNULL - null value for ALDB or any valid ALDB index value.
<i>Logical Index</i> =	DCDB index (channel).
<i>Dump</i> =	NA.

Table 4 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The index into PPDB (PSdcbd[].member) is out of range.
Possible Error Effect:	Because of the bad linkage between the PPDB and DCDB array entries, packets will not be received by the packet pipe.
Corrective Action Taken:	The audit will report the error and null out the member . DCDB recovery will not be called but the PPDB related audit will recover all associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	The index into the PSppdb[] (PSdcbd[].member).
<i>Good Data</i> =	The in-use data in the DCDB (PSdcbd[].isdata).
<i>Logical Index</i> =	The index into the DCDB (channel).
<i>Dump</i> =	The contents of the DCDB array entry.

Table 5 Error Code: MKNULL

Error Code:	MKNUL
Error Description:	For packet pipes, all ALDB indices in the DCDB (PSdcbd[].p2aldb[]) should be null, but they are not all null in this DCDB.
Possible Error Effect:	PSdcbd[].p2aldb[] should not have any linkages to the ALDB. If the DCDB owns ALDB's, it is tying up resources that could be used by other users.
Corrective Action Taken:	The audit will report the error and null out all entries in PSdcbd[].p2aldb[]. The channel is left intact.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the DCDB in which this error was found.
<i>Bad Data</i> =	The ALDB index in the DCDB (PSdcbd[].p2aldb[]).
<i>Good Data</i> =	The array index of p2aldb array in the DCDB where the bad value of ALDB index was found.
<i>Logical Index</i> =	The index into the DCDB (channel).
<i>Dump</i> =	The contents of the DCDB array entry.

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The "isdata" field of DCDB says that this channel is turned off (DBNO) and we found a value other than null value in any of the p2aldb array entry.
Possible Error Effect:	The p2aldb array entry has bad value. This could have been either wild write or

	because the DCDB data structure was not cleared properly.
Corrective Action Taken:	The audit will report the error and set the entry to PSALDBNULL value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in p2aldb array entry of DCDB.
<i>Good Data =</i>	PSALDBNULL - null value for ALDB.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 7 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	For LAPB DCDB, we should have a valid ALDB index only at the 0th entry of the p2aldb array in DCDB. All other entries, 1 through 7, should have a null value of ALDB index, PSALDBNULL. Audit has found a non null entry in any of these indexes.
Possible Error Effect:	The p2aldb array entry has bad value. This could have been either wild write or because the DCDB data structure was not cleared properly.
Corrective Action Taken:	The audit will report the error and set the entry to PSALDBNULL value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in p2aldb array entry of DCDB.
<i>Good Data =</i>	PSALDBNULL - null value for ALDB.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 8 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of ALDB index at 0th entry of p2aldb array for a LAPB DCDB is either null or out of range.
Possible Error Effect:	The p2aldb array entry has bad value. For LAPB channel, you should have a valid ALDB index value at 0th slot of the p2aldb array. Bad linkage.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will remove and restore this channel.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in 0th entry of p2aldb array for LAPB DCDB.
<i>Good Data =</i>	0. The value chosen here is 0 but any value between 0 and 127 will be a good value.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 9 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	We have a bad "phctype" value in DCDB. This value is not defined as phctype.
Possible Error Effect:	The channel had bad phctype value.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value of phctype in DCDB.
<i>Good Data =</i>	0. The value chosen here is zero but any valid value of phctype is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 10 Error Code: PIX2ALDB

Error Code:	PIX2ALDB
Error Description:	When we went to ALDB from the aldb index value found in the 0th entry of the

	p2aldb array in LAPB DCDB, the pix value in ALDB does not match the value of p2aldb array index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value in 0th entry of p2aldb array of DCDB.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 11 Error Code: PNT2PNT

Error Code:	PNT2PNT
Error Description:	When we went to ALDB from a valid aldb index value found in any entry of the p2aldb array in LAPD DCDB, the pix value in ALDB does not match the value of the index of the p2aldb array where we found this ALDB index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This value is usually an address but it is used here to dump the value of aldb index which is bad.
<i>Bad Data =</i>	The array index of p2aldb array in DCDB where we found the bad value of ALDB index.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

Table 12 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	When we went to ALDB from a valid aldb index value found in any entry of the p2aldb array in LAPD DCDB, the channel value in ALDB does not match the value of dcd index.
Possible Error Effect:	The p2aldb array entry has bad value. We have a bad link between LAPB DCDB and ALDB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will result in calling the idling function and the channel will be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the DCDB in which this error was found.
<i>Bad Data =</i>	Value of ALDB index in p2aldb array entry.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid ALDB index and has correct link with DCDB (channel) is good data.
<i>Logical Index =</i>	DCDB index (channel).
<i>Dump =</i>	NA.

DCHNPRI

Software Release: 5E10 - 5E13

1. PRIMARY RATE INTERFACE D-CHANNEL (DCHNPRI)

The primary rate interface D-channel (DCHNPRI) audit audits the DDPRISTATE and DDPRIMODE data delivery(DD) keys, used for the non-facility associated signaling (NFAS) and D-channel backup (DCBU) features available on the 5ESS[®] switch. These DD keys are used for all non operator services position system (OSPS) PRI. A tuple exists in the relation RLDCHNPRI for every non-OSPS PRI D-channel. An NFAS provides a Bellcore compatible primary rate interface (PRI) defined as $nB + D$, where $n=1-479$. PRI is a collection of B-channels and a signaling D-channel. All B-channels are controlled by that D-channel. Now there are two types of non-OSPS PRI. In the classic definition, n is a maximum of 23 and all B-channels are to be on the same digital signal 1 (DS1) facility known as digital facilities interface (DFI). This is called a custom PRI. In a standard PRI definition, n is a maximum of 479 and the B-channels can spread over multiple facilities with maximum up to 20 facilities within a single switching module (SM). The D-Channels must reside on the 24th channel of a DFI and must be on the same SM as the B-channels that it is providing signaling for (such as, all of the B-channels and its signaling D-Channels are in the same PRI group).

DCBU is a Bellcore compatible PRI with $mB + 2D$, where $m = 1 - 478$, and there are 2 D channels. One D channel is designated as "active" and the other D channel is designated as "standby". Should the "active" D channel fail due to hardware, Layer 2 or Layer 3, then the "standby" D channel should become the "active" D channel. Both D-channels must be on the same SM as the B-channels that it is providing signaling for (such as, all of the B-channels and its signaling D-channels are in the same PRI group).

Both NFAS and DCBU apply only to PRIs that can only interface to CPEs (includes the 5ESS[®] switch "User side PRIs") which are compatible with Bellcore's TR-1268. The existing $23B + D$ PRIs (that is, Lucent custom PRI) will not support NFAS and DCBU.

The DCHNPRI audit runs in the OSDS-M environment in the segmented mode. It audits the DDPRISTATE and DDPRIMODE keys designed to help determine the status of the whole PRI group on an SM for the OP:OFFNORM,PRIGRP command.

2. DATA STRUCTURES

There are three main data structures involved in this audit.

- The first major data structure for this audit is the RLDCHNPRI relation. It is a static, access type of hashed relation keyed by **dport**. For a given D-channel, this relation is used to find a PRI group or information about that D-channel that pertains to PRI group. Of these information, this audit uses the information about the index in to the DD keys(defined later in this section) for this D-channel, represented by the field **dchn_idx**. This is a semantic one to one linkage to both the DDPRISTATE and DDPRIMODE keys.
- New DD keys unique to each SM are defined for this feature to keep track of each possible PRI D-channel. The two items of information kept on each possible PRI D-channel are in service (IS)/out of service (OOS) state (DD key named DDPRISTATE) and the AUTO/MANUAL qualifier (DD key named DDPRIMODE). This data will be kept in two longs with only 20 bits in use. This DD structure is transported to the administrative module (AM) from each SM. In the AM there is a unique representation of this data for each individual SM. The **dchn_idx** is used to correlate the abstract index in to the DD key structure to a D-channel through the relation RLDCHNPRI in the SM and RLSDCHN in the AM.

Representative values for the DD keys are shown below:

DDPRISTATE	DDPRIMODE	STATE
0	X	Everything is OK

1	0	PRI OOS-AUTO
1	1	PRI OOS-MAN

X = Don't care.

(See Figure 1.)

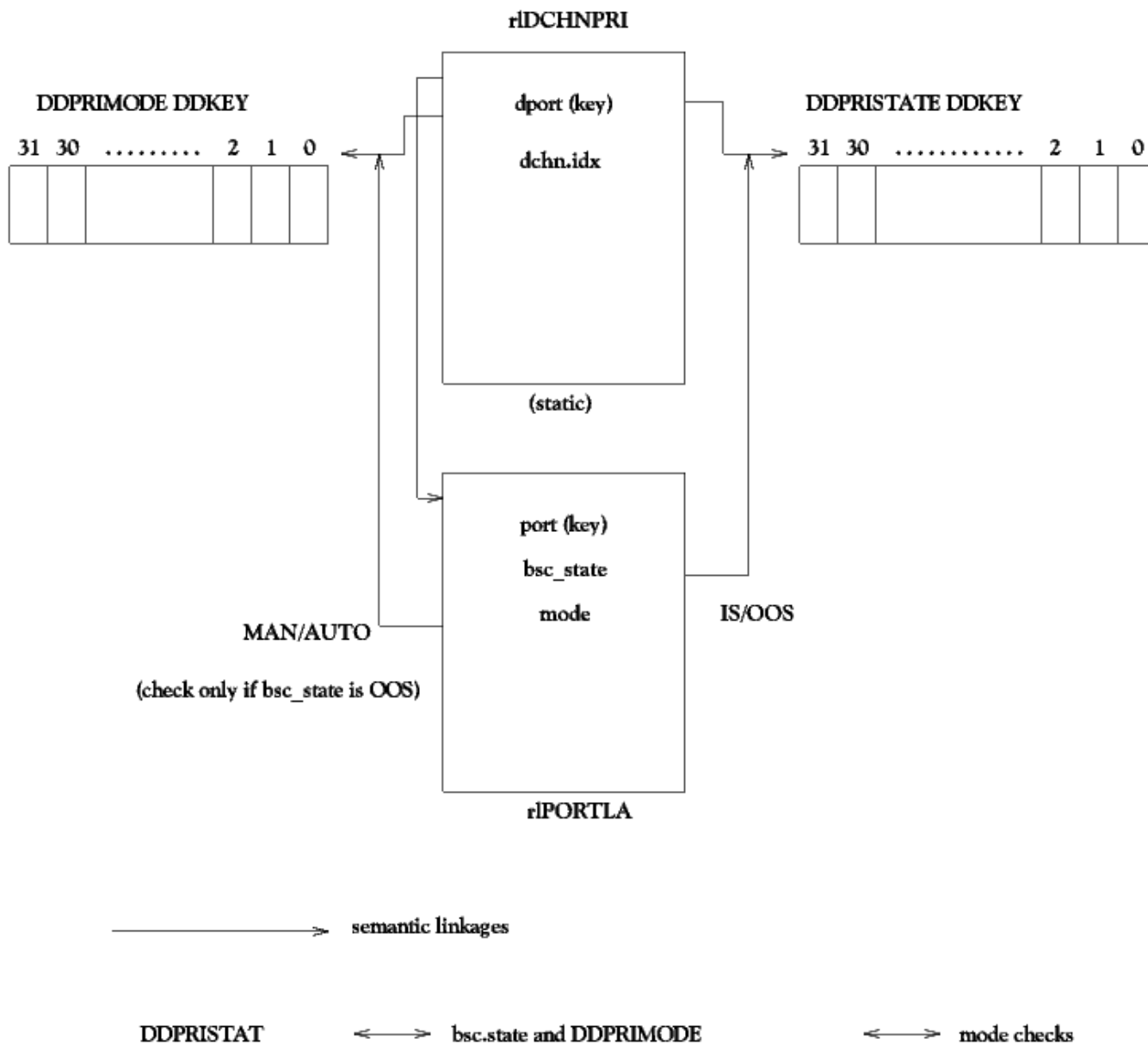


Figure 1 (DCHNPRI-1) DDPRISTATE and DDPRIMODE Keys

3. AUDIT CHECKS

The DCHNPRI audit invokes the DRSTATIC driver to audit the DDPRISTATE and DDPRIMODE keys. It also uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given.

BADUSE = Invalid value of the **mode** field of the rPORTLA tuple.

- MODE = Check the value of DDPRIMODE key's bit against the value calculated by audit.
- NOPORTLA = Check that audit got correct rIORTLA tuple for the D port.
- RANGE = Make range check on the dchn_idx field.
- STATE = Check the value of DDPRISTATE key's bit against the value calculated by audit.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	Audit found an error condition where the value of mode field of the rIORTLA tuple of the D port has a value other than MAN or AUTO when the bsc_state field of the rIORTLA tuple is OOS.
Possible Error Effect:	Varied according to conditions.
Corrective Action Taken:	The audit will report the error and return causing PORTLA audit to run as a related audit to this audit and correct the error condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The module field of the dport field of rIDCHNPRI tuple.
<i>Bad Data =</i>	The mode field of the rIORTLA tuple of the D port.
<i>Good Data =</i>	The value of the bit field in the DDPRIMODE DD key related to this D channel.
<i>Logical Index =</i>	The member field of the dport field of rIDCHNPRI tuple.
<i>Dump =</i>	The rIORTLA tuple of the D port.

Table 2 Error Code: MODE

Error Code:	MODE
Error Description:	The value of the bit related to the D channel being audited in the DDPRIMODE DD key does not match the value calculated by audit based on the mode field of the rIORTLA. The bsc_state of the rIORTLA tuple is in OOS state.
Possible Error Effect:	Incorrect information may be conveyed when inquiring about the status of this D-channel by the OP:OFFNORM-PRI input message.
Corrective Action Taken:	The bit field in DDPRIMODE will be updated with the value calculated by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The value of the bit field of the D-channel calculated by audit based on the value of mode field of the rIORTLA tuple.
<i>Bad Data =</i>	The value of the bit field of the D-channel read by the audit from the DDPRIMODE DD key.
<i>Good Data =</i>	The number of the bit related to the D-channel in the DDPRIMODE key. The dchn_idx field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	The member field of the dport field of rIDCHNPRI tuple.
<i>Dump =</i>	The rIORTLA tuple of the D port.

Table 3 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	Audit tried to read an rIORTLA tuple with its key equal to the member field of the dport field of the static rIDCHNPRI tuple, and failed.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned

	resources represented by this rIORTLA tuple, operations may be disrupted.
Corrective Action Taken:	The audit will report the error and return causing PORTLA audit to run as a related audit to this audit and correct the error condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple that audit could not read.
<i>Bad Data =</i>	The key of the rIORTLA tuple that audit could not read, port field.
<i>Good Data =</i>	The module field of the dport field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	The member field of the dport field of rIDCHNPRI tuple.
<i>Dump =</i>	NA.

Table 4 Error Code: RANGE

Error Code:	RANGE
Error Description:	The value of the dchn_idx field of the static rIDCHNPRI tuple failed the range check. Audit will continue to take this error until this problem is corrected.
Possible Error Effect:	Incorrect information may be conveyed when inquiring about the status of this D-channel by OP:OFFNORM-PRI input message.
Corrective Action Taken:	Audit will report the error and return waiting for craft to correct the ODD problem.
Manual Action Required:	The valid range for the dchn_idx field is 1 through MAXDCHANS. Update the rIDCHNPRI tuple using the dport field whose two fields are dumped below in good data and logical key fields and modify the value of dchn_idx field to be within the valid range. As to what should be the correct value, refer to terminal maintenance area and triggers.
Dump Description:	
<i>Error Address =</i>	1, but it is not applicable.
<i>Bad Data =</i>	The value of the dchn_idx field of the rIDCHNPRI tuple.
<i>Good Data =</i>	The module field of the dport field of rIDCHNPRI tuple.
<i>Logical Index =</i>	The member field of the dport field of rIDCHNPRI tuple.
<i>Dump =</i>	NA.

Table 5 Error Code: STATE

Error Code:	STATE
Error Description:	The value of the bit related to the D-channel being audited in the DDPRISTATE DD key does not match the value calculated by audit based on the bsc_state field of the rIORTLA.
Possible Error Effect:	Incorrect information may be conveyed when inquiring about the status of this D-channel by OP:OFFNORM-PRI input message.
Corrective Action Taken:	The bit field in DDPRISTATE DD key will be updated with the value calculated by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The value of the bit field of the D-channel calculated by audit based on the value of bsc_state field of the rIORTLA tuple.
<i>Bad Data =</i>	The value of the bit field of the D-channel read by the audit from the DDPRISTATE DD key.
<i>Good Data =</i>	The number of the bit related to the D-channel in the DDPRISTATE key. The dchn_idx field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	The member field of the dport field of rIDCHNPRI tuple.
<i>Dump =</i>	The rIORTLA tuple of the D port.

DD-A

Software Release: 5E10 - 5E11

1. DATA DELIVERY (DD) AUDIT

The data delivery (DD) mechanism and audit covers data in the operational kernel process (OKP) and switch maintenance kernel process (SMKP) environments in the administrative module (AM), communications module processor (CMP) as well as the operating system for distributed switching-switching modules (OSDS-SMs) environment. The audit verifies the consistency of different data delivery data against other static data, data delivery, and global data. The audit reports errors it detects and makes the slave data consistent with available master data. If no master data exists, the data delivery data is set to a safe state. Data delivery data in incorrect states could disable call processing and cause problems for maintenance, billing, and growth; hence the need for the DD audit.

The DD audit runs in the CMP, OKP, SM, and SMKP environments in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DD audit checks the consistency of a number of different DD data structures. The term "DD key" refers to an identifier for information maintained by data delivery. Some of these data structures exist in the AM as AM-only data, some exist in the AM on a per SM basis, some exist in the AM on a per message class basis, and some of the data exists only in the SMs. The DD data structures checked are referenced by the following DD keys:

DDAPDLDSL	DDDASCDLL	DDMGPRTBL	DDRSMISOL
DDASPTRIGL	DDDIOPMODEL	DDPCONFIGSTL	DDRSMSEP
DDAUEWRKL	DDEQSTATL	DDPSSWSTATL	DDSASWC
DDCM_VINTL	DDHOSTNUML	DDPVNALWL	DDSCMISOL
DDCNIMINL	DDIMTYPEL	DDRISOLL	DDSINIT
DDCNIOOSL	DDMATECMPL	DDRMSWCKL	DDSMIN
DDCTS0L	DDMGDSCBL	DDRSEPL	DDSREX
DDCTS1L	DDMGLOGBL	DDRSMINITL	

3. AUDIT CHECKS

The audit makes the following checks and takes the specified actions.

Check	Environment	Action
DDAPDLDSL	SM	Checks the new data delivery key DDAPDLDSL against the AP link status table PSaplst[]. DDAPDLDSL provides a summary of the status of the AP data link digital subscribe lines (DSL's) for a given SM. PSaplst[] contains information on the status of each of the individual AP data link DSL's. If the two values found do not match, an audit error is reported and DDAPDLDSL will be updated to agree with the information found in PSaplst[].
DDASPTRIG	OKP	Checks DD data associated with key DDASPTRIG against the data against the rIDS_APP tuple referenced by the key of CCAINR01. When the rIDS_APP tuple is not found, the DD data should indicate clear. When the rIDS_APP tuple is found and the global parameter is DBPVN_SSP, the DD data should be set.
DDAUEWRK	SMKP	Checks DD data associated with key DDAUEWRK against a global variable, AUEleact, which is maintained by audit control. When the global variable is set, the DD data should be set and when the global variable is clear, the DD data should indicate clear. If an inconsistency is detected, the DD data will be updated to be consistent with the global variable.
DDCM_VINT	OSDS-M	Checks DD data associated with key DDCM_VINT against the SM's office dependent data (ODD) global variable Glcm_vint. If an inconsistency is detected, the DD data will be updated to be consistent with the global variable. An audit error concerning this data is expected during the conversion of a communication module, model 1 (CM1) office to a CM2 office.
DDCNIMIN	OKP	Checks DD data associated with key DDCNIMIN against the DD data associated

		with key DDSMIN. If DDSMIN is set to indicate minmode, then DDCNIMIN should be set to indicate common network interface (CNI) minmode. If there is an inconsistency, both DD data will be set to not minmode.
DDCNIOOS	OKP	Checks DD data associated with key DDCNIOOS against the ODD variable Glccs. If an office is unequipped with CNI, the DD data should be set indicating that CNI is disabled. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDCTS0	OKP	Checks DD data associated with key DDCTS0 against the even CTS link number stored in MODATT. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDCTS1	OKP	Checks DD data associated with key DDCTS1 against the odd CTS link number stored in MODATT. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDDIOPMODE	OKP	Checks DD data associated with key DDDIOPMODE. If the DD data is set to indicate disk independent mode, a message will be sent to application integrity monitor (AIM) to verify a valid disk independent mode. This will be done by AIM checking the essential disks statuses in the equipment configuration database (ECD). If an inconsistency exists, AIM will reset the DD data to be consistent with the ECD.
DDEQPROCS	OKP/CMP	Checks DD data associated with the key DDEQPROCS against the processors equipage status maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data is made consistent with the ODD.
DDEQSTAT	OKP/CMP	Checks DD data associated with key DDEQSTAT against the SM equipage status maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD. (The DD data here is different than that associated with key DDEQIMS.)
DDHOSTNUM	OKP/CMP	Checks DD data associated with key DDHOSTNUM against the host module number maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDIMTYPE	OKP	Checks DD data associated with key DDIMTYPE against the SM type maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDMATECMP	SMKP/CMP	Checks DD data associated with key DDMATECMP against the hostnum field maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDMGDSCB	OKP	Checks DD data associated with key DDMGDSCB against DD data associated with keys DDMGLOGB and DDMGPRTB. If any of the DD data indicates to log or print messages, then the DD data for discard should not be set. If an inconsistency exists, the discard data will be set to not discard.
DDMGLOGB	OKP	This represents the DD data for logging messages. See the check for DDMGDSCB.
DDMGPRTB	OKP	This represents the DD data for printing messages. See the check for DDMGDSCB.
DDPCONFIGST	SMKP	Checks the key DDPCONFIGST for soft switch status by going off of the PCD state for the unit. In the case of a program update, the soft switch periodically resets the key so that the user knows that the program update uses the key.
DDPSSWSTAT	SMKP	Checks DD data associated with key DDPSSWSTAT against the soft switch status. If an inconsistency exists between the soft switch status and the PCD, the DD audit will update the DD key with the PCD information.
DDPVNALW	OKP	Checks DD data associated with key DDPVNALW against the existence of an rIDS_APP tuple referenced by the key of CCPVN (when Glpvn_ofc is set to DBPVN_SSP) or CCNS800. When the rIDS_APP tuple is not found, the DD data should indicate clear. When the rIDS_APP tuple is found and Glpvn_ofc is DBPVN_SSP or Glssp is DBYES, the DD data should be set.
DDRISOL	OSDS-M	Checks are done in a remote switching module (RSM) to verify that the separation information (DDRSEP) about an RSM in a multimodule site is consistent with the corresponding master data kept in DDRISOL.
DDRMSWCK	OSDS-M	Checks DD data associated with key DDRMSWCK against DDSASWCK data. If DDSASWCK data is off-normal, DDRMSWCK data should be off-normal.
DDRSEP	OSDS-M	The data in DDRSEP is the master copy representing the separation status of all

		the RSMs in a multimodule site it is arranged on a per module basis. The value in DDRSMSEP is compared against this data, and used in correcting any inconsistencies.
DDRSMINIT	OSDS-M	Checks are done to verify that the module initialization information (DDRSMINIT) about a module in a multimodule site is consistent with the corresponding master data kept in DDSINIT. When an inconsistency occurs, the DDRSMINIT status for a given module is set to the value of DDSINIT.
DDRSMISOL	OSDS-M	Checks are done to verify that the isolation status (DDRSMISOL) about a module in a multimodule site is consistent with the master status kept in DDRISOL. When an inconsistency occurs, the DDRSMISOL status for a given module is set to the value of DDRISOL.
DDRSMSEP	OSDS-M	The data in this 10-bit word is arranged based on the number RSMs in a multimodule site. The separation status of each RSM is maintained here. Checks are done to verify that the separation status (DDRSMSEP) is consistent with the master status kept in DDRSEP. When an error is found, the wrong bit in DDRSMSEP is set to the value of DDRSEP.
DDSASWCK	OKP	Checks DD data associated with key DDSASWCK against DDRMSWCK data. If DDRMSWCK data is off-normal, DDSASWCK data should be off-normal.
DDSINIT	OSDS-M	Checks DD data associated with key DDSINIT against the global variable INmode. If an inconsistency exists, the DD data will be made consistent with the global variable.
DDSMCOUNT	OKP	Checks the new data delivery key DDSMCOUNT against the actual number of SMs (including special growth). If the two values do not match, an audit error is reported, and the DDSMCOUNT is updated with the correct value.
DDSMIN	SM	Checks SM DD data associated with DDSMIN against the global variable GltHrtl. If the DD data indicates minmode and the global variable does not, the DD data will be set to not minmode.
DDSREX	OKP	Checks DD data associated with key DDSREX against the inhibit allow status of routine exercises maintained in a global routine exercise table (DGldb[module].rxnhlst. enable1). If an inconsistency exists, the DD data will be made consistent with the global table.
DDSCMISOL	SMKP	Checks DD data associated with the key DDSCMISOL against the status of the CM isolation request DD key DDRCMISOL and either the SMARS job queue or the SMrsmode AM/CM re-synchronization flag. If an inconsistency exists, the DD data is written to normal.

4. ERROR HANDLING

If the ASP service features find errors, the data delivery bit is updated to be consistent with the static relation RLDS_APP.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_DDASPTRIG

Error Code:	ACT_DDASPTRIG
Error Description:	The data delivery data from the key DDASPTRIG is set to indicate ASP 0.1 activation. The static data in rlds_APP does not find a match using a key of CCAINR01, so data delivery data should indicate ASP 0.1 is de-activated.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The data delivery data DDASPTRIG is set to indicate ASP 0.1 de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The database return from reading the RLDS_APP relation.

<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 2 Error Code: ACT_DDPVNALW

Error Code:	ACT_DDPVNALW
Error Description:	The value of the data delivery key, DDPVNALW, is set to indicate ASP activation. A match in static data rIDS_APP does not exist for either of the following keys: (1) A key of CCPVN when Glpvn_ofc is set to DBPVN_SSP, and (2) A key of CCNS800. Therefore, DDPVNALW should not be set.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The value of DDPVNALW is cleared to indicate ASP de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return code from DD read.
<i>Good Data</i> =	The database return from reading the RLDS_APP relation.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 3 Error Code: BAD_DDTBL

Error Code:	BAD_DDTBL
Error Description:	An invalid state field (not AUDDDO, AUDDSKIP or AUDDDONE) in the data delivery audit table (AUdd1tbl[]), has been detected.
Possible Error Effect:	The data delivery audit may not be able to audit all the data items it is supposed to.
Corrective Action Taken:	Reinitialize the index (AUddindx), and start from the beginning of the audit the next time it is entered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local check table, AUdd1tbl[].
<i>Bad Data</i> =	The state element for the indicated table entry (AUdd1tbl[AUddindx].state).
<i>Good Data</i> =	Maximum array index which when added to 1 equals the number of entries in the table (AUdd flag).
<i>Logical Index</i> =	Index into the table (AUddindx).
<i>Dump</i> =	The data delivery audit table (AUdd1tbl[]) of entries for the indicated processor.

Table 4 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	<i>In OKP or CMP:</i> The bit in DDeqmhppe[] associated with a particular switching module (SM) is set incorrectly. That is, either the bit indicates message handler (MH) connectivity for that SM when it should not (the bit is set) or the bit indicates no MH connectivity for that SM when the SM should have it (the bit is clear). See the dump description to determine which of the following cases applies [note that the "rIMODATT tuple" is the tuple associated with the particular SM (rIMODATT.module)]: • The bit is set but no rIMODATT tuple exists for the SM. • The bit is set but at least one of the rIMODATT <i>ql_ntwk0</i> or <i>ql_ntwk1</i> fields has value DBNO. • The bit is clear but both of the rIMODATT <i>ql_ntwk0</i> and <i>ql_ntwk1</i> fields

	have value DBYES. <i>In an SM-2000:</i> The SM's copy of DDeqmhpipeline[] is inconsistent with the <i>qends[]</i> fields in the rIQENDPTS tuples where rIQENDPTS.qservice = QLOSDS (representing MH connectivity) and rIQENDPTS.qserv_num = 0 (the default value, used for non-CCS connectivity). Specifically, for each bit in DDeqmhpipeline[], that bit should be set if, and only if, the corresponding bits are set in rIQENDPTS.qends[] for both networks (rIQENDPTS.qnetwork = QLNET0 and QLNET1).
Possible Error Effect:	If the bit is set when it should be clear, quad linked packet switching (QLPS) network traffic may be routed to an SM unable to accept it. If the bit is clear when it should be set, the QLPS network may be degraded because it is unable to route to the associated SM.
Corrective Action Taken:	<i>In OKP or CMP:</i> The audit triggers an update of the bit in DDeqmhpipeline[] and any other DD data associated with the SM. <i>In SM-2000:</i> The audit triggers a reinitialization of DDeqmhpipeline[].
Manual Action Required:	None.
Dump Description:	
<i>In OKP or CMP:</i>	
<i>Error Address =</i>	SM equipage status (rIMODATT.eqstat). A value of SMUNEQ may indicate that no rIMODATT tuple exists for this SM.
<i>Bad Data =</i>	rIMODATT.ql_nwkw0.
<i>Good Data =</i>	rIMODATT.ql_nwkw1.
<i>Logical Index =</i>	The associated SM number (used in rIMODATT.module when trying to read the rIMODATT tuple).
<i>Dump =</i>	NA.
<i>In an SM-2000:</i>	
<i>Error Address =</i>	Current entry in rIQENDPTS.qends[] for network QLNET0.
<i>Bad Data =</i>	Current entry in DDeqmhpipeline[] being checked.
<i>Good Data =</i>	Current entry in rIQENDPTS.qends[] for network QLNET1.
<i>Logical Index =</i>	Index of the entry in DDeqmhpipeline[] being checked. That is, the index of the first entry to have an error.
<i>Dump 1 =</i>	DDeqmhpipeline[].
<i>Dump 2 =</i>	The rIQENDPTS tuple associated with network QLNET0.

Table 5 Error Code: CLIDAT

Error Code:	CLIDAT
Error Description:	An attempt failed to obtain a rCLIDAT tuple with a given cluster ID. The audit has not taken any recovery measures other than to inform the user of the existence of the error.
Possible Error Effect:	Since this check occurred after it was determined that an RSM number existed for an HSM, the audit will simply go on to the next RSM, if any, that is hanging off the HSM.
Corrective Action Taken:	None.
Manual Action Required:	If this error is rolling, the user should investigate the condition. Most likely the office information for the RLMODATT and RLCLIDAT relations need correction.
Dump Description:	
<i>Error Address =</i>	Address of the rCLIDAT buffer.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Cluster ID to which RSM belongs.
<i>Dump =</i>	rCLIDAT buffer.

Table 6 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's count status and that value calculated by the AUddsmcnt() function.
Possible Error Effect:	Incorrect SM count status could be accessed.
Corrective Action Taken:	The data delivery data associated with the SM's count status will be made consistent with that calculated in the AUddsmcnt() function.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery SM count status.
<i>Good Data =</i>	SM count from AUddsmcnt().
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 7 Error Code: DD_MODATT

Error Code:	DD_MODATT
Error Description:	The audit can't read the ODD relation RLMODATT with the given generic plan (GP) number.
Possible Error Effect:	The GP audit data checks may not be done for a particular GP.
Corrective Action Taken:	None.
Manual Action Required:	Validate the GP number that the checks are failing for. Validate the tuple for this GP in the ODD relation RLMODATT.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return from DBfrdtup.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The number of the GP that the checks can't be done for.
<i>Dump =</i>	The tuple from the ODD relation RLMODATT for the GP in question.

Table 8 Error Code: DDAMABIL

Error Code:	DDAMABIL
Error Description:	The AM's billing stream indicators maintained in AMsmmap on a per SM basis do not agree with those maintained in the ODD relation RLMODATT.
Possible Error Effect:	Billing data may be sent to the wrong billing stream.
Corrective Action Taken:	Make the AMsmmap billing stream indicators consistent with the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	AMsmmap data for this SM.
<i>Good Data =</i>	ODD rIMODATT data for this SM.
<i>Logical Index =</i>	The number of the SM whose billing stream indicator data is in error.
<i>Dump =</i>	Copy of the rIMODATT tuple of the SM in question.

Table 9 Error Code: DDAUEWRK

Error Code:	DDAUEWRK
Error Description:	The relationship between data delivery data associated with key DDAUEWRK and global variable AUEleact, is not correct.
Possible Error Effect:	The routine and elevated audits in OKP of the AM may never be scheduled to run.
Corrective Action Taken:	The data delivery data associated with key DDAUEWRK will be made consistent with global variable AUEleact.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The data delivery DDAUEWRK data.
<i>Good Data =</i>	The global variable AUEleact, maintained by audit control.
<i>Logical Index =</i>	The mode the audit was running in.
<i>Dump =</i>	NA.

Table 10 Error Code: DDCMVINT

Error Code:	DDCMVINT
Error Description:	The audit has detected an inconsistency between the SM's data delivery data associated with key DDCM_VINT and the ODD global variable Glcm_vint.
Possible Error Effect:	Speed of the XPC chips may be temporarily lost.
Corrective Action Taken:	Set the data delivery data to agree with that in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery DDCM_VINT data.
<i>Good Data =</i>	ODD Glcm_vint.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 11 Error Code: DDCNIOOS

Error Code:	DDCNIOOS
Error Description:	The audit has detected an inconsistency between data delivery data associated with the office CNI equipage state and the office CNI equipage state maintained in the ODD variable Glccs.
Possible Error Effect:	CNI unique software may be executing abnormally.
Corrective Action Taken:	The data delivery data associated with the office CNI equipage state will be made consistent with the office CNI equipage state maintained in the ODD variable Glccs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery office CNI equipage state.
<i>Good Data =</i>	ODD office CNI equipage state.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 12 Error Code: DDEQIMS

Error Code:	DDEQIMS
Error Description:	The audit has detected an inconsistency in data delivery data between the processor's static equipage status (rlMODATT.eqstat) and the processor's dynamic equipage status in the DDeqims[] array.
Possible Error Effect:	Calls could be incorrectly routed to the processor.
Corrective Action Taken:	The data delivery data associated with the processor's equipage status will be made consistent with the processors equipage status maintained in the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery processor equipage data (bit from DDeqims[] array).
<i>Good Data =</i>	ODD processor equipage data (rlMODATT.eqstat).
<i>Logical Index =</i>	The number of the processor whose equipage is in question (rlMODATT.module).
<i>Dump =</i>	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 13 Error Code: DDEQPROC

Error Code:	DDEQPROC
Error Description:	The audit has detected an inconsistency between data delivery data associated with a processors equipage status and the processors equipage status maintained in the ODD relation RLMODATT.
Possible Error Effect:	Calls could be incorrectly routed to the processor.
Corrective Action Taken:	The data delivery data associated with the processors equipage state will be made consistent with the processors equipage state maintained in the ODD relation RLMODATT.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery processor equipage data.
<i>Good Data</i> =	ODD processor equipage data.
<i>Logical Index</i> =	The number of the processor whose equipage is in question.
<i>Dump</i> =	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 14 Error Code: DDEQSTAT

Error Code:	DDEQSTAT
Error Description:	The audit has detected an inconsistency between data delivery data associated with a processor's equipage status and the processor's equipage status maintained in the ODD relation RLMODATT. With the introduction of the CMP, if the DDEQSTAT error rolls out most likely there is an inconsistency between the AM's version of RLMODATT and the CMP's version of RLMODATT.
Possible Error Effect:	Calls could be incorrectly routed to the SM.
Corrective Action Taken:	The data delivery data associated with the SM's equipage state will be made consistent with the SM equipage state maintained in the ODD relation RLMODATT.
Manual Action Required:	Manual action is required only if the error is rolling out for the same SM. The office database editor (ODBE) should be used to make the CMP and AM's versions of rIMODATT consistent. The tuple to be updated is keyed off of the processor number given in the logical index field of the Dump Description.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery processor equipage data.
<i>Good Data</i> =	ODD processor equipage state.
<i>Logical Index</i> =	The number of the processor whose equipage is in question.
<i>Dump</i> =	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 15 Error Code: DDHOSTNUM

Error Code:	DDHOSTNUM
Error Description:	The audit has detected an inconsistency between data delivery data associated with a module's host SM number and the module's host SM number maintained in the ODD relation RLMODATT.
Possible Error Effect:	Fault recovery actions for an RSM may fail.
Corrective Action Taken:	The data delivery data associated with the module's host SM number will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery host SM number data.
<i>Good Data</i> =	ODD host SM number data.
<i>Logical Index</i> =	The number of the SM whose host number is in question.
<i>Dump</i> =	Copy of the tuple for this SM from the ODD relation RLMODATT.

Table 16 Error Code: DDIMTYPE

Error Code:	DDIMTYPE
Error Description:	The audit has detected an inconsistency between data delivery data associated with an SM's type and the SM's type maintained in the ODD relation RLMODATT.
Possible Error Effect:	Various commands associated with the type of SM may not give accurate results.
Corrective Action Taken:	The data delivery data associated with the SM's type will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery SM type data.
<i>Good Data</i> =	ODD SM type data.

<i>Logical Index</i> =	The module number of the module whose type is in question.
<i>Dump</i> =	Copy of the tuple for this SM from the ODD relation RLMODATT.

Table 17 Error Code: DDINDEX

Error Code:	DDINDEX
Error Description:	The index into the data delivery audit table, AUdd1tbl[], is out of range.
Possible Error Effect:	The data delivery audit may not be able to audit all the data items it is supposed to, or an out of range address may be tried and executed.
Corrective Action Taken:	Reinitialize the index, and start from the beginning of the audit the next time it is entered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the local check table, AUdd1tbl[].
<i>Bad Data</i> =	The index that is out of range (AUddindx).
<i>Good Data</i> =	Maximum array index which when added to 1 equals the number of entries in the table (AUdd_flag).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 18 Error Code: DDMATECMP

Error Code:	DDMATECMP
Error Description:	The audit has detected an inconsistency between data delivery data associated with a CMP's mate number and the CMP's hostnum field maintained in the ODD relation RLMODATT.
Possible Error Effect:	Bad return value from reading for a CMP's mate.
Corrective Action Taken:	The data delivery data associated with the CMP's mate number will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery mate CMP number data.
<i>Good Data</i> =	ODD hostnum field.
<i>Logical Index</i> =	The number of the CMP whose mate number is in question.
<i>Dump</i> =	Copy of the tuple for this CMP from the ODD relation RLMODATT.

Table 19 Error Code: DDMGDSCB

Error Code:	DDMGDSCB
Error Description:	The audit has detected an inconsistency between data delivery data associated with the log, print, and discard status of a message class.
Possible Error Effect:	Messages of a particular message class may be discarded when they should not be.
Corrective Action Taken:	The data delivery data associated with the print status will be set to print the messages. At the same time, internally to data delivery, the discard status for the message class will be turned off so as not to discard the messages associated with the message class involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Data delivery log status for the message class. This is not an address, it's a data value.
<i>Bad Data</i> =	Data delivery discard status for the message class.
<i>Good Data</i> =	Data delivery print status for the message class. This is not necessarily good data, it's just some data that's used in determining consistency.
<i>Logical Index</i> =	Index into data delivery message class arrays.
<i>Dump</i> =	NA.

Table 20 Error Code: DD_MODATT

Error Code:	DD_MODATT
Error Description:	The audit can't read the ODD relation RLMODATT with the given SM number.
Possible Error Effect:	The AM per SM audit data checks may not be done for a particular SM.
Corrective Action Taken:	None.

Manual Action Required:	Validate the SM number that the checks are falling for. Validate the tuple for this SM in the ODD relation RLMODATT.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from DBfrdtup.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The number of the SM that the checks can't be done for.
<i>Dump</i> =	The tuple from the ODD relation RLMODATT for the SM in question.

Table 21 Error Code: DDSASWCK

Error Code:	DDSASWCK
Error Description:	The audit has detected an inconsistency between data delivery data associated with access keys DDSASWCK and DDRMSWCK.
Possible Error Effect:	The wrong inhibit status of automatic software checks may be displayed to the user.
Corrective Action Taken:	The data delivery data associated with access DDSASWCK will be made consistent with data delivery data associated with access key DDRMSWCK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery data DDSASWCK.
<i>Good Data</i> =	Data delivery data DDRMSWCK.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 22 Error Code: DDSINIT

Error Code:	DDSDINIT
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's initialization state and global variable INmode.
Possible Error Effect:	An incorrect state of initialization could be indicated to the user.
Corrective Action Taken:	The data delivery data associated with the SM's initialization state will be made consistent with INmode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery SM initialization state.
<i>Good Data</i> =	INmode, SM's initialization state.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 23 Error Code: DDSINIT

Error Code:	DDSDINIT
Error Description:	The audit has detected an inconsistency between data delivery data associated with the AM's initialization status and that data in Sldapdata table, maintained by AIM, that reflects the AM initialization progress.
Possible Error Effect:	Incorrect AM initialization status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the AM's initialization allow status will be made consistent with that maintained by AIM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery AM initialization status.
<i>Good Data</i> =	AIM AM initialization status.
<i>Logical Index</i> =	The number of the module whose status is in question.
<i>Dump</i> =	NA.

Table 24 Error Code: DDSMIN

Error Code:	DDSDMIN
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's minmode status and the Glthrt1 global variable used to throttle call

	originations.
Possible Error Effect:	An invalid minmode condition may be indicated to user.
Corrective Action Taken:	The data delivery data associated with the SM's minmode status will be made consistent with the G1thrt1 variable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery SM minmode status.
<i>Good Data =</i>	G1thrt1 global variable.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 25 Error Code: DDSMIN

Error Code:	DDSMIN
Error Description:	The relationship between data delivery data associated with AM minmode and CN1 minmode is not correct.
Possible Error Effect:	An invalid AM minmode condition could be in effect.
Corrective Action Taken:	The data delivery data associated with AM minmode and CN1 minmode are set to NOT AM minmode and NOT CN1 minmode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The data delivery AM minmode data.
<i>Good Data =</i>	The data delivery AM CN1 minmode data. (This should not be interpreted as the good data, but rather as the other data used to determine the inconsistency.)
<i>Logical Index =</i>	The mode the audit was running in.
<i>Dump =</i>	NA.

Table 26 Error Code: DDSREX

Error Code:	DDSREX
Error Description:	The audit has detected an inconsistency between data delivery data associated with a module's routine exercise inhibit/allow status and the inhibit/allow status maintained by REX.
Possible Error Effect:	Incorrect routine exercise inhibit/allow status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the module's inhibit/allow status will be made consistent with that maintained by REX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery REX inhibit/allow status.
<i>Good Data =</i>	REX inhibit/allow status (example, 00010203). Reading from left: 00 = Not used. 01 = REX status of fabric. 02 = REX status of ELS. 03 = REX status of DGN.
<i>Logical Index =</i>	The number of the module whose status is in question.
<i>Dump =</i>	NA.

Table 27 Error Code: DDSREX

Error Code:	DDSREX
Error Description:	The audit has detected an inconsistency between data delivery data associated with the CM's routine exercise inhibit/allow status and the inhibit/allow status maintained by REX.
Possible Error Effect:	Incorrect routine exercise inhibit/allow status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the CM's inhibit/allow status will be made consistent with that maintained by REX.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery REX inhibit/allow status.
<i>Good Data =</i>	REX inhibit/allow status.
<i>Logical Index =</i>	The number of the module whose status is in question.
<i>Dump =</i>	NA.

Table 28 Error Code: DDSVCDGRD

Error Code:	DDSVCDGRD
Error Description:	A non MMRS M RSM has been found whose DDSVCDGRD ("service degraded") key is set. This key refers to the status of an SM's DLIs. Since RSMs use RLIs, the key should always be clear.
Possible Error Effect:	Erroneous link status information could cause faulty call handling.
Corrective Action Taken:	The DD data is reset to the correct value (DDCLEAR).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value of the DDSVCDGRD key for this RSM.
<i>Good Data =</i>	The correct value for the key (DDCLEAR).
<i>Logical Index =</i>	The SM number.
<i>Dump =</i>	NA.

Table 29 Error Code: DEACT_DDASPTRIG

Error Code:	DEACT_DDASPTRIG
Error Description:	The data delivery data from the key DDASPTRIG is cleared to indicate ASP 0.1 activation. The static data in rIDS_APP finds a match using a key of CCAINR01 and the global parameter <i>Glpvn_ofc</i> is set to DBPVN_SSP, so data delivery data should indicate ASP is de-activated.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The data delivery data DDASPTRIG is set to indicate ASP 0.1 de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The correct value for the key (DDSET).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 30 Error Code: DEACT_DDPVNALW

Error Code:	DEACT_DDPVNALW
Error Description:	The value of the data delivery key, DDPVNALW, is clear to indicate ASP de-activation. A match in static data rIDS_APP exists for one of the following keys: (1) A key of CCPVN when <i>Glpvn_ofc</i> is set to DBPVN_SSP, or (2) A key of CCNS800. Also, the office type is a service switching point (SSP) or Glssp is set to DBYES. Therefore, DDPVNALW should be set.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The value of DDPVNALW is set to indicate ASP activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The correct value for the key (DDSET).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 31 Error Code: EQPISOLSEP

Error Code:	EQPISOLSEP
Error Description:	An RSM in a MMRSM site was found to be separated without isolation, in the master copies RISOL and RSEP respectively.
Possible Error Effect:	Reduction in the number of RSM available in the MMRSM site.
Corrective Action Taken:	The separation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSM in RSEP.
<i>Good Data =</i>	Correct separation state of the RSM to the value of RISOL.
<i>Logical Index =</i>	Module number.
<i>Dump =</i>	NA.

Table 32 Error Code: EQPRSMISOL

Error Code:	EQPRSMISOL
Error Description:	The RSMISOL bit for an equipped RSM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	IM number whose isolation state is incorrect.
<i>Dump =</i>	NA.

Table 33 Error Code: EQPRSMINIT

Error Code:	EQPRSMINIT
Error Description:	The initialization status of an RSM in a MMRSM site does not agree with the information kept in the master copy SINIT.
Possible Error Effect:	This causes loss of MMRSM resources.
Corrective Action Taken:	The initialization state of the RSM is corrected to agree with SINIT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the RSMs in the DDRSMINIT byte.
<i>Good Data =</i>	Value of SINIT which is written on to DDRSMINIT.
<i>Logical Index =</i>	MMRSM Module number.
<i>Dump =</i>	NA.

Table 34 Error Code: EQPRSMSEP

Error Code:	EQPRSMSEP
Error Description:	The separation status of an RSM in a MMRSM site does not agree with the information kept in the master copy RSEP.
Possible Error Effect:	This causes loss of MMRSM resources.
Corrective Action Taken:	The separation state of the RSM is corrected to agree with RSEP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSMs in the DDRSMSEP byte.
<i>Good Data =</i>	Value of RSEP which is written on the DDRSMSEP.
<i>Logical Index =</i>	MMRSM module number.
<i>Dump =</i>	NA.

Table 35 Error Code: HIMRSMINIT

Error Code:	HIMRSMINIT
Error Description:	The RSMINIT byte for non MMRSM IMs is not zero.
Possible Error Effect:	The initialization state of the module is incorrect.
Corrective Action Taken:	The initialization state of the module is corrected.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Initialization state of the modules in the DDRSMINIT byte.
<i>Good Data</i> =	Correct initialization states of the modules.
<i>Logical Index</i> =	Module number whose initialization state is incorrect.
<i>Dump</i> =	NA.

Table 36 Error Code: HIMRSMISOL

Error Code:	HIMRSMISOL
Error Description:	The RSMISOL bit for an IM other than HIM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data</i> =	Correct isolation states of RSMs.
<i>Logical Index</i> =	IM number whose isolation state is incorrect.
<i>Dump</i> =	NA.

Table 37 Error Code: HIMRSMSEP

Error Code:	HIMRSMSEP
Error Description:	The RSMSEP byte for a non MMRSIM IM is not zero.
Possible Error Effect:	The separation state of the module is incorrect.
Corrective Action Taken:	The separation state of the module is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Separation state of the modules in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the modules.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 38 Error Code: INV

Error Code:	INV
Error Description:	The DDRSEP byte for an HSM is not zero.
Possible Error Effect:	The separation state of the HSM is incorrect.
Corrective Action Taken:	The separation state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Separation state of the HSM in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the HSM.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 39 Error Code: LNK_CTS_NUM

Error Code:	LNK_CTS_NUM
Error Description:	The data in either DDCTS0 or DDCTS1 pertaining to the even and odd CTS link numbers is inconsistent with the data maintained in the ODD relation RLMODATT.
Possible Error Effect:	Incorrect link information will be used.
Corrective Action Taken:	The data in either DDCTS0 or DDCTS1 will be updated to be consistent with the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0 if error is pertaining to DDCTS0 or 1 if error is pertaining to DDCTS1.
<i>Bad Data</i> =	Return value from DDREAD call.
<i>Good Data</i> =	ODD rIMODATT data for this SM. Even CTS link number for DDCTS0 and ODD CTS link number for DDCTS1.
<i>Logical Index</i> =	The number of the SM whose link information is in error.
<i>Dump</i> =	Copy of the rIMODATT tuple of the SM in question.

Table 40 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	The DD key summary status does not match the actual value returned by TM.
Possible Error Effect:	A major or minor alarm may sound, when it should or not sound.
Corrective Action Taken:	The corresponding DD key will be updated with the appropriate value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Bad DD key value.
<i>Good Data =</i>	Good DD key value.
<i>Logical Index =</i>	The DD key that had the error.
<i>Dump =</i>	NA.

Table 41 Error Code: MODATT

Error Code:	MODATT
Error Description:	An attempt to obtain a rIMODATT tuple with a given key failed. The audit has not taken any recovery measures other than to inform the user of the existence of the error.
Possible Error Effect:	Since this check occurred after it was determined that an RSM number existed for an HSM, the audit will simply go on to the next RSM, if any, hanging off the HSM.
Corrective Action Taken:	None.
Manual Action Required:	If this error is rolling that user should investigate the condition.
Dump Description:	
<i>Error Address =</i>	Address of the rIMODATT tuple key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key to the rIMODATT tuple which is the RSM module number.
<i>Dump =</i>	NA.

Table 42 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Corresponding rIHSMRSM tuple could not be found in the database. This error will continue to print out until fixed by the user.
Possible Error Effect:	May cause asserts and/or interrupts by various application programs.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, insert the missing rIHSMRSM tuple using the key which is dumped in the Logical Index of this error report.
Dump Description:	
<i>Error Address =</i>	Address of rIHSMRSM tuple buffer.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (<i>hsmnum</i>) of the rIHSMRSM tuple buffer.
<i>Dump =</i>	NA.

Table 43 Error Code: RCMISOL

Error Code:	RCMISOL
Error Description:	The communication module vintage is not CM2 and the data delivery keys DDRCMISOL and DDSCMISOL are not cleared. These two keys can only be set in CM2 offices.
Possible Error Effect:	The DD keys will contain the incorrect status. As a result, communication module maintenance functionality may not be available.
Corrective Action Taken:	Write both the DDRCMISOL and DDSCMISOL keys to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The data delivery DDSCMISOL data.
<i>Bad Data =</i>	The data delivery DDRCMISOL data.
<i>Good Data =</i>	Glcmm0vint.
<i>Logical Index =</i>	Glcmm1vint.
<i>Dump =</i>	NA.

Table 44 Error Code: RSMINITO

Error Code:	RSMINITO
Error Description:	The RMSINIT byte for an RSM is not zero.
Possible Error Effect:	The initialization state of the RSM is incorrect.
Corrective Action Taken:	The initialization state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the RSM in the DDRSMINIT byte.
<i>Good Data =</i>	Correct initialization states of the RSM.
<i>Logical Index =</i>	Module number whose initialization state is incorrect.
<i>Dump =</i>	NA.

Table 45 Error Code: RSMINITO

Error Code:	RSMINITO
Error Description:	The RSMINIT byte for an HSM is not zero.
Possible Error Effect:	The initialization state of the HSM is incorrect.
Corrective Action Taken:	The initialization state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the HSM in the DDRSMINIT byte.
<i>Good Data =</i>	Correct initialization states of the HSM.
<i>Logical Index =</i>	Module number whose initialization state is incorrect.
<i>Dump =</i>	NA.

Table 46 Error Code: RSMSEPO

Error Code:	RSMSEPO
Error Description:	The RSMSEP byte for an RSM is not zero.
Possible Error Effect:	The separation state of the RSM is incorrect.
Corrective Action Taken:	The separation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSM in the DDRSMSEP byte.
<i>Good Data =</i>	Correct separation states of the RSM.
<i>Logical Index =</i>	Module number whose separation state is incorrect.
<i>Dump =</i>	NA.

Table 47 Error Code: RSMSEPO

Error Code:	RSMSEPO
Error Description:	The RSMSEP byte for an HSM is not zero.
Possible Error Effect:	The separation state of the HSM is incorrect.
Corrective Action Taken:	The separation state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the HSM in the DDRSMSEP byte.
<i>Good Data =</i>	Correct separation states of the HSM.
<i>Logical Index =</i>	Module number whose separation state is incorrect.
<i>Dump =</i>	NA.

Table 48 Error Code: SCMISOL

Error Code:	SCMISOL
Error Description:	The CM isolation DD status key DDSCMISOL is incorrectly set in the pending state (DDPCMISOL). A <i>Good Data</i> value of 1 indicates that the CM isolation DD request key DDRCMISOL is not normal and entry into manual CM isolation should be in progress. However, there is no CM isolation request on the SMARS queue.

	A <i>Good Data</i> value of 0 indicates that DDRCMISOL is normal and an AM/CM re-synchronization should be in progress. However, the SMrsmode re-synchronization flag is SMRSNULL.
Possible Error Effect:	The DDSCMISOL DD key is stuck in what should be a transient pending state giving incorrect CM isolation status and stubbing off PCD auditing. Some communication module maintenance functionality may not be available.
Corrective Action Taken:	Write the DDSCMISOL key to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value for DDRCMISOL.
<i>Bad Data</i> =	Value for DDSCMISOL.
<i>Good Data</i> =	Either 0 or 1, see Error Description.
<i>Logical Index</i> =	Value for SMrsmode.
<i>Dump</i> =	NA.

Table 49 Error Code: STUCK

Error Code:	STUCK
Error Description:	The RSMISOL byte for an RSM is not zero.
Possible Error Effect:	Invalid state of the RSM single RSM offices should always have the isolation bits set to zero.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Isolation byte DDRSMISOL.
<i>Good Data</i> =	Written value of zero to DMRSMISOL.
<i>Logical Index</i> =	RSM module number.
<i>Dump</i> =	NA.

Table 50 Error Code: TYPE

Error Code:	TYPE
Error Description:	A switching module (SM) type in data delivery (DD) key DDSM2000 is inconsistent with RLMODATT (rIMODATT.unitttype = SMSM2000 and DDSM2000 = DD CLEAR for the SM or rIMODATT.unitttype != SMSM2000 and DDSM2000 = DDSET for the SM).
Possible Error Effect:	Call processing may take incorrect actions for the given SM.
Corrective Action Taken:	The audit writes DDSM2000 to agree with rIMODATT.unitttype. That is, if rIMODATT.unitttype = SMSM2000, the audit writes the value of DDSM2000 to DDSET for this SM. Otherwise, the audit writes the value of DDSM2000 for this SM to DD CLEAR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Value in DDSM2000 for the SM.
<i>Good Data</i> =	SM type (rIMODATT.unitttype)
<i>Logical Index</i> =	SM number (rIMODATT.module)
<i>Dump</i> =	rIMODATT tuple for this SM.

Table 51 Error Code: UEQRSMISOL

Error Code:	UEQRSMISOL
Error Description:	The RSMISOL bit for an unequipped RSM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data</i> =	Correct isolation states of RSMs.
<i>Logical Index</i> =	Offset of the unequipped RSM.
<i>Dump</i> =	NA.

Table 52 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up, after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	Reinitialize audit global data, return audit complete, and wait for next cycle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	AUddindex (tells audit which SM to check next).

DD-B

Software Release: 5E12 - 5E13

1. DATA DELIVERY (DD) AUDIT

The data delivery (DD) mechanism and audit covers data in the operational kernel process (OKP) and switch maintenance kernel process (SMKP) environments in the administrative module (AM), communications module processor (CMP) as well as the operating system for distributed switching-switching modules (OSDS-SMs) environment. The audit verifies the consistency of different data delivery data against other static data, data delivery, and global data. The audit reports errors it detects and makes the slave data consistent with available master data. If no master data exists, the data delivery data is set to a safe state. Data delivery data in incorrect states could disable call processing and cause problems for maintenance, billing, and growth; hence the need for the DD audit.

The DD audit runs in the CMP, OKP, SM, and SMKP environments in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DD audit checks the consistency of a number of different DD data structures. The term "DD key" refers to an identifier for information maintained by data delivery. Some of these data structures exist in the AM as AM-only data, some exist in the AM on a per SM basis, some exist in the AM on a per message class basis, and some of the data exists only in the SMs. The DD data structures checked are referenced by the following DD keys:

DDAPDLDSL	DDDIOPMODEL	DDPCONFIGSTL	DDRSMSEP
DDASPTRIGL	DDEQSTATL	DDPSSWSTATL	DDRTCDELNK
DDAUEWRKL	DDHOSTNUML	DDPVNALWL	DDRTCENET
DDCM_VINTL	DDIMTYPEL	DDRISOLL	DDSASWC
DDCNIMINL	DDMATECMPL	DDRMSWCKL	DDSCMISOL
DDCNIOOSL	DDMGDSCBL	DDRSEPL	DDSINIT
DDCTS0L	DDMGLOGBL	DDRSMINIL	DDSMIN
DDCTS1L	DDMGPRIBL	DDRSMISOLL	DDSREX
DDDASCDL			

3. AUDIT CHECKS

The audit makes the following checks and takes the specified actions.

Check	Environment	Action
DDAPDLDSL	SM	Checks the new data delivery key DDAPDLDSL against the AP link status table (PSaplst[]). DDAPDLDSL provides a summary of the status of the AP data link digital subscribe lines (DSLs) for a given SM. PSaplst[] contains information on the status of each of the individual AP data link DSLs. If the two values found do not match, an audit error is reported and DDAPDLDSL will be updated to agree with the information found in PSaplst[].
DDASPTRIG	OKP	Checks DD data associated with key DDASPTRIG against the data against the rIDS_APP tuple referenced by the key of CCAINR01. When the rIDS_APP tuple is not found, the DD data should indicate clear. When the rIDS_APP tuple is found and the global parameter is DBPVN_SSP, the DD data should be set.
DDAUEWRK	SMKP	Checks DD data associated with key DDAUEWRK against a global variable, AUEleact, which is maintained by audit control. When the global variable is set, the DD data should be set and when the global variable is clear, the DD data should indicate clear. If an inconsistency is detected, the DD data will be updated to be consistent with the global variable.
DDCM_VINT	OSDS-M	Checks DD data associated with key DDCM_VINT against the SM's office dependent data (ODD) global variable Glcm_vint. If an inconsistency is detected, the DD data will be updated to be consistent with the global variable. An audit error concerning this data is expected during the conversion of a communication module, model 1 (CM1) office to a CM2 office.

DDCNIMIN	OKP	Checks DD data associated with key DDCNIMIN against the DD data associated with key DDSMIN. If DDSMIN is set to indicate minmode, then DDCNIMIN should be set to indicate common network interface (CNI) minmode. If there is an inconsistency, both DD data will be set to not minmode.
DDCNIOOS	OKP	Checks DD data associated with key DDCNIOOS against the ODD variable Glccs. If an office is unequipped with CNI, the DD data should be set indicating that CNI is disabled. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDCTS0	OKP	Checks DD data associated with key DDCTS0 against the even CTS link number stored in MODATT. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDCTS1	OKP	Checks DD data associated with key DDCTS1 against the odd CTS link number stored in MODATT. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDDIOPMODE	OKP	Checks DD data associated with key DDDIOPMODE. If the DD data is set to indicate disk independent mode, a message will be sent to application integrity monitor (AIM) to verify a valid disk independent mode. This will be done by AIM checking the essential disks statuses in the equipment configuration database (ECD). If an inconsistency exists, AIM will reset the DD data to be consistent with the ECD.
DDEQPROCS	OKP/CMP	Checks DD data associated with the key DDEQPROCS against the processors equipage status maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data is made consistent with the ODD.
DDEQSTAT	OKP/CMP	Checks DD data associated with key DDEQSTAT against the SM equipage status maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD. (The DD data here is different than that associated with key DDEQIMS.)
DDHOSTNUM	OKP/CMP	Checks DD data associated with key DDHOSTNUM against the host module number maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDIMTYPE	OKP	Checks DD data associated with key DDIMTYPE against the SM type maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDMATECMP	SMKP/CMP	Checks DD data associated with key DDMATECMP against the hostnum field maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDMGDSCB	OKP	Checks DD data associated with key DDMGDSCB against DD data associated with keys DDMGLOGB and DDMGPRTB. If any of the DD data indicates to log or print messages, then the DD data for discard should not be set. If an inconsistency exists, the discard data will be set to not discard.
DDMGLOGB	OKP	This represents the DD data for logging messages. See the check for DDMGDSCB.
DDMGPRTB	OKP	This represents the DD data for printing messages. See the check for DDMGDSCB.
DDPCONFIGST	SMKP	Checks the key DDPCONFIGST for soft switch status by going off of the PCD state for the unit. In the case of a program update, the soft switch periodically resets the key so that the user knows that the program update uses the key.
DDPSSWSTAT	SMKP	Checks DD data associated with key DDPSSWSTAT against the soft switch status. If an inconsistency exists between the soft switch status and the PCD, the DD audit will update the DD key with the PCD information.
DDPVNALW	OKP	Checks DD data associated with key DDPVNALW against the existence of an rIDS_APP tuple referenced by the key of CCPVN (when Glpvn_ofc is set to DBPVN_SSP) or CCNS800. When the rIDS_APP tuple is not found, the DD data should indicate clear. When the rIDS_APP tuple is found and Glpvn_ofc is DBPVN_SSP or Glssp is DBYES, the DD data should be set.
DDRISOL	OSDS-M	Checks are done in a remote switching module (RSM) to verify that the separation information (DDRSEP) about an RSM in a multimodule site is consistent with the corresponding master data kept in DDRISOL.
DDRMSWCK	OSDS-M	Checks DD data associated with key DDRMSWCK against DDSASWCK data. If DDSASWCK data is off-normal, DDRMSWCK data should be off-normal.

DDRSEP	OSDS-M	The data in DDRSEP is the master copy representing the separation status of all the RSMs in a multimodule site it is arranged on a per module basis. The value in DDRSMSEP is compared against this data, and used in correcting any inconsistencies.
DDRSMINIT	OSDS-M	Checks are done to verify that the module initialization information (DDRSMINIT) about a module in a multimodule site is consistent with the corresponding master data kept in DDSINIT. When an inconsistency occurs, the DDRSMINIT status for a given module is set to the value of DDSINIT.
DDRSMISOL	OSDS-M	Checks are done to verify that the isolation status (DDRSMISOL) about a module in a multimodule site is consistent with the master status kept in DDRISOL. When an inconsistency occurs, the DDRSMISOL status for a given module is set to the value of DDRISOL.
DDRSMSSEP	OSDS-M	The data in this 10-bit word is arranged based on the number RSMs in a multimodule site. The separation status of each RSM is maintained here. Checks are done to verify that the separation status (DDRSMSSEP) is consistent with the master status kept in DDRSEP. When an error is found, the wrong bit in DDRSMSEP is set to the value of DDRSEP.
DDRTCENET, DDRTCDELNK	OKP	Checks DD data associated with the key DDRTCENET against the data delivery key DDRTCDELNK sent by the global billing SM. If the two values found do not match, an audit error is reported and DDRTCENET will be updated to agree with DDRTCDELNK.
DDSASWCK	OKP	Checks DD data associated with key DDSASWCK against DDRMSWCK data. If DDRMSWCK data is off-normal, DDSASWCK data should be off-normal.
DDSINIT	OSDS-M	Checks DD data associated with key DDSINIT against the global variable INmode. If an inconsistency exists, the DD data will be made consistent with the global variable.
DDSMCOUNT	OKP	Checks the new data delivery key DDSMCOUNT against the actual number of SMS (including special growth). If the two values do not match, an audit error is reported, and the DDSMCOUNT is updated with the correct value.
DDSMIN	SM	Checks SM DD data associated with DDSMIN against the global variable G1thrt1. If the DD data indicates minmode and the global variable does not, the DD data will be set to not minmode.
DDSREX	OKP	Checks DD data associated with key DDSREX against the inhibit allow status of routine exercises maintained in a global routine exercise table (DGlddb[module].rxinh1st. enable1). If an inconsistency exists, the DD data will be made consistent with the global table.
DDSCMISOL	SMKP	Checks DD data associated with the key DDSCMISOL against the status of the CM isolation request DD key DDRCMISOL and either the SMARS job queue or the SMrsmode AM/CM re-synchronization flag. If an inconsistency exists, the DD data is written to normal.

4. ERROR HANDLING

If the ASP service features find errors, the data delivery bit is updated to be consistent with the static relation RLDS_APP.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_DDASPTRIG

Error Code:	ACT_DDASPTRIG
Error Description:	The data delivery data from the key DDASPTRIG is set to indicate ASP 0.1 activation. The static data in rlds_APP does not find a match using a key of CCAINR01, so data delivery data should indicate ASP 0.1 is de-activated.

Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The data delivery data DDASPTRIG is set to indicate ASP 0.1 de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The database return from reading the RLDS_APP relation.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: ACT_DDPVNALW

Error Code:	ACT_DDPVNALW
Error Description:	The value of the data delivery key, DDPVNALW, is set to indicate ASP activation. A match in static data rIDS_APP does not exist for either of the following keys: (1) A key of CCPVN when Glpvn_ofc is set to DBPVN_SSP, and (2) A key of CCNS800. Therefore, DDPVNALW should not be set.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The value of DDPVNALW is cleared to indicate ASP de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The database return from reading the RLDS_APP relation.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: BAD_DDTBL

Error Code:	BAD_DDTBL
Error Description:	An invalid state field (not AUDDDO, AUDDSKIP or AUDDDONE) in the data delivery audit table (AUdd1tbl[]), has been detected.
Possible Error Effect:	The data delivery audit may not be able to audit all the data items it is supposed to.
Corrective Action Taken:	Reinitialize the index (AUddindx), and start from the beginning of the audit the next time it is entered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local check table, AUdd1tbl[].
<i>Bad Data =</i>	The state element for the indicated table entry (AUdd1tbl[AUddindx].state).
<i>Good Data =</i>	Maximum array index which when added to 1 equals the number of entries in the table (AUdd_flag).
<i>Logical Index =</i>	Index into the table (AUddindx).
<i>Dump =</i>	The data delivery audit table (AUdd1tbl[]) of entries for the indicated processor.

Table 4 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	<i>In OKP or CMP:</i> The bit in DDeqmhppe[] associated with a particular switching module (SM) is set incorrectly. That is, either the bit indicates message handler (MH) connectivity for that SM when it should not (the bit is set) or the bit indicates no MH connectivity for that SM when the SM should have it (the bit is clear). See the dump description to determine which of the following cases applies [note that the "rIMODATT tuple" is the tuple associated with the particular SM (rIMODATT.module)]:

	- The bit is set but no rMODATT tuple exists for the SM. - The bit is set but at least one of the rMODATT <i>ql_ntwk0</i> or <i>ql_ntwk1</i> fields has value DBNO. - The bit is clear but both of the rMODATT <i>ql_ntwk0</i> and <i>ql_ntwk1</i> fields have value DBYES. <i>In an SM-2000:</i> The SM's copy of DDeqmhpipe[] is inconsistent with the <i>qends[]</i> fields in the rIQENDPTS tuples where rIQENDPTS.qservice = QLOSDS (representing MH connectivity) and rIQENDPTS.qserv_num = 0 (the default value, used for non-CCS connectivity). Specifically, for each bit in DDeqmhpipe[], that bit should be set if, and only if, the corresponding bits are set in rIQENDPTS.qends[] for both networks (rIQENDPTS.qnetwork = QLNET0 and QLNET1).
Possible Error Effect:	If the bit is set when it should be clear, quad linked packet switching (QLPS) network traffic may be routed to an SM unable to accept it. If the bit is clear when it should be set, the QLPS network may be degraded because it is unable to route to the associated SM.
Corrective Action Taken:	<i>In OKP or CMP:</i> The audit triggers an update of the bit in DDeqmhpipe[] and any other DD data associated with the SM. <i>In SM-2000:</i> The audit triggers a reinitialization of DDeqmhpipe[].
Manual Action Required:	None.
Dump Description:	
<i>In OKP or CMP:</i>	
<i>Error Address =</i>	SM equipage status (rMODATT.eqstat). A value of SMUNEQ may indicate that no rMODATT tuple exists for this SM.
<i>Bad Data =</i>	rMODATT.ql_ntwk0.
<i>Good Data =</i>	rMODATT.ql_ntwk1.
<i>Logical Index =</i>	The associated SM number (used in rMODATT.module when trying to read the rMODATT tuple).
<i>Dump =</i>	NA.
<i>In an SM-2000:</i>	
<i>Error Address =</i>	Current entry in rIQENDPTS.qends[] for network QLNET0.
<i>Bad Data =</i>	Current entry in DDeqmhpipe[] being checked.
<i>Good Data =</i>	Current entry in rIQENDPTS.qends[] for network QLNET1.
<i>Logical Index =</i>	Index of the entry in DDeqmhpipe[] being checked. That is, the index of the first entry to have an error.
<i>Dump 1 =</i>	DDeqmhpipe[].
<i>Dump 2 =</i>	The rIQENDPTS tuple associated with network QLNET0.

Table 5 Error Code: CLIDAT

Error Code:	CLIDAT
Error Description:	An attempt failed to obtain a rCLIDAT tuple with a given cluster ID. The audit has not taken any recovery measures other than to inform the user of the existence of the error.
Possible Error Effect:	Since this check occurred after it was determined that an RSM number existed for an HSM, the audit will simply go on to the next RSM, if any, that is hanging off the HSM.
Corrective Action Taken:	None.
Manual Action Required:	If this error is rolling, the user should investigate the condition. Most likely the

	office information for the RLMODATT and RLCLIDAT relations need correction.
Dump Description:	
<i>Error Address</i> =	Address of the rCLIDAT buffer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Cluster ID to which RSM belongs.
<i>Dump</i> =	rCLIDAT buffer.

Table 6 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's count status and that value calculated by the AUddsmcnt() function.
Possible Error Effect:	Incorrect SM count status could be accessed.
Corrective Action Taken:	The data delivery data associated with the SM's count status will be made consistent with that calculated in the AUddsmcnt() function.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery SM count status.
<i>Good Data</i> =	SM count from AUddsmcnt().
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 7 Error Code: DD_MODATT

Error Code:	DD_MODATT
Error Description:	The audit can't read the ODD relation RLMODATT with the given generic plan (GP) number.
Possible Error Effect:	The GP audit data checks may not be done for a particular GP.
Corrective Action Taken:	None.
Manual Action Required:	Validate the GP number that the checks are failing for. Validate the tuple for this GP in the ODD relation RLMODATT.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from DBfrdtup.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The number of the GP that the checks can't be done for.
<i>Dump</i> =	The tuple from the ODD relation RLMODATT for the GP in question.

Table 8 Error Code: DDAMABIL

Error Code:	DDAMABIL
Error Description:	The AM's billing stream indicators maintained in AMsmmap on a per SM basis do not agree with those maintained in the ODD relation RLMODATT.
Possible Error Effect:	Billing data may be sent to the wrong billing stream.
Corrective Action Taken:	Make the AMsmmap billing stream indicators consistent with the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	AMsmmap data for this SM.
<i>Good Data</i> =	ODD rIMODATT data for this SM.
<i>Logical Index</i> =	The number of the SM whose billing stream indicator data is in error.
<i>Dump</i> =	Copy of the rIMODATT tuple of the SM in question.

Table 9 Error Code: DDAUEWRK

Error Code:	DDAUEWRK
Error Description:	The relationship between data delivery data associated with key DDAUEWRK and global variable AUEleact, is not correct.
Possible Error Effect:	The routine and elevated audits in OKP of the AM may never be scheduled to run.
Corrective Action Taken:	The data delivery data associated with key DDAUEWRK will be made consistent

	with global variable AUEleact.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The data delivery DDAUEWRK data.
<i>Good Data</i> =	The global variable AUEleact, maintained by audit control.
<i>Logical Index</i> =	The mode the audit was running in.
<i>Dump</i> =	NA.

Table 10 Error Code: DDCMVINT

Error Code:	DDCMVINT
Error Description:	The audit has detected an inconsistency between the SM's data delivery data associated with key DDCM_VINT and the ODD global variable Glcm_vint.
Possible Error Effect:	Speed of the XPC chips may be temporarily lost.
Corrective Action Taken:	Set the data delivery data to agree with that in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery DDCM_VINT data.
<i>Good Data</i> =	ODD Glcm_vint.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 11 Error Code: DDCNIOOS

Error Code:	DDCNIOOS
Error Description:	The audit has detected an inconsistency between data delivery data associated with the office CNI equipage state and the office CNI equipage state maintained in the ODD variable Glccs.
Possible Error Effect:	CNI unique software may be executing abnormally.
Corrective Action Taken:	The data delivery data associated with the office CNI equipage state will be made consistent with the office CNI equipage state maintained in the ODD variable Glccs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery office CNI equipage state.
<i>Good Data</i> =	ODD office CNI equipage state.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 12 Error Code: DDEQIMS

Error Code:	DDEQIMS
Error Description:	The audit has detected an inconsistency in data delivery data between the processor's static equipage status (rIMODATT.eqstat) and the processor's dynamic equipage status in the DDeqims[] array.
Possible Error Effect:	Calls could be incorrectly routed to the processor.
Corrective Action Taken:	The data delivery data associated with the processor's equipage status will be made consistent with the processors equipage status maintained in the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery processor equipage data (bit from DDeqims[] array).
<i>Good Data</i> =	ODD processor equipage data (rIMODATT.eqstat).
<i>Logical Index</i> =	The number of the processor whose equipage is in question (rIMODATT.module).
<i>Dump</i> =	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 13 Error Code: DDEQPROC

Error Code:	DDEQPROC
Error Description:	The audit has detected an inconsistency between data delivery data associated

	with a processors equipage status and the processors equipage status maintained in the ODD relation RLMODATT.
Possible Error Effect:	Calls could be incorrectly routed to the processor.
Corrective Action Taken:	The data delivery data associated with the processors equipage state will be made consistent with the processors equipage state maintained in the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery processor equipage data.
<i>Good Data =</i>	ODD processor equipage data.
<i>Logical Index =</i>	The number of the processor whose equipage is in question.
<i>Dump =</i>	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 14 Error Code: DDEQSTAT

Error Code:	DDEQSTAT
Error Description:	The audit has detected an inconsistency between data delivery data associated with a processor's equipage status and the processor's equipage status maintained in the ODD relation RLMODATT. With the introduction of the CMP, if the DDEQSTAT error rolls out most likely there is an inconsistency between the AM's version of RLMODATT and the CMP's version of RLMODATT.
Possible Error Effect:	Calls could be incorrectly routed to the SM.
Corrective Action Taken:	The data delivery data associated with the SM's equipage state will be made consistent with the SM equipage state maintained in the ODD relation RLMODATT.
Manual Action Required:	Manual action is required only if the error is rolling out for the same SM. The office database editor (ODBE) should be used to make the CMP and AM's versions of RLMODATT consistent. The tuple to be updated is keyed off of the processor number given in the logical index field of the Dump Description.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery processor equipage data.
<i>Good Data =</i>	ODD processor equipage state.
<i>Logical Index =</i>	The number of the processor whose equipage is in question.
<i>Dump =</i>	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 15 Error Code: DDHOSTNUM

Error Code:	DDHOSTNUM
Error Description:	The audit has detected an inconsistency between data delivery data associated with a module's host SM number and the module's host SM number maintained in the ODD relation RLMODATT.
Possible Error Effect:	Fault recovery actions for an RSM may fail.
Corrective Action Taken:	The data delivery data associated with the module's host SM number will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery host SM number data.
<i>Good Data =</i>	ODD host SM number data.
<i>Logical Index =</i>	The number of the SM whose host number is in question.
<i>Dump =</i>	Copy of the tuple for this SM from the ODD relation RLMODATT.

Table 16 Error Code: DDIMTYPE

Error Code:	DDIMTYPE
Error Description:	The audit has detected an inconsistency between data delivery data associated with an SM's type and the SM's type maintained in the ODD relation RLMODATT.
Possible Error Effect:	Various commands associated with the type of SM may not give accurate results.

Corrective Action Taken:	The data delivery data associated with the SM's type will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery SM type data.
<i>Good Data</i> =	ODD SM type data.
<i>Logical Index</i> =	The module number of the module whose type is in question.
<i>Dump</i> =	Copy of the tuple for this SM from the ODD relation RLMODATT.

Table 17 Error Code: DDINDEX

Error Code:	DDINDEX
Error Description:	The index into the data delivery audit table, AUdd1tbl[], is out of range.
Possible Error Effect:	The data delivery audit may not be able to audit all the data items it is supposed to, or an out of range address may be tried and executed.
Corrective Action Taken:	Reinitialize the index, and start from the beginning of the audit the next time it is entered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the local check table, AUdd1tbl[].
<i>Bad Data</i> =	The index that is out of range (AUddindx).
<i>Good Data</i> =	Maximum array index which when added to 1 equals the number of entries in the table (AUdd flag).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 18 Error Code: DDMATECMP

Error Code:	DDMATECMP
Error Description:	The audit has detected an inconsistency between data delivery data associated with a CMP's mate number and the CMP's hostnum field maintained in the ODD relation RLMODATT.
Possible Error Effect:	Bad return value from reading for a CMP's mate.
Corrective Action Taken:	The data delivery data associated with the CMP's mate number will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery mate CMP number data.
<i>Good Data</i> =	ODD hostnum field.
<i>Logical Index</i> =	The number of the CMP whose mate number is in question.
<i>Dump</i> =	Copy of the tuple for this CMP from the ODD relation RLMODATT.

Table 19 Error Code: DDMGDSCB

Error Code:	DDMGDSCB
Error Description:	The audit has detected an inconsistency between data delivery data associated with the log, print, and discard status of a message class.
Possible Error Effect:	Messages of a particular message class may be discarded when they should not be.
Corrective Action Taken:	The data delivery data associated with the print status will be set to print the messages. At the same time, internally to data delivery, the discard status for the message class will be turned off so as not to discard the messages associated with the message class involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Data delivery log status for the message class. This is not an address, it's a data value.
<i>Bad Data</i> =	Data delivery discard status for the message class.
<i>Good Data</i> =	Data delivery print status for the message class. This is not necessarily good data, it's just some data that's used in determining consistency.
<i>Logical Index</i> =	Index into data delivery message class arrays.
<i>Dump</i> =	NA.

Table 20 Error Code: DD_MODATT

Error Code:	DD_MODATT
Error Description:	The audit can't read the ODD relation RLMODATT with the given SM number.
Possible Error Effect:	The AM per SM audit data checks may not be done for a particular SM.
Corrective Action Taken:	None.
Manual Action Required:	Validate the SM number that the checks are falling for. Validate the tuple for this SM in the ODD relation RLMODATT.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return from DBfrdtup.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The number of the SM that the checks can't be done for.
<i>Dump =</i>	The tuple from the ODD relation RLMODATT for the SM in question.

Table 21 Error Code: DDRTCD

Error Code:	DDRTCD
Error Description:	The DDRTCDENET bit in the AM does not match DDRTCDELNK bit sent from the Global Billing SM.
Possible Error Effect:	Status information that controls whether real time billing data should be sent to the global billing SM is incorrect. Billing data may be inappropriately buffered (or not buffered) in an originating SM's real time billing memory (RTBM).
Corrective Action Taken:	The DDRTCDENET is set equal to DDRTCDELNK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DDRTCDENET.
<i>Good Data =</i>	DDRTCDELNK.
<i>Logical Index =</i>	Global billing SM number.
<i>Dump =</i>	NA.

Table 22 Error Code: DDSASWCK

Error Code:	DDSASWCK
Error Description:	The audit has detected an inconsistency between data delivery data associated with access keys DDSASWCK and DDRMSWCK.
Possible Error Effect:	The wrong inhibit status of automatic software checks may be displayed to the user.
Corrective Action Taken:	The data delivery data associated with access DDSASWCK will be made consistent with data delivery data associated with access key DDRMSWCK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery data DDSASWCK.
<i>Good Data =</i>	Data delivery data DDRMSWCK.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 23 Error Code: DDSINIT

Error Code:	DDSinIT
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's initialization state and global variable INmode.
Possible Error Effect:	An incorrect state of initialization could be indicated to the user.
Corrective Action Taken:	The data delivery data associated with the SM's initialization state will be made consistent with INmode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery SM initialization state.
<i>Good Data =</i>	INmode, SM's initialization state.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 24 Error Code: DDSINIT

Error Code:	DDSINIT
Error Description:	The audit has detected an inconsistency between data delivery data associated with the AM's initialization status and that data in Sldapdata table, maintained by AIM, that reflects the AM initialization progress.
Possible Error Effect:	Incorrect AM initialization status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the AM's initialization allow status will be made consistent with that maintained by AIM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery AM initialization status.
<i>Good Data =</i>	AIM AM initialization status.
<i>Logical Index =</i>	The number of the module whose status is in question.
<i>Dump =</i>	NA.

Table 25 Error Code: DDSMIN

Error Code:	DDSMIN
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's minmode status and the G1thrt1 global variable used to throttle call originations.
Possible Error Effect:	An invalid minmode condition may be indicated to user.
Corrective Action Taken:	The data delivery data associated with the SM's minmode status will be made consistent with the G1thrt1 variable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery SM minmode status.
<i>Good Data =</i>	G1thrt1 global variable.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 26 Error Code: DDSMIN

Error Code:	DDSMIN
Error Description:	The relationship between data delivery data associated with AM minmode and CNI minmode is not correct.
Possible Error Effect:	An invalid AM minmode condition could be in effect.
Corrective Action Taken:	The data delivery data associated with AM minmode and CNI minmode are set to NOT AM minmode and NOT CNI minmode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The data delivery AM minmode data.
<i>Good Data =</i>	The data delivery AM CNI minmode data. (This should not be interpreted as the good data, but rather as the other data used to determine the inconsistency.)
<i>Logical Index =</i>	The mode the audit was running in.
<i>Dump =</i>	NA.

Table 27 Error Code: DDSREX

Error Code:	DDSREX
Error Description:	The audit has detected an inconsistency between data delivery data associated with a module's routine exercise inhibit/allow status and the inhibit/allow status maintained by REX.
Possible Error Effect:	Incorrect routine exercise inhibit/allow status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the module's inhibit/allow status will be made consistent with that maintained by REX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery REX inhibit/allow status.

<i>Good Data</i> =	REX inhibit/allow status (example, 00010203). Reading from left: 00 = Not used. 01 = REX status of fabric. 02 = REX status of ELS. 03 = REX status of DGN.
<i>Logical Index</i> =	The number of the module whose status is in question.
<i>Dump</i> =	NA.

Table 28 Error Code: DDSREX

Error Code:	DDSREX
Error Description:	The audit has detected an inconsistency between data delivery data associated with the CM's routine exercise inhibit/allow status and the inhibit/allow status maintained by REX.
Possible Error Effect:	Incorrect routine exercise inhibit/allow status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the CM's inhibit/allow status will be made consistent with that maintained by REX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery REX inhibit/allow status.
<i>Good Data</i> =	REX inhibit/allow status.
<i>Logical Index</i> =	The number of the module whose status is in question.
<i>Dump</i> =	NA.

Table 29 Error Code: DDSVCDGRD

Error Code:	DDSVCDGRD
Error Description:	A non MMRS RSM has been found whose DDSVCDGRD ("service degraded") key is set. This key refers to the status of an SM's DLIs. Since RSMs use RLIs, the key should always be clear.
Possible Error Effect:	Erroneous link status information could cause faulty call handling.
Corrective Action Taken:	The DD data is reset to the correct value (DDCLEAR).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of the DDSVCDGRD key for this RSM.
<i>Good Data</i> =	The correct value for the key (DDCLEAR).
<i>Logical Index</i> =	The SM number.
<i>Dump</i> =	NA.

Table 30 Error Code: DEACT_DDASPTRIG

Error Code:	DEACT_DDASPTRIG
Error Description:	The data delivery data from the key DDASPTRIG is cleared to indicate ASP 0.1 activation. The static data in rIDS_APP finds a match using a key of CCAINR01 and the global parameter <i>Glpvn_ofc</i> is set to DBPVN_SSP, so data delivery data should indicate ASP is de-activated.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The data delivery data DDASPTRIG is set to indicate ASP 0.1 de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return code from DD read.
<i>Good Data</i> =	The correct value for the key (DDSET).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 31 Error Code: DEACT_DDPVNALW

Error Code:	DEACT_DDPVNALW
Error Description:	The value of the data delivery key, DDPVNALW, is clear to indicate ASP de-activation. A match in static data rIDS_APP exists for one of the following keys: (1) A key of CCPVN when Glpvn_ofc is set to DBPVN_SSP, or (2) A key of CCNS800. Also, the office type is a service switching point (SSP) or Glssp is set to DBYES. Therefore, DDPVNALW should be set.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The value of DDPVNALW is set to indicate ASP activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The correct value for the key (DDSET).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 32 Error Code: EQPISOLSEP

Error Code:	EQPISOLSEP
Error Description:	An RSM in a MMRS site was found to be separated without isolation, in the master copies RISOL and RSEP respectively.
Possible Error Effect:	Reduction in the number of RSM available in the MMRS site.
Corrective Action Taken:	The separation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSM in RSEP.
<i>Good Data =</i>	Correct separation state of the RSM to the value of RISOL.
<i>Logical Index =</i>	Module number.
<i>Dump =</i>	NA.

Table 33 Error Code: EQPRSMISOL

Error Code:	EQPRSMISOL
Error Description:	The RSMISOL bit for an equipped RSM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	IM number whose isolation state is incorrect.
<i>Dump =</i>	NA.

Table 34 Error Code: EQPRSMINIT

Error Code:	EQPRSMINIT
Error Description:	The initialization status of an RSM in a MMRS site does not agree with the information kept in the master copy SINIT.
Possible Error Effect:	This causes loss of MMRS resources.
Corrective Action Taken:	The initialization state of the RSM is corrected to agree with SINIT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the RSMs in the DDRSMINIT byte.
<i>Good Data =</i>	Value of SINIT which is written on to DDRSMINIT.
<i>Logical Index =</i>	MMRS module number.
<i>Dump =</i>	NA.

Table 35 Error Code: EQPRSMSEP

Error Code:	EQPRSMSEP
Error Description:	The separation status of an RSM in a MMRSM site does not agree with the information kept in the master copy RSEP.
Possible Error Effect:	This causes loss of MMRSM resources.
Corrective Action Taken:	The separation state of the RSM is corrected to agree with RSEP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSMs in the DDRSMSEP byte.
<i>Good Data =</i>	Value of RSEP which is written on the DDRSMSEP.
<i>Logical Index =</i>	MMRSM module number.
<i>Dump =</i>	NA.

Table 36 Error Code: HIMRSMINIT

Error Code:	HIMRSMINIT
Error Description:	The RSMINIT byte for non MMRSM IMs is not zero.
Possible Error Effect:	The initialization state of the module is incorrect.
Corrective Action Taken:	The initialization state of the module is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the modules in the DDRSMINIT byte.
<i>Good Data =</i>	Correct initialization states of the modules.
<i>Logical Index =</i>	Module number whose initialization state is incorrect.
<i>Dump =</i>	NA.

Table 37 Error Code: HIMRSMISOL

Error Code:	HIMRSMISOL
Error Description:	The RSMISOL bit for an IM other than HIM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	IM number whose isolation state is incorrect.
<i>Dump =</i>	NA.

Table 38 Error Code: HIMRSMSEP

Error Code:	HIMRSMSEP
Error Description:	The RSMSEP byte for a non MMRSM IM is not zero.
Possible Error Effect:	The separation state of the module is incorrect.
Corrective Action Taken:	The separation state of the module is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the modules in the DDRSMSEP byte.
<i>Good Data =</i>	Correct separation states of the modules.
<i>Logical Index =</i>	Module number whose separation state is incorrect.
<i>Dump =</i>	NA.

Table 39 Error Code: INV

Error Code:	INV
Error Description:	The DDRSEP byte for an HSM is not zero.
Possible Error Effect:	The separation state of the HSM is incorrect.
Corrective Action Taken:	The separation state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	Separation state of the HSM in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the HSM.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 40 Error Code: LNK_CTS_NUM

Error Code:	LNK_CTS_NUM
Error Description:	The data in either DDCTS0 or DDCTS1 pertaining to the even and odd CTS link numbers is inconsistent with the data maintained in the ODD relation RLMODATT.
Possible Error Effect:	Incorrect link information will be used.
Corrective Action Taken:	The data in either DDCTS0 or DDCTS1 will be updated to be consistent with the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0 if error is pertaining to DDCTS0 or 1 if error is pertaining to DDCTS1.
<i>Bad Data</i> =	Return value from DDREAD call.
<i>Good Data</i> =	ODD rIMODATT data for this SM. Even CTS link number for DDCTS0 and ODD CTS link number for DDCTS1.
<i>Logical Index</i> =	The number of the SM whose link information is in error.
<i>Dump</i> =	Copy of the rIMODATT tuple of the SM in question.

Table 41 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	The DD key summary status does not match the actual value returned by TM.
Possible Error Effect:	A major or minor alarm may sound, when it should or not sound.
Corrective Action Taken:	The corresponding DD key will be updated with the appropriate value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Bad DD key value.
<i>Good Data</i> =	Good DD key value.
<i>Logical Index</i> =	The DD key that had the error.
<i>Dump</i> =	NA.

Table 42 Error Code: MODATT

Error Code:	MODATT
Error Description:	An attempt to obtain a rIMODATT tuple with a given key failed. The audit has not taken any recovery measures other than to inform the user of the existence of the error.
Possible Error Effect:	Since this check occurred after it was determined that an RSM number existed for an HSM, the audit will simply go on to the next RSM, if any, hanging off the HSM.
Corrective Action Taken:	None.
Manual Action Required:	If this error is rolling that user should investigate the condition.
Dump Description:	
<i>Error Address</i> =	Address of the rIMODATT tuple key.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to the rIMODATT tuple which is the RSM module number.
<i>Dump</i> =	NA.

Table 43 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Corresponding rHSMRSM tuple could not be found in the database. This error will continue to print out until fixed by the user.
Possible Error Effect:	May cause asserts and/or interrupts by various application programs.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, insert the missing rHSMRSM tuple using the key which is dumped in the Logical Index of this error report.

Dump Description:	
<i>Error Address</i> =	Address of rHSMRSM tuple buffer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (<i>hsmnum</i>) of the rHSMRSM tuple buffer.
<i>Dump</i> =	NA.

Table 44 Error Code: RCMISOL

Error Code:	RCMISOL
Error Description:	The communication module vintage is not CM2 and the data delivery keys DDRCMISOL and DDSCMISOL are not cleared. These two keys can only be set in CM2 offices.
Possible Error Effect:	The DD keys will contain the incorrect status. As a result, communication module maintenance functionality may not be available.
Corrective Action Taken:	Write both the DDRCMISOL and DDSCMISOL keys to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The data delivery DDSCMISOL data.
<i>Bad Data</i> =	The data delivery DDRCMISOL data.
<i>Good Data</i> =	Gldm0vint.
<i>Logical Index</i> =	Gldm1vint.
<i>Dump</i> =	NA.

Table 45 Error Code: RSMINITO

Error Code:	RSMINITO
Error Description:	The RMSINIT byte for an RSM is not zero.
Possible Error Effect:	The initialization state of the RSM is incorrect.
Corrective Action Taken:	The initialization state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Initialization state of the RSM in the DDRSMINIT byte.
<i>Good Data</i> =	Correct initialization states of the RSM.
<i>Logical Index</i> =	Module number whose initialization state is incorrect.
<i>Dump</i> =	NA.

Table 46 Error Code: RSMINITO

Error Code:	RSMINITO
Error Description:	The RSMINIT byte for an HSM is not zero.
Possible Error Effect:	The initialization state of the HSM is incorrect.
Corrective Action Taken:	The initialization state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Initialization state of the HSM in the DDRSMINIT byte.
<i>Good Data</i> =	Correct initialization states of the HSM.
<i>Logical Index</i> =	Module number whose initialization state is incorrect.
<i>Dump</i> =	NA.

Table 47 Error Code: RSMSEPO

Error Code:	RSMSEPO
Error Description:	The RSMSEP byte for an RSM is not zero.
Possible Error Effect:	The separation state of the RSM is incorrect.
Corrective Action Taken:	The separation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Separation state of the RSM in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the RSM.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 48 Error Code: RSMSEPO

Error Code:	RSMSEPO
Error Description:	The RSMSEP byte for an HSM is not zero.
Possible Error Effect:	The separation state of the HSM is incorrect.
Corrective Action Taken:	The separation state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the HSM in the DDRSMSEP byte.
<i>Good Data =</i>	Correct separation states of the HSM.
<i>Logical Index =</i>	Module number whose separation state is incorrect.
<i>Dump =</i>	NA.

Table 49 Error Code: SCMISOL

Error Code:	SCMISOL
Error Description:	The CM isolation DD status key DDSCMISOL is incorrectly set in the pending state (DDPCMISOL). A <i>Good Data</i> value of 1 indicates that the CM isolation DD request key DDRCMISOL is not normal and entry into manual CM isolation should be in progress. However, there is no CM isolation request on the SMARS queue. A <i>Good Data</i> value of 0 indicates that DDRCMISOL is normal and an AM/CM re-synchronization should be in progress. However, the SMrsmode re-synchronization flag is SMRSNULL.
Possible Error Effect:	The DDSCMISOL DD key is stuck in what should be a transient pending state giving incorrect CM isolation status and stubbing off PCD auditing. Some communication module maintenance functionality may not be available.
Corrective Action Taken:	Write the DDSCMISOL key to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value for DDRCMISOL.
<i>Bad Data =</i>	Value for DDSCMISOL.
<i>Good Data =</i>	Either 0 or 1, see Error Description.
<i>Logical Index =</i>	Value for SMrsmode.
<i>Dump =</i>	NA.

Table 50 Error Code: STUCK

Error Code:	STUCK
Error Description:	The RSMISOL byte for an RSM is not zero.
Possible Error Effect:	Invalid state of the RSM single RSM offices should always have the isolation bits set to zero.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation byte DDRSMISOL.
<i>Good Data =</i>	Written value of zero to DMRSMISOL.
<i>Logical Index =</i>	RSM module number.
<i>Dump =</i>	NA.

Table 51 Error Code: TYPE

Error Code:	TYPE
Error Description:	A switching module (SM) type in data delivery (DD) key DDSM2000 is inconsistent with RLMODATT (rlMODATT.unitttype = SMSM2000 and DDSM2000 = DD CLEAR for the SM or rlMODATT.unitttype != SMSM2000 and DDSM2000 = DD SET for the SM).
Possible Error Effect:	Call processing may take incorrect actions for the given SM.
Corrective Action Taken:	The audit writes DDSM2000 to agree with rlMODATT.unitttype. That is, if

	rIMODATT.unittype = SMSM2000, the audit writes the value of DDSM2000 to DDSET for this SM. Otherwise, the audit writes the value of DDSM2000 for this SM to DDCLEAR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value in DDSM2000 for the SM.
<i>Good Data =</i>	SM type (rIMODATT.unittype)
<i>Logical Index =</i>	SM number (rIMODATT.module)
<i>Dump =</i>	rIMODATT tuple for this SM.

Table 52 Error Code: UEQRSMISOL

Error Code:	UEQRSMISOL
Error Description:	The RSMISOL bit for an unequipped RSM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	Offset of the unequipped RSM.
<i>Dump =</i>	NA.

Table 53 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up, after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	Reinitialize audit global data, return audit complete, and wait for next cycle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	AUddindex (tells audit which SM to check next).

DD-C

Software Release: 5E14 and later

1. DATA DELIVERY (DD) AUDIT

The data delivery (DD) mechanism and audit covers data in the operational kernel process (OKP) and switch maintenance kernel process (SMKP) environments in the administrative module (AM), communications module processor (CMP) as well as the operating system for distributed switching-switching modules (OSDS-SMs) environment. The audit verifies the consistency of different data delivery data against other static data, data delivery, and global data. The audit reports errors it detects and makes the slave data consistent with available master data. If no master data exists, the data delivery data is set to a safe state. Data delivery data in incorrect states could disable call processing and cause problems for maintenance, billing, and growth; hence the need for the DD audit.

The DD audit runs in the CMP, OKP, SM, and SMKP environments in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DD audit checks the consistency of a number of different DD data structures. The term "DD key" refers to an identifier for information maintained by data delivery. Some of these data structures exist in the AM as AM-only data, some exist in the AM on a per SM basis, some exist in the AM on a per message class basis, and some of the data exists only in the SMs. The DD data structures checked are referenced by the following DD keys:

DDAPDLDSL	DDDIOPMODEL	DDPCONFIGSTL	DDRSMSEP
DDASPTRIGL	DDEQSTATL	DDPSSWSTATL	DDRTCDELNK
DDAUEWRKL	DDHOSTNUML	DDPVNALWL	DDRTCENET
DDCM_VINTL	DDIMYPEL	DDRISOLL	DDSASWC
DDCNIMINL	DDMATECMPL	DDRMSWCKL	DDSCMISOL
DDCNIOOSL	DDMGDSCBL	DDRSEPL	DDSINIT
DDCTS0L	DDMGLOGBL	DDRSMINITL	DDSMIN
DDCTS1L	DDMGPRTL	DDRSMISOLL	DDSREX
DDDASCDL			

3. AUDIT CHECKS

The audit makes the following checks and takes the specified actions.

Check	Environment	Action
DDAPDLDSL	SM	Checks the new data delivery key DDAPDLDSL against the AP link status table (PSaplst[]). DDAPDLDSL provides a summary of the status of the AP data link digital subscribe lines (DSLs) for a given SM. PSaplst[] contains information on the status of each of the individual AP data link DSLs. If the two values found do not match, an audit error is reported and DDAPDLDSL will be updated to agree with the information found in PSaplst[].
DDASPTRIG	OKP	Checks DD data associated with key DDASPTRIG against the data against the rIDS_APP tuple referenced by the key of CCAINR01. When the rIDS_APP tuple is not found, the DD data should indicate clear. When the rIDS_APP tuple is found and the global parameter is DBPVN_SSP, the DD data should be set.
DDAUEWRK	SMKP	Checks DD data associated with key DDAUEWRK against a global variable, AUEleact, which is maintained by audit control. When the global variable is set, the DD data should be set and when the global variable is clear, the DD data should indicate clear. If an inconsistency is detected, the DD data will be updated to be consistent with the global variable.
DDCM_VINT	OSDS-M	Checks DD data associated with key DDCM_VINT against the SM's office dependent data (ODD) global variable Glcm_vint. If an inconsistency is detected, the DD data will be updated to be consistent with the global variable. An audit error concerning this data is expected during the conversion of a communication module, model 1 (CM1) office to a CM2 office.

DDCNIMIN	OKP	Checks DD data associated with key DDCNIMIN against the DD data associated with key DDSMIN. If DDSMIN is set to indicate minmode, then DDCNIMIN should be set to indicate common network interface (CNI) minmode. If there is an inconsistency, both DD data will be set to not minmode.
DDCNIOOS	OKP	Checks DD data associated with key DDCNIOOS against the ODD variable Glccs. If an office is unequipped with CNI, the DD data should be set indicating that CNI is disabled. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDCTS0	OKP	Checks DD data associated with key DDCTS0 against the even CTS link number stored in MODATT. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDCTS1	OKP	Checks DD data associated with key DDCTS1 against the odd CTS link number stored in MODATT. If there is an inconsistency, the DD data will be made consistent with the ODD.
DDDIOPMODE	OKP	Checks DD data associated with key DDDIOPMODE. If the DD data is set to indicate disk independent mode, a message will be sent to application integrity monitor (AIM) to verify a valid disk independent mode. This will be done by AIM checking the essential disks statuses in the equipment configuration database (ECD). If an inconsistency exists, AIM will reset the DD data to be consistent with the ECD.
DDEQPROCS	OKP/CMP	Checks DD data associated with the key DDEQPROCS against the processors equipage status maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data is made consistent with the ODD.
DDEQSTAT	OKP/CMP	Checks DD data associated with key DDEQSTAT against the SM equipage status maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD. (The DD data here is different than that associated with key DDEQIMS.)
DDHOSTNUM	OKP/CMP	Checks DD data associated with key DDHOSTNUM against the host module number maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDIMTYPE	OKP	Checks DD data associated with key DDIMTYPE against the SM type maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDMATECMP	SMKP/CMP	Checks DD data associated with key DDMATECMP against the hostnum field maintained in the ODD relation RLMODATT. If an inconsistency exists, the DD data will be made consistent with the ODD.
DDMGDSCB	OKP	Checks DD data associated with key DDMGDSCB against DD data associated with keys DDMGLOGB and DDMGPRTB. If any of the DD data indicates to log or print messages, then the DD data for discard should not be set. If an inconsistency exists, the discard data will be set to not discard.
DDMGLOGB	OKP	This represents the DD data for logging messages. See the check for DDMGDSCB.
DDMGPRTB	OKP	This represents the DD data for printing messages. See the check for DDMGDSCB.
DDPCONFIGST	SMKP	Checks the key DDPCONFIGST for soft switch status by going off of the PCD state for the unit. In the case of a program update, the soft switch periodically resets the key so that the user knows that the program update uses the key.
DDPSSWSTAT	SMKP	Checks DD data associated with key DDPSSWSTAT against the soft switch status. If an inconsistency exists between the soft switch status and the PCD, the DD audit will update the DD key with the PCD information.
DDPVNALW	OKP	Checks DD data associated with key DDPVNALW against the existence of an rIDS_APP tuple referenced by the key of CCPVN (when Glpvn_ofc is set to DBPVN_SSP) or CCNS800. When the rIDS_APP tuple is not found, the DD data should indicate clear. When the rIDS_APP tuple is found and Glpvn_ofc is DBPVN_SSP or Glssp is DBYES, the DD data should be set.
DDRISOL	OSDS-M	Checks are done in a remote switching module (RSM) to verify that the separation information (DDRSEP) about an RSM in a multimodule site is consistent with the corresponding master data kept in DDRISOL.
DDRMSWCK	OSDS-M	Checks DD data associated with key DDRMSWCK against DDSASWCK data. If DDSASWCK data is off-normal, DDRMSWCK data should be off-normal.

DDRSEP	OSDS-M	The data in DDRSEP is the master copy representing the separation status of all the RSMs in a multimodule site it is arranged on a per module basis. The value in DDRSMSEP is compared against this data, and used in correcting any inconsistencies.
DDRSMINIT	OSDS-M	Checks are done to verify that the module initialization information (DDRSMINIT) about a module in a multimodule site is consistent with the corresponding master data kept in DDSINIT. When an inconsistency occurs, the DDRSMINIT status for a given module is set to the value of DDSINIT.
DDRSMISOL	OSDS-M	Checks are done to verify that the isolation status (DDRSMISOL) about a module in a multimodule site is consistent with the master status kept in DDRISOL. When an inconsistency occurs, the DDRSMISOL status for a given module is set to the value of DDRISOL.
DDRSMSSEP	OSDS-M	The data in this 10-bit word is arranged based on the number RSMs in a multimodule site. The separation status of each RSM is maintained here. Checks are done to verify that the separation status (DDRSMSSEP) is consistent with the master status kept in DDRSEP. When an error is found, the wrong bit in DDRSMSEP is set to the value of DDRSEP.
DDRTCENET, DDRTCDELNK	OKP	Checks DD data associated with the key DDRTCENET against the data delivery key DDRTCDELNK sent by the global billing SM. If the two values found do not match, an audit error is reported and DDRTCENET will be updated to agree with DDRTCDELNK.
DDSASWCK	OKP	Checks DD data associated with key DDSASWCK against DDRMSWCK data. If DDRMSWCK data is off-normal, DDSASWCK data should be off-normal.
DDSINIT	OSDS-M	Checks DD data associated with key DDSINIT against the global variable INmode. If an inconsistency exists, the DD data will be made consistent with the global variable.
DDSMCOUNT	OKP	Checks the new data delivery key DDSMCOUNT against the actual number of SMS (including special growth). If the two values do not match, an audit error is reported, and the DDSMCOUNT is updated with the correct value.
DDSMIN	SM	Checks SM DD data associated with DDSMIN against the global variable G1thrtl. If the DD data indicates minmode and the global variable does not, the DD data will be set to not minmode.
DDSREX	OKP	Checks DD data associated with key DDSREX against the inhibit allow status of routine exercises maintained in a global routine exercise table (DGldb[module].rxinh1st. enable1). If an inconsistency exists, the DD data will be made consistent with the global table.
DDSCMISOL	SMKP	Checks DD data associated with the key DDSCMISOL against the status of the CM isolation request DD key DDRCMISOL and either the SMARS job queue or the SMrsmode AM/CM re-synchronization flag. If an inconsistency exists, the DD data is written to normal.

4. ERROR HANDLING

If the ASP service features find errors, the data delivery bit is updated to be consistent with the static relation RLDS_APP.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_DDASPTRIG

Error Code:	ACT_DDASPTRIG
Error Description:	The data delivery data from the key DDASPTRIG is set to indicate ASP 0.1 activation. The static data in rlds_APP does not find a match using a key of CCAINR01, so data delivery data should indicate ASP 0.1 is de-activated.

Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The data delivery data DDASPTRIG is set to indicate ASP 0.1 de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The database return from reading the RLDS_APP relation.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: ACT_DDPVNALW

Error Code:	ACT_DDPVNALW
Error Description:	The value of the data delivery key, DDPVNALW, is set to indicate ASP activation. A match in static data rIDS_APP does not exist for either of the following keys: (1) A key of CCPVN when Glpvn_ofc is set to DBPVN_SSP, and (2) A key of CCNS800. Therefore, DDPVNALW should not be set.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The value of DDPVNALW is cleared to indicate ASP de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The database return from reading the RLDS_APP relation.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: BAD_DDTBL

Error Code:	BAD_DDTBL
Error Description:	An invalid state field (not AUDDDO, AUDDSKIP or AUDDDONE) in the data delivery audit table (AUdd1tbl[]), has been detected.
Possible Error Effect:	The data delivery audit may not be able to audit all the data items it is supposed to.
Corrective Action Taken:	Reinitialize the index (AUddindx), and start from the beginning of the audit the next time it is entered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local check table, AUdd1tbl[].
<i>Bad Data =</i>	The state element for the indicated table entry (AUdd1tbl[AUddindx].state).
<i>Good Data =</i>	Maximum array index which when added to 1 equals the number of entries in the table (AUdd_flag).
<i>Logical Index =</i>	Index into the table (AUddindx).
<i>Dump =</i>	The data delivery audit table (AUdd1tbl[]) of entries for the indicated processor.

Table 4 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	<i>In OKP or CMP:</i> The bit in DDeqmhppe[] associated with a particular switching module (SM) is set incorrectly. That is, either the bit indicates message handler (MH) connectivity for that SM when it should not (the bit is set) or the bit indicates no MH connectivity for that SM when the SM should have it (the bit is clear). See the dump description to determine which of the following cases applies [note that the "rIMODATT tuple" is the tuple associated with the particular SM (rIMODATT.module)]:

	- The bit is set but no rIMODATT tuple exists for the SM. - The bit is set but at least one of the rIMODATT <i>ql_ntwk0</i> or <i>ql_ntwk1</i> fields has value DBNO. - The bit is clear but both of the rIMODATT <i>ql_ntwk0</i> and <i>ql_ntwk1</i> fields have value DBYES. <i>In an SM-2000:</i> The SM's copy of DDeqmhpipe[] is inconsistent with the <i>qends[]</i> fields in the rIQENDPTS tuples where rIQENDPTS.qservice = QLOSDS (representing MH connectivity) and rIQENDPTS.qserv_num = 0 (the default value, used for non-CCS connectivity). Specifically, for each bit in DDeqmhpipe[], that bit should be set if, and only if, the corresponding bits are set in rIQENDPTS.qends[] for both networks (rIQENDPTS.qnetwork = QLNET0 and QLNET1).
Possible Error Effect:	If the bit is set when it should be clear, quad linked packet switching (QLPS) network traffic may be routed to an SM unable to accept it. If the bit is clear when it should be set, the QLPS network may be degraded because it is unable to route to the associated SM.
Corrective Action Taken:	<i>In OKP or CMP:</i> The audit triggers an update of the bit in DDeqmhpipe[] and any other DD data associated with the SM. <i>In SM-2000:</i> The audit triggers a reinitialization of DDeqmhpipe[].
Manual Action Required:	None.
Dump Description:	
<i>In OKP or CMP:</i>	
<i>Error Address =</i>	SM equipage status (rIMODATT.eqstat). A value of SMUNEQ may indicate that no rIMODATT tuple exists for this SM.
<i>Bad Data =</i>	rIMODATT.ql_ntwk0.
<i>Good Data =</i>	rIMODATT.ql_ntwk1.
<i>Logical Index =</i>	The associated SM number (used in rIMODATT.module when trying to read the rIMODATT tuple).
<i>Dump =</i>	NA.
<i>In an SM-2000:</i>	
<i>Error Address =</i>	Current entry in rIQENDPTS.qends[] for network QLNET0.
<i>Bad Data =</i>	Current entry in DDeqmhpipe[] being checked.
<i>Good Data =</i>	Current entry in rIQENDPTS.qends[] for network QLNET1.
<i>Logical Index =</i>	Index of the entry in DDeqmhpipe[] being checked. That is, the index of the first entry to have an error.
<i>Dump 1 =</i>	DDeqmhpipe[].
<i>Dump 2 =</i>	The rIQENDPTS tuple associated with network QLNET0.

Table 5 Error Code: CLIDAT

Error Code:	CLIDAT
Error Description:	An attempt failed to obtain a rCLIDAT tuple with a given cluster ID. The audit has not taken any recovery measures other than to inform the user of the existence of the error.
Possible Error Effect:	Since this check occurred after it was determined that an RSM number existed for an HSM, the audit will simply go on to the next RSM, if any, that is hanging off the HSM.
Corrective Action Taken:	None.
Manual Action Required:	If this error is rolling, the user should investigate the condition. Most likely the

	office information for the RLMODATT and RLCLIDAT relations need correction.
Dump Description:	
<i>Error Address</i> =	Address of the rCLIDAT buffer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Cluster ID to which RSM belongs.
<i>Dump</i> =	rCLIDAT buffer.

Table 6 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's count status and that value calculated by the AUddsmcnt() function.
Possible Error Effect:	Incorrect SM count status could be accessed.
Corrective Action Taken:	The data delivery data associated with the SM's count status will be made consistent with that calculated in the AUddsmcnt() function.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery SM count status.
<i>Good Data</i> =	SM count from AUddsmcnt().
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 7 Error Code: DD_MODATT

Error Code:	DD_MODATT
Error Description:	The audit can't read the ODD relation RLMODATT with the given generic plan (GP) number.
Possible Error Effect:	The GP audit data checks may not be done for a particular GP.
Corrective Action Taken:	None.
Manual Action Required:	Validate the GP number that the checks are failing for. Validate the tuple for this GP in the ODD relation RLMODATT.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return from DBfrdtup.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The number of the GP that the checks can't be done for.
<i>Dump</i> =	The tuple from the ODD relation RLMODATT for the GP in question.

Table 8 Error Code: DDAMABIL

Error Code:	DDAMABIL
Error Description:	The AM's billing stream indicators maintained in AMsmmap on a per SM basis do not agree with those maintained in the ODD relation RLMODATT.
Possible Error Effect:	Billing data may be sent to the wrong billing stream.
Corrective Action Taken:	Make the AMsmmap billing stream indicators consistent with the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	AMsmmap data for this SM.
<i>Good Data</i> =	ODD rIMODATT data for this SM.
<i>Logical Index</i> =	The number of the SM whose billing stream indicator data is in error.
<i>Dump</i> =	Copy of the rIMODATT tuple of the SM in question.

Table 9 Error Code: DDAUEWRK

Error Code:	DDAUEWRK
Error Description:	The relationship between data delivery data associated with key DDAUEWRK and global variable AUEleact, is not correct.
Possible Error Effect:	The routine and elevated audits in OKP of the AM may never be scheduled to run.
Corrective Action Taken:	The data delivery data associated with key DDAUEWRK will be made consistent

	with global variable AUEleact.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The data delivery DDAUEWRK data.
<i>Good Data</i> =	The global variable AUEleact, maintained by audit control.
<i>Logical Index</i> =	The mode the audit was running in.
<i>Dump</i> =	NA.

Table 10 Error Code: DDCMVINT

Error Code:	DDCMVINT
Error Description:	The audit has detected an inconsistency between the SM's data delivery data associated with key DDCM_VINT and the ODD global variable Glcm_vint.
Possible Error Effect:	Speed of the XPC chips may be temporarily lost.
Corrective Action Taken:	Set the data delivery data to agree with that in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery DDCM_VINT data.
<i>Good Data</i> =	ODD Glcm_vint.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 11 Error Code: DDCNIOOS

Error Code:	DDCNIOOS
Error Description:	The audit has detected an inconsistency between data delivery data associated with the office CNI equipage state and the office CNI equipage state maintained in the ODD variable Glccs.
Possible Error Effect:	CNI unique software may be executing abnormally.
Corrective Action Taken:	The data delivery data associated with the office CNI equipage state will be made consistent with the office CNI equipage state maintained in the ODD variable Glccs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery office CNI equipage state.
<i>Good Data</i> =	ODD office CNI equipage state.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 12 Error Code: DDEQIMS

Error Code:	DDEQIMS
Error Description:	The audit has detected an inconsistency in data delivery data between the processor's static equipage status (rIMODATT.eqstat) and the processor's dynamic equipage status in the DDeqims[] array.
Possible Error Effect:	Calls could be incorrectly routed to the processor.
Corrective Action Taken:	The data delivery data associated with the processor's equipage status will be made consistent with the processors equipage status maintained in the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery processor equipage data (bit from DDeqims[] array).
<i>Good Data</i> =	ODD processor equipage data (rIMODATT.eqstat).
<i>Logical Index</i> =	The number of the processor whose equipage is in question (rIMODATT.module).
<i>Dump</i> =	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 13 Error Code: DDEQPROC

Error Code:	DDEQPROC
Error Description:	The audit has detected an inconsistency between data delivery data associated

	with a processors equipage status and the processors equipage status maintained in the ODD relation RLMODATT.
Possible Error Effect:	Calls could be incorrectly routed to the processor.
Corrective Action Taken:	The data delivery data associated with the processors equipage state will be made consistent with the processors equipage state maintained in the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery processor equipage data.
<i>Good Data =</i>	ODD processor equipage data.
<i>Logical Index =</i>	The number of the processor whose equipage is in question.
<i>Dump =</i>	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 14 Error Code: DDEQSTAT

Error Code:	DDEQSTAT
Error Description:	The audit has detected an inconsistency between data delivery data associated with a processor's equipage status and the processor's equipage status maintained in the ODD relation RLMODATT. With the introduction of the CMP, if the DDEQSTAT error rolls out most likely there is an inconsistency between the AM's version of RLMODATT and the CMP's version of RLMODATT.
Possible Error Effect:	Calls could be incorrectly routed to the SM.
Corrective Action Taken:	The data delivery data associated with the SM's equipage state will be made consistent with the SM equipage state maintained in the ODD relation RLMODATT.
Manual Action Required:	Manual action is required only if the error is rolling out for the same SM. The office database editor (ODBE) should be used to make the CMP and AM's versions of RLMODATT consistent. The tuple to be updated is keyed off of the processor number given in the logical index field of the Dump Description.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery processor equipage data.
<i>Good Data =</i>	ODD processor equipage state.
<i>Logical Index =</i>	The number of the processor whose equipage is in question.
<i>Dump =</i>	Copy of the tuple for this processor from the ODD relation RLMODATT.

Table 15 Error Code: DDHOSTNUM

Error Code:	DDHOSTNUM
Error Description:	The audit has detected an inconsistency between data delivery data associated with a module's host SM number and the module's host SM number maintained in the ODD relation RLMODATT.
Possible Error Effect:	Fault recovery actions for an RSM may fail.
Corrective Action Taken:	The data delivery data associated with the module's host SM number will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery host SM number data.
<i>Good Data =</i>	ODD host SM number data.
<i>Logical Index =</i>	The number of the SM whose host number is in question.
<i>Dump =</i>	Copy of the tuple for this SM from the ODD relation RLMODATT.

Table 16 Error Code: DDIMTYPE

Error Code:	DDIMTYPE
Error Description:	The audit has detected an inconsistency between data delivery data associated with an SM's type and the SM's type maintained in the ODD relation RLMODATT.
Possible Error Effect:	Various commands associated with the type of SM may not give accurate results.

Corrective Action Taken:	The data delivery data associated with the SM's type will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery SM type data.
<i>Good Data</i> =	ODD SM type data.
<i>Logical Index</i> =	The module number of the module whose type is in question.
<i>Dump</i> =	Copy of the tuple for this SM from the ODD relation RLMODATT.

Table 17 Error Code: DDINDEX

Error Code:	DDINDEX
Error Description:	The index into the data delivery audit table, AUdd1tbl[, is out of range.
Possible Error Effect:	The data delivery audit may not be able to audit all the data items it is supposed to, or an out of range address may be tried and executed.
Corrective Action Taken:	Reinitialize the index, and start from the beginning of the audit the next time it is entered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the local check table, AUdd1tbl[.]
<i>Bad Data</i> =	The index that is out of range (AUddindx).
<i>Good Data</i> =	Maximum array index which when added to 1 equals the number of entries in the table (AUdd flag).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 18 Error Code: DDMATECMP

Error Code:	DDMATECMP
Error Description:	The audit has detected an inconsistency between data delivery data associated with a CMP's mate number and the CMP's hostnum field maintained in the ODD relation RLMODATT.
Possible Error Effect:	Bad return value from reading for a CMP's mate.
Corrective Action Taken:	The data delivery data associated with the CMP's mate number will be made consistent with that maintained in the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery mate CMP number data.
<i>Good Data</i> =	ODD hostnum field.
<i>Logical Index</i> =	The number of the CMP whose mate number is in question.
<i>Dump</i> =	Copy of the tuple for this CMP from the ODD relation RLMODATT.

Table 19 Error Code: DDMGDSCB

Error Code:	DDMGDSCB
Error Description:	The audit has detected an inconsistency between data delivery data associated with the log, print, and discard status of a message class.
Possible Error Effect:	Messages of a particular message class may be discarded when they should not be.
Corrective Action Taken:	The data delivery data associated with the print status will be set to print the messages. At the same time, internally to data delivery, the discard status for the message class will be turned off so as not to discard the messages associated with the message class involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Data delivery log status for the message class. This is not an address, it's a data value.
<i>Bad Data</i> =	Data delivery discard status for the message class.
<i>Good Data</i> =	Data delivery print status for the message class. This is not necessarily good data, it's just some data that's used in determining consistency.
<i>Logical Index</i> =	Index into data delivery message class arrays.
<i>Dump</i> =	NA.

Table 20 Error Code: DD_MODATT

Error Code:	DD_MODATT
Error Description:	The audit can't read the ODD relation RLMODATT with the given SM number.
Possible Error Effect:	The AM per SM audit data checks may not be done for a particular SM.
Corrective Action Taken:	None.
Manual Action Required:	Validate the SM number that the checks are falling for. Validate the tuple for this SM in the ODD relation RLMODATT.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return from DBfrdtup.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The number of the SM that the checks can't be done for.
<i>Dump =</i>	The tuple from the ODD relation RLMODATT for the SM in question.

Table 21 Error Code: DDRTCD

Error Code:	DDRTCD
Error Description:	The DDRTCDENET bit in the AM does not match DDRTCDELNK bit sent from the Global Billing SM.
Possible Error Effect:	Status information that controls whether real time billing data should be sent to the global billing SM is incorrect. Billing data may be inappropriately buffered (or not buffered) in an originating SM's real time billing memory (RTBM).
Corrective Action Taken:	The DDRTCDENET is set equal to DDRTCDELNK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DDRTCDENET.
<i>Good Data =</i>	DDRTCDELNK.
<i>Logical Index =</i>	Global billing SM number.
<i>Dump =</i>	NA.

Table 22 Error Code: DDSASWCK

Error Code:	DDSASWCK
Error Description:	The audit has detected an inconsistency between data delivery data associated with access keys DDSASWCK and DDRMSWCK.
Possible Error Effect:	The wrong inhibit status of automatic software checks may be displayed to the user.
Corrective Action Taken:	The data delivery data associated with access DDSASWCK will be made consistent with data delivery data associated with access key DDRMSWCK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery data DDSASWCK.
<i>Good Data =</i>	Data delivery data DDRMSWCK.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 23 Error Code: DDSINIT

Error Code:	DDSinIT
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's initialization state and global variable INmode.
Possible Error Effect:	An incorrect state of initialization could be indicated to the user.
Corrective Action Taken:	The data delivery data associated with the SM's initialization state will be made consistent with INmode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery SM initialization state.
<i>Good Data =</i>	INmode, SM's initialization state.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 24 Error Code: DDSINIT

Error Code:	DDSINIT
Error Description:	The audit has detected an inconsistency between data delivery data associated with the AM's initialization status and that data in Sldapdata table, maintained by AIM, that reflects the AM initialization progress.
Possible Error Effect:	Incorrect AM initialization status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the AM's initialization allow status will be made consistent with that maintained by AIM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery AM initialization status.
<i>Good Data =</i>	AIM AM initialization status.
<i>Logical Index =</i>	The number of the module whose status is in question.
<i>Dump =</i>	NA.

Table 25 Error Code: DDSMIN

Error Code:	DDSMIN
Error Description:	The audit has detected an inconsistency between data delivery data associated with the SM's minmode status and the G1thrt1 global variable used to throttle call originations.
Possible Error Effect:	An invalid minmode condition may be indicated to user.
Corrective Action Taken:	The data delivery data associated with the SM's minmode status will be made consistent with the G1thrt1 variable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery SM minmode status.
<i>Good Data =</i>	G1thrt1 global variable.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 26 Error Code: DDSMIN

Error Code:	DDSMIN
Error Description:	The relationship between data delivery data associated with AM minmode and CNI minmode is not correct.
Possible Error Effect:	An invalid AM minmode condition could be in effect.
Corrective Action Taken:	The data delivery data associated with AM minmode and CNI minmode are set to NOT AM minmode and NOT CNI minmode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The data delivery AM minmode data.
<i>Good Data =</i>	The data delivery AM CNI minmode data. (This should not be interpreted as the good data, but rather as the other data used to determine the inconsistency.)
<i>Logical Index =</i>	The mode the audit was running in.
<i>Dump =</i>	NA.

Table 27 Error Code: DDSREX

Error Code:	DDSREX
Error Description:	The audit has detected an inconsistency between data delivery data associated with a module's routine exercise inhibit/allow status and the inhibit/allow status maintained by REX.
Possible Error Effect:	Incorrect routine exercise inhibit/allow status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the module's inhibit/allow status will be made consistent with that maintained by REX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Data delivery REX inhibit/allow status.

<i>Good Data</i> =	REX inhibit/allow status (example, 00010203). Reading from left: 00 = Not used. 01 = REX status of fabric. 02 = REX status of ELS. 03 = REX status of DGN.
<i>Logical Index</i> =	The number of the module whose status is in question.
<i>Dump</i> =	NA.

Table 28 Error Code: DDSREX

Error Code:	DDSREX
Error Description:	The audit has detected an inconsistency between data delivery data associated with the CM's routine exercise inhibit/allow status and the inhibit/allow status maintained by REX.
Possible Error Effect:	Incorrect routine exercise inhibit/allow status could be reflected to user.
Corrective Action Taken:	The data delivery data associated with the CM's inhibit/allow status will be made consistent with that maintained by REX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data delivery REX inhibit/allow status.
<i>Good Data</i> =	REX inhibit/allow status.
<i>Logical Index</i> =	The number of the module whose status is in question.
<i>Dump</i> =	NA.

Table 29 Error Code: DDSVCDGRD

Error Code:	DDSVCDGRD
Error Description:	A non MMRS RSM has been found whose DDSVCDGRD ("service degraded") key is set. This key refers to the status of an SM's DLIs. Since RSMs use RLIs, the key should always be clear.
Possible Error Effect:	Erroneous link status information could cause faulty call handling.
Corrective Action Taken:	The DD data is reset to the correct value (DDCLEAR).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of the DDSVCDGRD key for this RSM.
<i>Good Data</i> =	The correct value for the key (DDCLEAR).
<i>Logical Index</i> =	The SM number.
<i>Dump</i> =	NA.

Table 30 Error Code: DEACT_DDASPTRIG

Error Code:	DEACT_DDASPTRIG
Error Description:	The data delivery data from the key DDASPTRIG is cleared to indicate ASP 0.1 activation. The static data in rIDS_APP finds a match using a key of CCAINR01 and the global parameter <i>Glpvn_ofc</i> is set to DBPVN_SSP, so data delivery data should indicate ASP is de-activated.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The data delivery data DDASPTRIG is set to indicate ASP 0.1 de-activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return code from DD read.
<i>Good Data</i> =	The correct value for the key (DDSET).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 31 Error Code: DEACT_DDPVNALW

Error Code:	DEACT_DDPVNALW
Error Description:	The value of the data delivery key, DDPVNALW, is clear to indicate ASP de-activation. A match in static data rIDS_APP exists for one of the following keys: (1) A key of CCPVN when Glpvn_ofc is set to DBPVN_SSP, or (2) A key of CCNS800. Also, the office type is a service switching point (SSP) or Glssp is set to DBYES. Therefore, DDPVNALW should be set.
Possible Error Effect:	Wasting SM data resources.
Corrective Action Taken:	The value of DDPVNALW is set to indicate ASP activation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from DD read.
<i>Good Data =</i>	The correct value for the key (DDSET).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 32 Error Code: EQPISOLSEP

Error Code:	EQPISOLSEP
Error Description:	An RSM in a MMRS site was found to be separated without isolation, in the master copies RISOL and RSEP respectively.
Possible Error Effect:	Reduction in the number of RSM available in the MMRS site.
Corrective Action Taken:	The separation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSM in RSEP.
<i>Good Data =</i>	Correct separation state of the RSM to the value of RISOL.
<i>Logical Index =</i>	Module number.
<i>Dump =</i>	NA.

Table 33 Error Code: EQPRSMISOL

Error Code:	EQPRSMISOL
Error Description:	The RSMISOL bit for an equipped RSM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	IM number whose isolation state is incorrect.
<i>Dump =</i>	NA.

Table 34 Error Code: EQPRSMINIT

Error Code:	EQPRSMINIT
Error Description:	The initialization status of an RSM in a MMRS site does not agree with the information kept in the master copy SINIT.
Possible Error Effect:	This causes loss of MMRS resources.
Corrective Action Taken:	The initialization state of the RSM is corrected to agree with SINIT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the RSMs in the DDRSMINIT byte.
<i>Good Data =</i>	Value of SINIT which is written on to DDRSMINIT.
<i>Logical Index =</i>	MMRS module number.
<i>Dump =</i>	NA.

Table 35 Error Code: EQPRSMSEP

Error Code:	EQPRSMSEP
Error Description:	The separation status of an RSM in a MMRSM site does not agree with the information kept in the master copy RSEP.
Possible Error Effect:	This causes loss of MMRSM resources.
Corrective Action Taken:	The separation state of the RSM is corrected to agree with RSEP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the RSMs in the DDRSMSEP byte.
<i>Good Data =</i>	Value of RSEP which is written on the DDRSMSEP.
<i>Logical Index =</i>	MMRSM module number.
<i>Dump =</i>	NA.

Table 36 Error Code: HIMRSMINIT

Error Code:	HIMRSMINIT
Error Description:	The RSMINIT byte for non MMRSM IMs is not zero.
Possible Error Effect:	The initialization state of the module is incorrect.
Corrective Action Taken:	The initialization state of the module is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the modules in the DDRSMINIT byte.
<i>Good Data =</i>	Correct initialization states of the modules.
<i>Logical Index =</i>	Module number whose initialization state is incorrect.
<i>Dump =</i>	NA.

Table 37 Error Code: HIMRSMISOL

Error Code:	HIMRSMISOL
Error Description:	The RSMISOL bit for an IM other than HIM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	IM number whose isolation state is incorrect.
<i>Dump =</i>	NA.

Table 38 Error Code: HIMRSMSEP

Error Code:	HIMRSMSEP
Error Description:	The RSMSEP byte for a non MMRSM IM is not zero.
Possible Error Effect:	The separation state of the module is incorrect.
Corrective Action Taken:	The separation state of the module is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Separation state of the modules in the DDRSMSEP byte.
<i>Good Data =</i>	Correct separation states of the modules.
<i>Logical Index =</i>	Module number whose separation state is incorrect.
<i>Dump =</i>	NA.

Table 39 Error Code: INV

Error Code:	INV
Error Description:	The DDRSEP byte for an HSM is not zero.
Possible Error Effect:	The separation state of the HSM is incorrect.
Corrective Action Taken:	The separation state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	Separation state of the HSM in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the HSM.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 40 Error Code: LNK_CTS_NUM

Error Code:	LNK_CTS_NUM
Error Description:	The data in either DDCTS0 or DDCTS1 pertaining to the even and odd CTS link numbers is inconsistent with the data maintained in the ODD relation RLMODATT.
Possible Error Effect:	Incorrect link information will be used.
Corrective Action Taken:	The data in either DDCTS0 or DDCTS1 will be updated to be consistent with the ODD relation RLMODATT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0 if error is pertaining to DDCTS0 or 1 if error is pertaining to DDCTS1.
<i>Bad Data</i> =	Return value from DDREAD call.
<i>Good Data</i> =	ODD rIMODATT data for this SM. Even CTS link number for DDCTS0 and ODD CTS link number for DDCTS1.
<i>Logical Index</i> =	The number of the SM whose link information is in error.
<i>Dump</i> =	Copy of the rIMODATT tuple of the SM in question.

Table 41 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	The DD key summary status does not match the actual value returned by TM.
Possible Error Effect:	A major or minor alarm may sound, when it should or not sound.
Corrective Action Taken:	The corresponding DD key will be updated with the appropriate value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Bad DD key value.
<i>Good Data</i> =	Good DD key value.
<i>Logical Index</i> =	The DD key that had the error.
<i>Dump</i> =	NA.

Table 42 Error Code: MODATT

Error Code:	MODATT
Error Description:	A remote switching module (RSM) from static relation RLHSMRSM (rIHSMRSM.rsmnum[index]) has no corresponding tuple in the RLMODATT relation. That is, a database read failed for relation RLMODATT using rIMODATT.module = rIHSMRSM.rsmnum[index] where the RSM number was valid (!= 0).
Possible Error Effect:	The RSM can not be made operational.
Corrective Action Taken:	None.
Manual Action Required:	Either RLHSMRSM is corrupt (it contains an incorrect RSM number) or the RSM has not been fully specified in the static ODD. Technicians should determine if the RSM number corresponds to a real SM and either remove the reference from RLHSMRSM or add all missing static ODD using recent change or ODBE. See poprules for RLMODATT and RLHSMRSM for more information.
Dump Description:	
<i>Error Address</i> =	Index into rIHSMRSM.rsmnum[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The host SM (HSM) number. This is the key to the rIHSMRSM tuple (rIHSMRSM.hsmnum).
<i>Logical Index</i> =	The RSM number. This is the key to the rIMODATT tuple (rIMODATT.module) for which the database read failed.
<i>Dump</i> =	The RLHSMRSM tuple.

Table 43 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Corresponding rHSMRSM tuple could not be found in the database. This error will continue to print out until fixed by the user.
Possible Error Effect:	May cause asserts and/or interrupts by various application programs.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, insert the missing rHSMRSM tuple using the key which is dumped in the Logical Index of this error report.
Dump Description:	
<i>Error Address =</i>	Address of rHSMRSM tuple buffer.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (<i>hsmnum</i>) of the rHSMRSM tuple buffer.
<i>Dump =</i>	NA.

Table 44 Error Code: RCMISOL

Error Code:	RCMISOL
Error Description:	The communication module vintage is not CM2 and the data delivery keys DDRCMISOL and DDSCMISOL are not cleared. These two keys can only be set in CM2 offices.
Possible Error Effect:	The DD keys will contain the incorrect status. As a result, communication module maintenance functionality may not be available.
Corrective Action Taken:	Write both the DDRCMISOL and DDSCMISOL keys to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The data delivery DDSCMISOL data.
<i>Bad Data =</i>	The data delivery DDRCMISOL data.
<i>Good Data =</i>	Glcml0vint.
<i>Logical Index =</i>	Glcml1vint.
<i>Dump =</i>	NA.

Table 45 Error Code: RSMINITO

Error Code:	RSMINITO
Error Description:	The RMSINIT byte for an RSM is not zero.
Possible Error Effect:	The initialization state of the RSM is incorrect.
Corrective Action Taken:	The initialization state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the RSM in the DDRSMINIT byte.
<i>Good Data =</i>	Correct initialization states of the RSM.
<i>Logical Index =</i>	Module number whose initialization state is incorrect.
<i>Dump =</i>	NA.

Table 46 Error Code: RSMINITO

Error Code:	RSMINITO
Error Description:	The RSMINIT byte for an HSM is not zero.
Possible Error Effect:	The initialization state of the HSM is incorrect.
Corrective Action Taken:	The initialization state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Initialization state of the HSM in the DDRSMINIT byte.
<i>Good Data =</i>	Correct initialization states of the HSM.
<i>Logical Index =</i>	Module number whose initialization state is incorrect.
<i>Dump =</i>	NA.

Table 47 Error Code: RSMSEPO

Error Code:	RSMSEPO
Error Description:	The RSMSEP byte for an RSM is not zero.
Possible Error Effect:	The separation state of the RSM is incorrect.
Corrective Action Taken:	The separation state of the RSM is corrected.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Separation state of the RSM in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the RSM.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 48 Error Code: RSMSEPO

Error Code:	RSMSEPO
Error Description:	The RSMSEP byte for an HSM is not zero.
Possible Error Effect:	The separation state of the HSM is incorrect.
Corrective Action Taken:	The separation state of the HSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Separation state of the HSM in the DDRSMSEP byte.
<i>Good Data</i> =	Correct separation states of the HSM.
<i>Logical Index</i> =	Module number whose separation state is incorrect.
<i>Dump</i> =	NA.

Table 49 Error Code: SCMISOL

Error Code:	SCMISOL
Error Description:	The CM isolation DD status key DDSCMISOL is incorrectly set in the pending state (DDPCMISOL). A <i>Good Data</i> value of 1 indicates that the CM isolation DD request key DDRCMISOL is not normal and entry into manual CM isolation should be in progress. However, there is no CM isolation request on the SMARS queue. A <i>Good Data</i> value of 0 indicates that DDRCMISOL is normal and an AM/CM re-synchronization should be in progress. However, the SMrsmode re-synchronization flag is SMRSNULL.
Possible Error Effect:	The DDSCMISOL DD key is stuck in what should be a transient pending state giving incorrect CM isolation status and stubbing off PCD auditing. Some communication module maintenance functionality may not be available.
Corrective Action Taken:	Write the DDSCMISOL key to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value for DDRCMISOL.
<i>Bad Data</i> =	Value for DDSCMISOL.
<i>Good Data</i> =	Either 0 or 1, see Error Description.
<i>Logical Index</i> =	Value for SMrsmode.
<i>Dump</i> =	NA.

Table 50 Error Code: STUCK

Error Code:	STUCK
Error Description:	The RSMISOL byte for an RSM is not zero.
Possible Error Effect:	Invalid state of the RSM single RSM offices should always have the isolation bits set to zero.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Isolation byte DDRSMISOL.
<i>Good Data</i> =	Written value of zero to DMRSMISOL.
<i>Logical Index</i> =	RSM module number.
<i>Dump</i> =	NA.

Table 51 Error Code: TYPE

Error Code:	TYPE

Error Description:	A switching module (SM) type in data delivery (DD) key DDSM2000 is inconsistent with RMODATT (rMODATT.unitttype = SMSM2000 and DDSM2000 = DDCLEAR for the SM or rMODATT.unitttype != SMSM2000 and DDSM2000 = DDSET for the SM).
Possible Error Effect:	Call processing may take incorrect actions for the given SM.
Corrective Action Taken:	The audit writes DDSM2000 to agree with rMODATT.unitttype. That is, if rMODATT.unitttype = SMSM2000, the audit writes the value of DDSM2000 to DDSET for this SM. Otherwise, the audit writes the value of DDSM2000 for this SM to DDCLEAR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value in DDSM2000 for the SM.
<i>Good Data =</i>	SM type (rMODATT.unitttype)
<i>Logical Index =</i>	SM number (rMODATT.module)
<i>Dump =</i>	rMODATT tuple for this SM.

Table 52 Error Code: UEQRSMISOL

Error Code:	UEQRSMISOL
Error Description:	The RSMISOL bit for an unequipped RSM does not match RISOL bit.
Possible Error Effect:	The isolation state of the RSM is incorrect.
Corrective Action Taken:	The isolation state of the RSM is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Isolation state of the RSMs in the DDRSMISOL byte.
<i>Good Data =</i>	Correct isolation states of RSMs.
<i>Logical Index =</i>	Offset of the unequipped RSM.
<i>Dump =</i>	NA.

Table 53 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up, after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	Reinitialize audit global data, return audit complete, and wait for next cycle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	AUddindex (tells audit which SM to check next).

DFIMP

Software Release: 5E10 and later

1. DIGITAL FACILITIES INTERFACE MICROPROCESSOR (DFIMP) AUDIT

The SLC® 96 digital facility interface (S-DFI) contains several circuits, including a time-slot interchange unit (TSI), a signal framer, and a microprocessor (MP) that controls the flow of signals through the S-DFI. Refer to Figure 1. Control information regulated by the microprocessor includes whether the derived data link (a sample of bits from the T1 line compared to a pattern for control purposes) is used and whether the TSI of the S-DFI is bypassed (it is bypassed for mode II lines). Other functions of the microprocessor include checking the format of signals between the S-DFI and the remote terminal and checking the status of the time assignment unit (TAU) bits. The microprocessor stores the control information in random access memory (RAM). The DFIMP audit verifies the information in the control RAM to ensure that service degradations do not result from errors such as overwriting a mode I S-DFI with mode II data.

The audit is part of the Switch Maintenance audits subgroup.

2. DATA STRUCTURES

DFIMP audit is the control RAM in the S-DFI microprocessor. The audit checks four bytes of information against the circuit data relation (RLCKTDATA) and various header files. The four bytes of information checked in the control RAM are the firmware configuration byte, the framer mode byte, the receive sink/transmit formatter (RS/TF) mode byte, and the TAU activity byte.

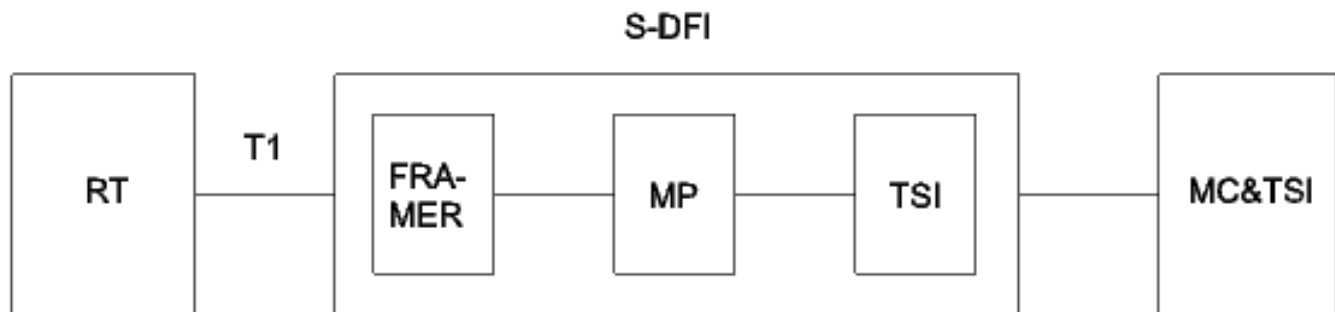


Figure 1 (DFIMP-1) DFIMP Hardware

3. AUDIT CHECKS

The audit reads the control RAM bytes and compares information in them with appropriate fields in RLCKTDATA and the header files.

- (a) When checking the firmware configuration byte, the audit determines whether the derived data link field is used, whether the S-DFI is bypassed, and other maintenance information.
- (b) The framer mode byte controls the format of the bit frames on the T1 lines. For American systems, the byte is set to zero because the T1 lines contain 24 time slots. The framer mode byte of other systems specifies different formats.
- (c) The receive sink/transmit formatter mode byte is checked to ensure that control signaling between the S-DFI and the remote terminal is correct.
- (d) The TAU activity byte, used only by mode II lines, indicates whether TAU activity should be on or off. This check is excluded if the S-DFI is not active.

- (e) The audit also ensures that the system on/off counter is on and the talk-back channel on the S-DFI is functioning when the S-DFI is active.

4. ERROR HANDLING

The audit corrects errors by sending a recovery message to the peripheral fault recovery program for the S-DFI. Peripheral fault recovery sends a message to the maintenance request administrator (MRA) program, which switches calls to the protection T1 line and its associated S-DFI. The faulty S-DFI is removed from service, diagnosed, initialized, and restored to service when all the diagnostic tests pass.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	Cannot access the RLCKTDATA relation.
Possible Error Effect:	Cannot initialize the S-DFI hardware.
Corrective Action Taken:	None.
Manual Action Required:	Yes.
Dump Description:	
<i>Error Address</i> =	Address of the rICKTDATA tuple.
<i>Bad Data</i> =	Expected value of the tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding riSMEST tuple.

Table 2 Error Code: C_FAIL_SLC

Error Code:	C_FAIL_SLC
Error Description:	The SLC [®] 96 A facility firmware configuration for the control information indicates not to send the C field.
Possible Error Effect:	The DFI will lose communication with the remote terminal.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the firmware configuration byte.
<i>Bad Data</i> =	Value of bits 0 and 1 of the byte.
<i>Good Data</i> =	Correct value of bits 0 and 1 of the byte.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding riSMEST tuple.

Table 3 Error Code: DDL_FLD

Error Code:	DDL_FLD
Error Description:	The firmware configuration byte is checked for DDL fields. None are expected.
Possible Error Effect:	None. Not a good error state.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the firmware configuration byte.
<i>Bad Data</i> =	Bits 0 and 1 of the firmware configuration byte.
<i>Good Data</i> =	Correct value for bits 0 and 1.

<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rISMEST tuple.

Table 4 Error Code: DDL_MODE

Error Code:	DDL_MODE
Error Description:	The firmware configuration byte is checked for D4 mode.
Possible Error Effect:	The DFI will go out of frame and lose communication with the PICB.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the firmware configuration byte.
<i>Bad Data</i> =	Bits 3 through 5 of the firmware configuration byte.
<i>Good Data</i> =	Correct value for bits 3 through 5.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rISMEST tuple.

Table 5 Error Code: DDL_MSG_FLD

Error Code:	DDL_MSG_FLD
Error Description:	The SLC [®] 96 A facility firmware configuration for the control information indicates not to send A, M, S and C fields.
Possible Error Effect:	The DFI will lose communication with the remote terminal.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the firmware configuration byte.
<i>Bad Data</i> =	Value of bits 0 and 1 of the byte.
<i>Good Data</i> =	Correct value of bits 0 and 1 of the byte.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rISMEST tuple.

Table 6 Error Code: DFI_MODE

Error Code:	DFI_MODE
Error Description:	Cannot find the firmware configuration byte in the ODD.
Possible Error Effect:	The S-DFI may not be initialized correctly.
Corrective Action Taken:	None.
Manual Action Required:	Yes. Incorrect office data is indicated.
Dump Description:	
<i>Error Address</i> =	Address of the rCKTDATA tuple being read.
<i>Bad Data</i> =	Expected value of the tuple.
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rISMEST tuple.

Table 7 Error Code: FIRMW_CONFIG

Error Code:	FIRMW_CONFIG
Error Description:	The control information in the firmware configuration byte of the SLC [®] 96 A facility is incorrect.
Possible Error Effect:	The Remote Terminal will not process calls. May also cause maintenance alarms.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the firmware configuration byte.
<i>Bad Data</i> =	Bits 0 and 1 of the firmware configuration byte.
<i>Good Data</i> =	Correct value for bits 0 and 1.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 8 Error Code: FRAME_MODE

Error Code:	FRAME_MODE
Error Description:	The framer mode byte is not marked SLC [®] 96.
Possible Error Effect:	The remote terminal will not work.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the framer mode byte.
<i>Bad Data</i> =	Value of bits 0 - 6 of the framer mode byte.
<i>Good Data</i> =	Correct value for bits 0 - 6.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 9 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Could not read the DFI circuit name from rIS96DFI.
Possible Error Effect:	Cannot initialize.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not used.
<i>Bad Data</i> =	Not used.
<i>Good Data</i> =	Expected value of the tuple.
<i>Logical Index</i> =	DFI name in the tuple.
<i>Dump</i> =	rIS96DFI structure.

Table 10 Error Code: MODE

Error Code:	MODE
Error Description:	One or more of bits 2 through 4 of the firmware configuration byte are incorrect.
Possible Error Effect:	May kill call processing. SLC [®] 96 will not function properly. May not be able to make phone calls. Stable calls are usually dropped.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the firmware configuration byte.
<i>Bad Data</i> =	Value of bits 2 through 4.
<i>Good Data</i> =	Correct value of bits 2 through 4.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 11 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	Echoed data from the DFI does not match data sent.
Possible Error Effect:	DFI isn't really working.
Corrective Action Taken:	Audit will send a message to SMIM requesting a restoral of the DFI.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the data returned by the DFI.
<i>Bad Data</i> =	Char received.
<i>Good Data</i> =	Complement of char sent.
<i>Logical Index</i> =	SDFI name.
<i>Dump</i> =	NA.

Table 12 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A S-DFI is in use, but the TAU activity byte is OFF for MODE II SLC [®] 96.
Possible Error Effect:	Turns off originations from shelf served by the DFI.
Corrective Action Taken:	Message is sent to SMIM to recover the SDFI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TAU activity byte.
<i>Bad Data</i> =	TAU activity byte.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	rIS96DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 13 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A S-DFI is not in use, but the TAU activity byte is ON for MODE II SLC [®] 96.
Possible Error Effect:	Can cause unwanted activity messages from a non-active DFI.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TAU activity byte.
<i>Bad Data</i> =	TAU activity byte.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	rIS96DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 14 Error Code: NOT_SLC

Error Code:	NOT_SLC
Error Description:	Bits 0 through 6 of the framer mode byte do not indicate SLC [®] 96.
Possible Error Effect:	The DFI may go out of frame and/or lose communication with the PICB.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the framer mode byte.
<i>Bad Data</i> =	Bits 0 through 6 of the framer mode byte.
<i>Good Data</i> =	Correct value for bits 0 through 6.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 15 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	Cannot find the firmware configuration byte in the ODD.
Possible Error Effect:	The S-DFI may not be initialized correctly.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the CKTDATA tuple being read.
<i>Bad Data</i> =	Expected value of the tuple.
<i>Good Data</i> =	Not used.

<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding riSMEST tuple.

Table 16 Error Code: RS_TF_BITS

Error Code:	RS_TF_BITS
Error Description:	Bits 2 through 6 of the RS/TF mode are incorrect.
Possible Error Effect:	No traffic will get through the S-DFI.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the RS_TF Mode byte.
<i>Bad Data</i> =	Value of bits 2 through 6 of the RS_TF Mode byte.
<i>Good Data</i> =	Correct value of bits 2 through 6.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding riSMEST tuple.

Table 17 Error Code: RS_TF_MODE

Error Code:	RS_TF_MODE
Error Description:	One or more of bits 2 through 6 of the RS_TF (receive sync transmit formatter) mode byte are incorrect.
Possible Error Effect:	Signaling between S-DFI and the Remote Terminal will be messed up (that is, can not ring).
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS and diagnosed. If it is ATP, it is initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the RS_TF Mode byte.
<i>Bad Data</i> =	Value of bits 2 through 6.
<i>Good Data</i> =	Correct value of bits 2 through 6.
<i>Logical Index</i> =	DFI circuit name.
<i>Dump</i> =	Corresponding riSMEST tuple.

Table 18 Error Code: SANCLK

Error Code:	SANCLK
Error Description:	SDFI system counter was turned off. Never valid for active DFI.
Possible Error Effect:	DFI isn't really working.
Corrective Action Taken:	Audit will send a message to SMIM requesting a restoral of the DFI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the data returned by the DFI.
<i>Bad Data</i> =	Char received.
<i>Good Data</i> =	Complement of char sent.
<i>Logical Index</i> =	SDFI name.
<i>Dump</i> =	NA.

Table 19 Error Code: SLC_MODE

Error Code:	SLC_MODE
Error Description:	Cannot find the S-DFI mode attribute in the ODD.
Possible Error Effect:	The S-DFI may not be initialized correctly.
Corrective Action Taken:	None.
Manual Action Required:	Yes. Incorrect office data is indicated.
Dump Description:	
<i>Error Address</i> =	Address of the riCKTDATA tuple being read.
<i>Bad Data</i> =	Expected value of the tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DFI circuit name.

<i>Dump</i> =	Corresponding rSMEST tuple.

Table 20 Error Code: STUFF_PAT

Error Code:	STUFF_PAT
Error Description:	The stuffing pattern for Mode II concentrated SLC [®] 96 is incorrect.
Possible Error Effect:	Some calls will be blocked from getting through the DFI.
Corrective Action Taken:	A message is sent to PFR, which then sends a message to MRA to do a protection line switch. The S-DFI in question is taken OOS, diagnosed, (if ATP) initialized and put back into service by performing a protection line switch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the ODD stuffing pattern.
<i>Bad Data</i> =	Value of the ODD stuffing pattern.
<i>Good Data</i> =	DFI stuffing pattern.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 21 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	Failed to get response from active DFI. It shouldn't be active if it won't respond.
Possible Error Effect:	DFI isn't really working.
Corrective Action Taken:	Audit will send a message to SMIM requesting a restoral of the DFI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCKTDATA tuple.
<i>Bad Data</i> =	Expected value of the tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rS96DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

Table 22 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	Could not read the DFI circuit name from rCKTDATA.
Possible Error Effect:	Cannot initialize the S-DFI hardware.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCKTDATA tuple.
<i>Bad Data</i> =	Expected value of the tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rS96DFI circuit name.
<i>Dump</i> =	Corresponding rSMEST tuple.

DFISHELF

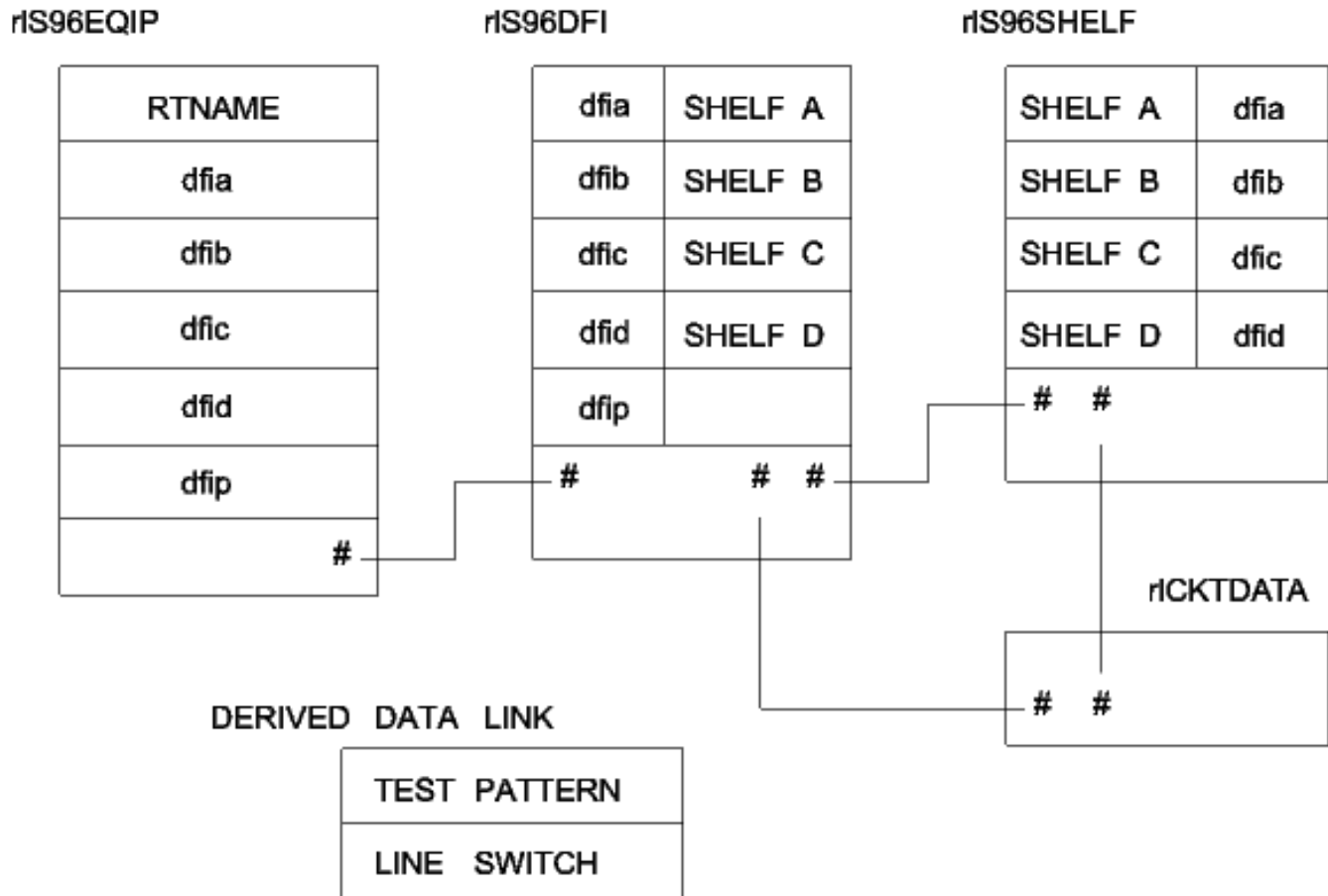
Software Release: 5E10 and later

1. DIGITAL FACILITIES INTERFACE SHELF (DFISHELF) AUDIT

The *SLC*® 96 remote terminal contains as many as four shelves of customer lines; 24 customer lines may terminate on one shelf. Two shelves make up one shelf group, or bank. The T1 lines, each of which contains 24 time slots, are allocated by shelf for mode I (unconcentrated) service and by bank for mode III (concentrated) service. The T1 lines terminate on *SLC*® 96 digital facility interfaces (S-DFIs) in the digital carrier line unit (DCLU) in the switching module. The digital facilities interface shelf audit (DFISHELF) ensures that the shelves of each remote terminal are assigned properly to the S-DFIs. The audit is part of the Switch Maintenance audits subgroup.

2. DATA STRUCTURES

This audit checks two dynamic relations, the *SLC*® 96 digital facilities interface relation (RLS96DFI) and the *SLC*® 96 shelf relation (RLS96SHELF), and the static *SLC*® 96 equipment configuration relation (RLS96EQUIP). As Figure 1 shows, the RLS96DFI relation contains the name of the shelf that each S-DFI serves. The tuples of the relation are indexed by the S-DFI name; each tuple contains the name of the shelf associated with it. There is a separate tuple for the protection S-DFI. The RLS96SHELF relation contains a tuple for each equipped shelf or shelf group. The tuples of the relation are indexed by the shelf name; each tuple contains the name of the S-DFI associated with it. The static RLS96EQUIP relation contains the name of the remote terminal and the S-DFIs associated with the shelves in it.



——— # = One-To-One Linkage

Figure 1 (DFISHELF-1) Data Structures of the DFISHELF Audit

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) to read a tuple of RLS96EQUIP and obtain an S-DFI circuit name. Using the S-DFI circuit name, the audit reads rIS96DFI and checks the shelf circuit name. Valid values are either a shelf circuit name or null. The audit proceeds to RLS96SHELF and verifies that the tuple with the shelf circuit name obtained in RLS96DFI contains the same S-DFI circuit name. After the audit makes these consistency checks, it examines the line switch field of the derived data link (the derived data link is a sample of T1 signals used for control purposes). The line switch field tells the state of the protection line. Valid states are in-service, out-of-service, and the number of the shelf being served.

- If the protection line is not serving a shelf, the dfip field in rIS96DFI should be null, as it is in Figure 1 .
- If the protection line is serving a specific shelf (for example, shelf A), the dfip field in rIS96DFI must contain the name of shelf A. The tuple in rIS96SHELF for shelf A must contain the value of dfip. The audit ensures that the dfia field in rIS96DFI is null.
- If the protection line is being switched in or out, the audit verifies that a terminal process is associated with the switching activity. For every remote terminal it checks, the audit sends a message to the T1 peripheral fault recovery program to check the state of the shelves for the remote terminal. The audit steps through

RLS96DFI and RLS96SHELF and checks RLCKTDATA and RLS96EQUIP to determine the existence of the dynamic tuples in RLS96DFI and RLS96SHELF.

4. ERROR HANDLING

The DFISHELF audit recovers from errors by inserting tuples into RLS96DFI and RLS96SHELF based on the circuit names obtained from RLS96EQUIP, if needed. It then sends a recovery message to the T1 line peripheral fault recovery system process, which calls a recovery function. The audit treats consistency errors by initializing the data structures to their normal state (the state in which S-DFI a serves shelf A, S-DFI b serves shelf B, etc.) because the audit cannot determine the assignment of the protection line.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DDL_PDFI_EQP

Error Code:	DDL_PDFI_EQP
Error Description:	The protection line DFI is not equipped in rIS96EQUIP but the derived data link S field is not NULL.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Copy of the shelf name from the DDL.
<i>Bad Data</i> =	Shelf name from DDL.
<i>Good Data</i> =	Value indicating no shelf assigned to P-DFI.
<i>Logical Index</i> =	Remote terminal name.
<i>Dump</i> =	rIS96EQUIP tuple.

Table 2 Error Code: DDL_SHELF

Error Code:	DDL_SHELF
Error Description:	Inconsistent hardware (from S fields of derived data link) and software information (from rIS96DFI tuple) was detected.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Protection line rIS96DFI tuple.
<i>Bad Data</i> =	Shelf name from tuple.
<i>Good Data</i> =	Shelf name from the derived data link S field.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	Protection line rIS96DFI tuple.

Table 3 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	A shelf is undergoing a protection line switch, but the process performing it is invalid.
Possible Error Effect:	Possible problems routing calls.
Corrective Action Taken:	The remote terminal is initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Entry in table of processes performing PLS.
<i>Bad Data</i> =	PID in error.
<i>Good Data</i> =	Index into table of PLS processes.
<i>Logical Index</i> =	Remote terminal name from rIS96EQUIP tuple.
<i>Dump 1</i> =	PLS process.
<i>Dump 2</i> =	rIS96DFI tuple.
<i>Dump 3</i> =	rIS96SHELF tuple.

Table 4 Error Code: DER_DFI

Error Code:	DER_DFI
Error Description:	The dfi name in the rIS96SHELF tuple contains a different name than what was derived from the rIS96EQUIP tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIS96SHELF tuple.
<i>Bad Data =</i>	dfi name in that tuple.
<i>Good Data =</i>	dfi name derived from the rIS96EQUIP tuple.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rIS96SHELF tuple.
<i>Dump 2 =</i>	rIS96DFI tuple.

Table 5 Error Code: DER_SHELF

Error Code:	DER_SHELF
Error Description:	The shelf name in the rIS96DFI tuple contains a different name than what was derived from the rIS96EQUIP tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIS96DFI tuple.
<i>Bad Data =</i>	Shelf name in that tuple.
<i>Good Data =</i>	Shelf name derived from the rIS96EQUIP tuple.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rIS96DFI tuple.
<i>Dump 2 =</i>	rIS96SHELF tuple.

Table 6 Error Code: DFILOSTTUP

Error Code:	DFILOSTTUP
Error Description:	There is a tuple missing in the rIS96DFI relation.
Possible Error Effect:	Possible problems when routing SLC [®] 96 calls.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIS96DFI tuple.
<i>Bad Data =</i>	Relation ID.
<i>Good Data =</i>	Remote terminal name.
<i>Logical Index =</i>	Circuit name used to read the rIS96DFI relation.
<i>Dump 1 =</i>	rIS96DFI tuple.
<i>Dump 2 =</i>	rIS96EQUIP tuple.

Table 7 Error Code: DFI_MODE

Error Code:	DFI_MODE
Error Description:	An inconsistent shelf mode was detected between the rIS96DFI tuple and the rIS96EQUIP tuple.
Possible Error Effect:	Problems routing SLC [®] 96 calls.
Corrective Action Taken:	The shelf mode in the rIS96DFI tuple is changed to agree with the rIS96EQUIP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIS96DFI tuple.
<i>Bad Data =</i>	rIS96DFI shfmode field.
<i>Good Data =</i>	rIS96EQUIP bankmode field.
<i>Logical Index =</i>	Circuit name.
<i>Dump 1 =</i>	rIS96DFI tuple.
<i>Dump 2 =</i>	rIS96EQUIP tuple.

Table 8 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIS96DFI or rIS96SHELF tuple exists for rICKTDATA. The <i>Bad Data</i> field indicates which one.
Possible Error Effect:	Call processing resources will decrease due to hung tuples.
Corrective Action Taken:	If <i>Bad Data</i> equals 1, then the rIS96DFI tuple will be deleted from the data base because the corresponding static rICKTDATA tuple does not exist or is unequipped. If <i>Bad Data</i> equals 2, then the rIS96SHELF tuple will be deleted from the data base because the corresponding static rIS96EQUIP tuple does not exist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS96DFI or rIS96SHELF tuple.
<i>Bad Data</i> =	Error refers to an extra: 1 = rIS96DFI tuple. 2 = rIS96SHELF tuple.
<i>Good Data</i> =	1 = rICKTDATA or rIS96EQUIP does not exist. 2 = rICKTDATA is not equipped.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rIS96DFI or rIS96SHELF tuple.

Table 9 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a missing tuple in the rSMEST relation corresponding to the SHELF.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The remote terminal is initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rSMEST tuple.
<i>Bad Data</i> =	Relation ID for rSMEST.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rSMEST tuple.

Table 10 Error Code: LOST_SHELF

Error Code:	LOST_SHELF
Error Description:	There is a tuple missing in the rIS96SHELF relation.
Possible Error Effect:	Possible problems routing SLC [®] 96 calls.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS96SHELF tuple.
<i>Bad Data</i> =	Relation ID for rIS96SHELF.
<i>Good Data</i> =	Remote terminal name indicating which rIS96EQUIP tuple.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rIS96SHELF tuple.
<i>Dump 2</i> =	rIS96DFI tuple.

Table 11 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	A point-to-point-back check failed.
Possible Error Effect:	Loss of call processing resources due to unlinked tuples.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS96DFI or rIS96SHELF tuple.

<i>Bad Data</i> =	The shelf or dfl name field in that tuple in error.
<i>Good Data</i> =	Assumed name from rIS96EQUIP tuple.
<i>Logical Index</i> =	Remote terminal name.
<i>Dump 1</i> =	Original rIS96DFI tuple.
<i>Dump 2</i> =	rIS96SHELF tuple.
<i>Dump 3</i> =	Protection rIS96DFI tuple.

Table 12 Error Code: PLS_COMP

Error Code:	PLS_COMP
Error Description:	The rIS96DFI tuple indicates that a protection line switch has occurred, but the protection line DFI in the rIS96EQUIP tuple is not equipped.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS96DFI tuple which should have been replaced by the protection line DFI.
<i>Bad Data</i> =	Shelf field of the rIS96DFI tuple.
<i>Good Data</i> =	Shelf name from the rIS96EQUIP tuple.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	rIS96DFI tuple.
<i>Dump 2</i> =	rIS96SHELF tuple.
<i>Dump 3</i> =	rIS96EQUIP tuple.

Table 13 Error Code: PLS_PDFI

Error Code:	PLS_PDFI
Error Description:	The protection line DFI does not indicate that a protection line switch is happening (at least according to the rIS96DFI tuple).
Possible Error Effect:	Unknown.
Corrective Action Taken:	The remote terminal is initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS96DFI tuple (for the protection line).
<i>Bad Data</i> =	Shelf field of the protection line rIS96DFI tuple.
<i>Good Data</i> =	Value for protection line switching.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	Protection line rIS96DFI tuple.
<i>Dump 2</i> =	Original rIS96DFI tuple.
<i>Dump 3</i> =	rIS96SHELF tuple.

Table 14 Error Code: PLS_SDFI

Error Code:	PLS_SDFI
Error Description:	This rIS96DFI tuple is under protection line switch, as such the s96shelf field should be null and it is not.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Initialize the remote terminal.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS96DFI tuple.
<i>Bad Data</i> =	Shelf field of tuple.
<i>Good Data</i> =	Remote terminal name.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	Original rIS96DFI tuple.
<i>Dump 2</i> =	rIS96SHELF tuple.
<i>Dump 3</i> =	Protection rIS96DFI tuple.

Table 15 Error Code: SLCLOSTTUP

Error Code:	SLCLOSTTUP
Error Description:	There is a tuple missing in the rIS96DFI relation corresponding to a protection line DFI.
Possible Error Effect:	Possible problems routing SLC [®] 96 calls.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	rlS96DFI tuple.
<i>Bad Data</i> =	Relation ID.
<i>Good Data</i> =	Remote terminal name from rlS96EQUIP tuple.
<i>Logical Index</i> =	Circuit name.
<i>Dump 1</i> =	Protection rlS96DFI tuple.
<i>Dump 2</i> =	Original rlS96DFI tuple.
<i>Dump 3</i> =	rlS96SHELF tuple.

DFMAP

Software Release: 5E10 and later

1. DIGITAL SUBSCRIBER LINE FACILITIES MAP (DFMAP) AUDIT

The hardware that connects the "T" card in the line group (LGC) in the integrated services line unit (ISLU) to the protocol handlers (PHs) in the packet switching unit (PSU) includes the data fanout circuit (DF), time slot interchange unit (TSIU), peripheral interface data buses (PIDBs), directly connected PIDBs (DPIDBs), common data boards (CDs), and protocol handler data buses (PDBs), as Figure 1 shows. The PSU contains five shelves with 16 protocol handlers. There is one PDB per protocol handler, and 32 time slots per PDB. Digital subscriber line (DSL) maintenance audits check various aspects of this integrated service digital network (ISDN) hardware to ensure that the switch does not lose resources. The audits include the DFMAP audit, the integrated services port administration buffer audit (ISPABUF), and the terminal process control buffer audit (TPCBUF). Routine port conditioning (RPC) is also used to recover lost resources.

The DFMAP audit is part of the Peripheral Control audits subgroup.

2. DATA STRUCTURES

The DFMAP audit uses three data structures: the peripheral control data fanout map array (PCdfmap), the peripheral control data fanout link array (PCdflink), and the PIDB time slot data block relation (rIPTSB). The PCdfmap array is a bit map that contains a busy/idle entry for each time slot on the PDBs. The PCdflink array contains the key of the rIPTSB, which identifies the PIDB and time slot connected to the PDB. For correct interpretation of the data in PCdfmap and PCdflink, the DFMAP audit checks the circuit data relation (RLCKTDATA) and the switch maintenance equipment status relation (RLSMEST) to determine whether the protocol handler is in either the active or standby state. These data structures are shown in Figure 2 .

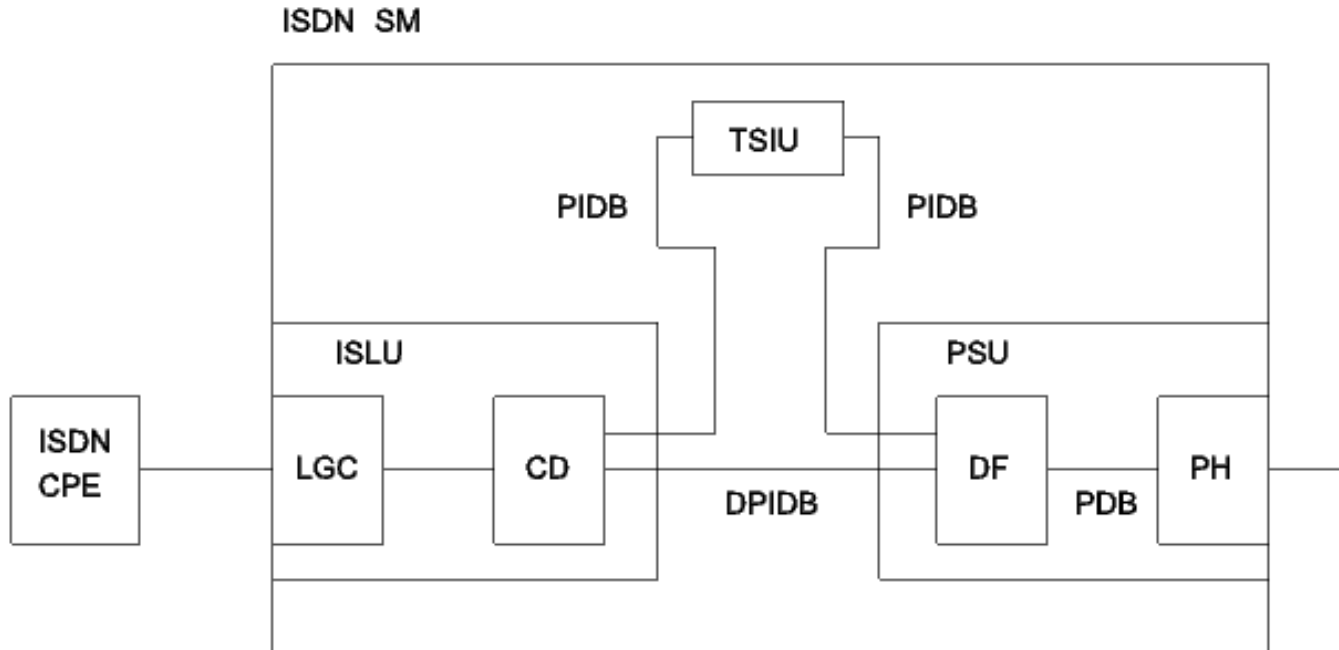


Figure 1 (DFMAP-1) ISLU Hardware

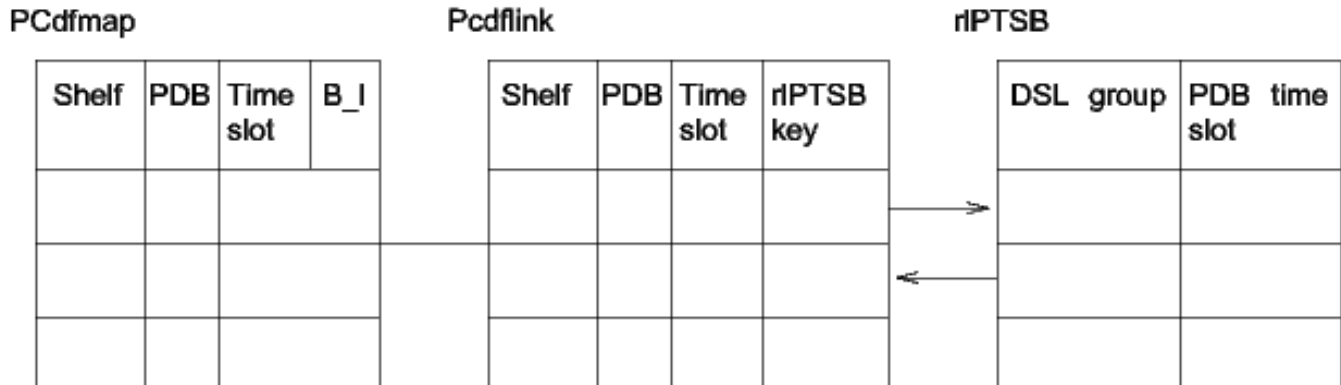


Figure 2 (DFMAP-2) Data Structures of the DFMAP Audit

3. AUDIT CHECKS

The DFMAP audit starts from the protocol handler side of the data fanout (DF) circuit and looks at information in rIPTSB on the DPIDB side. It makes the following checks.

- The audit determines whether each time slot entry in PCdfmap is busy or idle. It then checks the corresponding entry in PCdflink.
- If the PCdfmap entry is busy or idle and the PCdflink entry is not null, the audit assumes the PCdflink entry is a key and makes a point-to/point-back check to ensure that the corresponding tuple rIPTSB points back to the PDB on which the time slot exists.
- For a protocol handler that is neither active nor standby, the audit checks PCdfmap to ensure that all the time slot bits are busy and PCdflink to ensure that all the time slot entries are null.

When maintenance is being performed on a protocol handler, there is not a peripheral interface data bus (PIDB) associated with the PH data bus (PDB). However, all of the time slots on the PDB must be marked busy so that a path does not get set up using one of the time slots. This scenario requires that the PCdfmap be marked busy for all of the time slots on the given PDB, but a rIPTSB tuple will not exist. The PCdflink entry will contain PCDMYPTSB to indicate this situation.

4. ERROR HANDLING

Idle PCdfmap entries with a key in PCdflink and an rIPTSB tuple that points back to PCdflink are made busy. Busy PCdfmap entries with a key in PCdflink and no corresponding rIPTSB tuples are made idle and the PCdflink entry is made null.

When the semantic check function encounters the PCDMYPTSB value in PCdflink, it will not look for a rIPTSB. Instead, it will check the corresponding bit in PCdfmap to make sure that it is marked busy by calling the macro AUISIDLDFMAP(). If not, an error will be reported and the PCdfmap bit will be set to busy. The bit will be busied using the macro AUBSYDFMAP().

The DFMAP audits schedule the PTSB, CKTDATA, and PHDSLGP audits as related audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BUSY3MAP

Error Code:	BUSY3MAP
Error Description:	The rISMEST tuple indicates that the PH is neither active nor standby, but PCdfmap[][] is marked idle.
Possible Error Effect:	Path hunt might try to use the time slots.
Corrective Action Taken:	PCdfmap[][] is marked with all time slots busy for the given PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCdfmap[][].
<i>Bad Data =</i>	Value of PCdfmap[][].
<i>Good Data =</i>	0xffffffff.
<i>Logical Index =</i>	Circuit name of PH.
<i>Dump 1 =</i>	rUNITDATA tuple.
<i>Dump 2 =</i>	rISMEST tuple.

Table 2 Error Code: BUSY4MAP

Error Code:	BUSY4MAP
Error Description:	The rUNITDATA tuple indicates that the PH is not operational, but PCdfmap[][] is marked idle.
Possible Error Effect:	Path hunt might try to use the time slots.
Corrective Action Taken:	PCdfmap[][] is marked with all time slots busy for the given PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCdfmap[][].
<i>Bad Data =</i>	Value of PCdfmap[][].
<i>Good Data =</i>	0xffffffff.
<i>Logical Index =</i>	Circuit name of PH.
<i>Dump =</i>	rUNITDATA tuple.

Table 3 Error Code: BUSY5MAP

Error Code:	BUSY5MAP
Error Description:	There is no rUNITDATA tuple which indicates that the PH is not equipped, but PCdfmap[][] is marked idle.
Possible Error Effect:	Path hunt might try to use the time slots.
Corrective Action Taken:	PCdfmap[][] is marked with all time slots busy for the given PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCdfmap[][].
<i>Bad Data =</i>	Value of PCdfmap[][].
<i>Good Data =</i>	0xffffffff.
<i>Logical Index =</i>	Circuit name of PH.
<i>Dump =</i>	NA.

Table 4 Error Code: DSL_GRP

Error Code:	DSL_GRP
Error Description:	For an in-service PH, the DSL group found in the rIPTSB tuple does not match the indices of PCdflink[][] that was used to retrieve the key of the rIPTSB tuple.
Possible Error Effect:	The linkage between rIPTSB and PCdflink[][] has been broken.
Corrective Action Taken:	The time slot in PCdfmap[] idled and the PCdflink[][] is idled and the PCdflink[][] entry is set to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of dsl_group attribute in rIPTSB tuple.
<i>Bad Data =</i>	Value of dsl_group attribute in rIPTSB tuple.
<i>Good Data =</i>	Circuit name of PH.
<i>Logical Index =</i>	Key of rIPTSB tuple.
<i>Dump =</i>	rIPTSB tuple.

Table 5 Error Code: MAINT_STATE

Error Code:	MAINT_STATE

Error Description:	PCdflink[][] is marked with the PCDMYPTSB key which is used to indicate that a PDB time slot is being used for maintenance. But this is not possible because the PH is not out of service.
Possible Error Effect:	One of the time slots is being tied up needlessly.
Corrective Action Taken:	The PDB time slot in PCdfmap[][] is marked available (idle) and PCdflink[][] is marked with null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCdflink[][].
<i>Bad Data</i> =	Value of PCdflink[][].
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Circuit name of PH in first two bytes and PDB time slot in last two bytes.
<i>Dump</i> =	PCmnbstat structure.

Table 6 Error Code: MAINT_STATE2

Error Code:	MAINT_STATE2
Error Description:	According to the PCmnbstat structure, the PDB time slot is not being used for maintenance, although PCdflink[][] looks like it is.
Possible Error Effect:	One of the time slots is being tied up needlessly.
Corrective Action Taken:	The PDB time slot in PCdflink[][] is marked with null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCdflink[][].
<i>Bad Data</i> =	Value of PCdflink[][].
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Circuit name of PH in first two bytes and PDB time slot in last two bytes.
<i>Dump</i> =	PCmnbstat structure.

Table 7 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	PCdfmap[][] has a PDB time slot for an in-service PH marked busy but PCdflink[][] does not contain a key for a rIPTSB tuple.
Possible Error Effect:	One of the time slots is being tied up needlessly.
Corrective Action Taken:	The PDB time slot in PCdfmap[] does not contain a key for a rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCdfmap[][].
<i>Bad Data</i> =	Value of PCdfmap[][].
<i>Good Data</i> =	Circuit name of PH.
<i>Logical Index</i> =	PDB time slot.
<i>Dump</i> =	NA.

Table 8 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	For an in-service PH, the rIPTSB and PCdflink[][] data indicate that a D-channel path is set up on the given time slot, but PCdfmap[][] indicates that the time slot is idle.
Possible Error Effect:	Path hunt might try to use the time slot for another path.
Corrective Action Taken:	The time slot bit in PCdfmap[][] indicates that the time slot is idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCdfmap[][].
<i>Bad Data</i> =	Value of PCdfmap[][].
<i>Good Data</i> =	Circuit name of PH.
<i>Logical Index</i> =	PDB time slot in error.
<i>Dump</i> =	NA.

Table 9 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rSMEST tuple indicates that the PH is not active or standby, but PCdflink[][] is not marked with null.

Possible Error Effect:	Path hunt might try to use the time slots.
Corrective Action Taken:	PCdflink[][] is marked with null for the given PDB time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCdflink[][].
<i>Bad Data =</i>	Value of PCdflink[][].
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Circuit name of PH in first two bytes and PDB time slot in last two bytes.
<i>Dump 1 =</i>	rUNITDATA tuple.
<i>Dump 2 =</i>	rSMEST tuple.

Table 10 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The rUNITDATA tuple indicates that the PH is not operational, but PCdflink[][] is not marked with null.
Possible Error Effect:	Path hunt might try to use the time slot.
Corrective Action Taken:	PCdflink[][] is marked with null for the given PDB time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCdflink[][].
<i>Bad Data =</i>	Value of PCdflink[][].
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Circuit name of PH in first two bytes and PDB time slot in last two bytes.
<i>Dump =</i>	rUNITDATA tuple.

Table 11 Error Code: NULL5LINK

Error Code:	NULL5LINK
Error Description:	There is no rUNITDATA tuple which indicates that the PH is not equipped, but PCdflink[][] is not marked with null.
Possible Error Effect:	Path hunt might try to use the time slot.
Corrective Action Taken:	PCdflink[][] is marked with null for the given PDB time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCdflink[][].
<i>Bad Data =</i>	Value of PCdflink[][].
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Circuit name of PH in first two bytes and PDB time slot in last two bytes.
<i>Dump =</i>	NA.

Table 12 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	A rUNITDATA tuple was found for the PH, but a corresponding rSMEST tuple was not found.
Possible Error Effect:	The status (such as, ACT, STBY, OOS, and so forth) of the PH cannot be determined.
Corrective Action Taken:	Recovery of the missing rSMEST tuple is handled by the CKTDATA audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit Name of the PH.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	rUNITDATA tuple.

Table 13 Error Code: TSBUSYIDLE

Error Code:	TSBUSYIDLE
Error Description:	The audit found the tsuse field in the rIPTSB tuple associated with the PCdfmap[][] array entry under audit to be in an invalid state for the PCdfmap[][] entry to have been marked busy. For every PCdfmap[][] array entry, if there exists a corresponding rIPTSB tuple, a relationship should exist where:

	PCdfmap[PSIU shelf][PSIU PH] and PCdflink[PSIU shelf][PSIU PH][timeslot] corresponding array entries exist for a given PSIU shelf and PH, and PCdflink[PSIU shelf][PSIU PH][timeslot] is not 0 and not PCDMYPTSB, and (key) rIPTSB.ptsb = PCdflink[PSIU shelf][PSIU PH][timeslot], and rIPTSB.use is marked one of the legal full or quarter time slot "uses" as specified by type DMPTSBUSE.
Possible Error Effect:	ROP clutter and call failures since the invalid data will not be cleaned up until after the selective initialization completes and the DFMAP and PTSB audits are scheduled routinely.
Corrective Action Taken:	The corresponding PCdfmap[][] and PCdflink[][] entries for this PSIU shelf, PSIU PH, and time slot are idled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCdfmap[][] array entry under audit.
<i>Bad Data</i> =	The contents of the PCdfmap[][] array entry under audit.
<i>Good Data</i> =	The use field in the rIPTSB tuple.
<i>Logical Index</i> =	The logical key of the rIPTSB tuple (ptsb).
<i>Dump</i> =	The contents of the rIPTSB tuple.

Table 14 Error Code: TSMAP2

Error Code:	TSMAP2
Error Description:	For an in-service PH, the PDB time slot found in the rIPTSB tuple does not match the index of PCdflink[][] that was used to retrieve the key of the rIPTSB tuple.
Possible Error Effect:	The linkage between rIPTSB and PCdflink[][] has been broken.
Corrective Action Taken:	The time slot in PCdfmap[][] is idled and the PCdflink[][] entry is set to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPTSB tuple.
<i>Bad Data</i> =	Value of pdb_ts attribute in rIPTSB tuple.
<i>Good Data</i> =	Circuit name of PH in first two bytes and PDB time slot in last two bytes.
<i>Logical Index</i> =	Key of rIPTSB tuple.
<i>Dump</i> =	rIPTSB tuple.

Table 15 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

DFNIBBLE-A

Software Release: 5E10 - 5E11

1. DATA FANOUT NIBBLE AUDIT (DFNIBBLE)

The data fanout nibble (DFNIBBLE) audit monitors data fanout (DF) flexible time slot assignment. The DF flexible time slot assignment allows the assignment of up to 128 D-channels to a single protocol handler.

2. DATA STRUCTURES

The data structures contain the following information:

- The bit maps are considered as slave data when point-to-point-back (PTPB) checks pass between rIPTSB and the link maps.
- The DF idle nibble bus time slot count are considered slave data to the bit map.
- The DF nibble bus time slot summaries are treated as slave data to the DF nibble bus time slot count.
- Audit control mechanisms, which schedule related audits to run immediately after an audit has found an error, is utilized to maintain consistency across all of the data structures.

3. AUDIT CHECKS

A function does all of the error detection and error recovery in-line.

- The current configuration of the PSUCOM determines how to interpret the data found in the arrays.
- Conditional time breaks are taken by calling AUCTB() after looping on every eight nibble bus time slots.
- The error detection parameters are range checked after every time break.
- The DT bits are read after every time break because of the possibility of a reconfiguration during the time break. Consequently the data structures could have one interpretation before the time break and a different interpretation after the time break.
- A count of the number of nibble bus time slots found idle a packet data bus (PDB) time slot will be kept. The data structures are updated if they do not agree with the count taken while traversing through all eight nibble bus time slots.
- The PTPB check between rIPTSB and the link is performed in the following manner. The rIPTSB tuple is derived by taking the physical key found in the link and calling the macro DAGETPTSB(key) to get a pointer to the rIPTSB tuple. From the physical PH circuit name the shelf can be derived by using the macro SMGSLFPPH(). Now all of the information in the rIPTSB has been translated into a form that will be compared with the error detection parameters.

4. ERROR HANDLING

All error detection and recovery is built into the semantic check function. What follows is the discussion on the actions which are taken for the different values found in the new data. Table 1 provides a summary of the following paragraphs.

Table 1 Audit Recovery Actions

PCdfnblink	PCdfnibble	rIPTSB PTPB ^a	Error Check	Recovery Action Found ^b
null	idle	not done	no	
null	busy	not done	yes	idle
key	idle	success	yes	busy
key	idle	failure	yes	null
key	busy	success	no	
key	busy	failure	yes	idle
				null
mcte ^c	idle	not done	yes	busy
mcte ^c	busy	not done	no	
Notes: a. point-to-point-back b. Audit reports an error and schedules PTSB and DFMAP audits when an error is found. c. Special value is used to indicate diagnostics is performing maintenance.				

If the PSUCOM is found to be in duplex failure, the DFNIBBLE audit expects to find the functions set properly. If any entry in any of these arrays is incorrect, it is changed to the expected value.

When the PSUCOM is in the normal mode of operation (active/standby), the audit takes the following actions based upon what is found in the link.

null = The corresponding function entry is expected to be idle and is recovered to this value when in error.

The idle count is incremented by one.

key = A PTPB check with rIPTSB is performed as described in the previous section. If the check passes, a function is busy and is recovered to this value if it is not. If the PTPB check fails, the appropriate functions are restored to their correct values. The idle count variable is incremented by one if the PTPB check fails.

maintenance flag = The special value PCDMYPTSB is placed into the link indicating that diagnostics is performing maintenance. At this time no rIPTSB tuple is allocated. Also, it is expected that a particular function is marked busy and the audit enforces this.

After every eight nibble bus time slots are looped through, the functions are checked against the idle count variable and updated to reflect what is calculated with the idle count variable.

Note that some of the recovery actions described here result in inconsistencies in other data structures. These are resolved by the related audit mechanisms. When the DFNIBBLE audit finds an error, the PTSB, DFMAP, and PHDSLGP audits are scheduled to run in the elevated mode.

Table 2 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The maintenance status structure appears to be idle, but some of the data is not the correct for idle.
Possible Error Effect:	The data in PCmnbstat structure cannot be believed.
Corrective Action Taken:	The maintenance nibble bus status structure (PCmnbstat) is set to the correct idle values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCmnbstat.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	PCmnbstat structure.

Table 3 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	The count of idle nibble bus time slots (PCdfnbcnt) does not agree with what is in PCdfnibble.
Possible Error Effect:	PC might be distributing the nibble bus time slots unequally which increases the probability of blocking.
Corrective Action Taken:	The count is corrected to how many are marked busy in PCdfnibble.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCdfnbcnt entry.
<i>Bad Data</i> =	Value of the PCdfnbcnt entry.
<i>Good Data</i> =	Count of idle time slots in PCdfnibble.
<i>Logical Index</i> =	Two bytes in the form H'xyy where: xx = PSU shelf yy = PDB time slot.
<i>Dump</i> =	NA.

Table 4 Error Code: BADTSSUM

Error Code:	BADTSSUM
Error Description:	The summary of idle nibble bus time slots does not agree with the count of idle nibble bus time slots.
Possible Error Effect:	PC might be distributing the nibble bus time slots unequally which increases the probability of blocking.
Corrective Action Taken:	The summary arrays (PCdfnbidle and PCdf1nbidle) are made consistent with PCdfnbcnt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Count of idle time slots in PCdfnbcnt.
<i>Logical Index</i> =	Two bytes in the form H'xyy where: xx = PSU shelf. yy = PDB time slot.
<i>Dump 1</i> =	Summary of idle NB time slots (PCdfnbidle).
<i>Dump 2</i> =	Critical summary of idle NB time slots (PCdf1nbidle).

Table 5 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The PCdfnblink entry points to a rIPTSB tuple which does not have the proper timeslot use. The rIPTSB tuple is probably corrupted.
Possible Error Effect:	The data within the rIPTSB tuple cannot be used. There appears to be some internal inconsistencies within the rIPTSB tuple.
Corrective Action Taken:	The bad use is reported as an error and the PTSB audit is scheduled to run. The nibble bus time slot is idled in PCdfnibble and set to null in PCdfnblink.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple.
<i>Bad Data</i> =	Use field of the rIPTSB tuple.
<i>Good Data</i> =	The tsuse field in the rIPTSB tuple.
<i>Logical Index</i> =	Three bytes in the form H'xyyzz where:

	xx = PSU shelf.
	yy = PSU shelf.
	zz = Nibble bus time slot.
<i>Dump 1 =</i>	rIPTSB tuple.
<i>Dump 2 =</i>	PCdfnibble entry.
<i>Dump 3 =</i>	PCdfnblink entry.

Table 6 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The PCmnbstat, PCdflink, and PCdfnblink structures failed a point-to-point-back check. For the PDB and nibble bus time slots found in PCmnbstat, PCdflink and PCdfnblink should have had the dummy rIPTSB key which indicates maintenance.
Possible Error Effect:	One of the PDB and/or nibble bus time slots might be tied up needlessly.
Corrective Action Taken:	The PCmnbstat structure is idled. The remaining portion of the DFNIBBLE audit and the DFMAP audit will clean up PCdfnblink and PCdflink if necessary.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Four bytes in the form H'wwxxyzz where: ww = PSU shelf. xx = PSU shelf. yy = PSU shelf. zz = Nibble bus time slot.
<i>Dump 1 =</i>	PCmnbstat global structure.
<i>Dump 2 =</i>	PCdflink entry.
<i>Dump 3 =</i>	PCdfnblink entry.

Table 7 Error Code: DTREAD

Error Code:	DTREAD
Error Description:	The DT key DTACTPBS was being read to determine if the PSUCOM was in duplex failure. The DT read returned an invalid value.
Possible Error Effect:	The configuration of the PSUCOM cannot be determined. Thus the audit is terminated.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value returned from DT read.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	DTACTPBS.
<i>Dump =</i>	NA.

Table 8 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The time found in PCmnbstat is invalid.
Possible Error Effect:	The PCmnbstat structure contains useless data.
Corrective Action Taken:	The PCmnbstat structure is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	PCmnbstat structure.
<i>Dump 2</i> =	The current time.

Table 9 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The PCdfnblink entry is marked as being used for maintenance, but the global maintenance nibble bus status structure PCmnbstat, which keeps track of which nibble bus time slot is being used, does not agree.
Possible Error Effect:	One of the nibble bus time slots is being tied up needlessly.
Corrective Action Taken:	The PCdfnblink entry is set to null and the PCdfnibble time slot is set to idle (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCdfnblink entry.
<i>Bad Data</i> =	Bad PCdfnblink entry.
<i>Good Data</i> =	Null value for PCdfnblink.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	PCmnbstat structure.

Table 10 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	The PSUCOM is in duplex failure according to DT bits or the given PSU shelf is not operational. However, some of the bits in PCdfnibble are idle.
Possible Error Effect:	PC could hunt time slots on the DF nibble bus.
Corrective Action Taken:	All bits in PCdfnibble for the given shelf and PDB time slot are marked busy (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCdfnibble entry.
<i>Bad Data</i> =	Bad PCdfnibble entry.
<i>Good Data</i> =	Value for all time slots busy.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = First nibble bus time slot in error.
<i>Dump</i> =	NA.

Table 11 Error Code: MKNULL

Error Code:	MKNULL
Error Description:	The PSUCOM is in duplex failure according to DT bits or the given PSU shelf is not operational. However, one of the entries in PCdfnblink appears to have a rIPTSB key.
Possible Error Effect:	PC could hunt time slots on the DF nibble bus.
Corrective Action Taken:	The PCdfnblink entry for the given PSU shelf, PDB time slot, and nibble bus time slot is marked with null (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCdfnblink entry.
<i>Bad Data</i> =	Bad PCdfnblink entry.

<i>Good Data</i> =	Null value for PCdfnblink.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	NA.

Table 12 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIPTSB tuple and PCdfnblink entry pass the point-to-point-back check, but the time slot in PCdfnibble is marked idle.
Possible Error Effect:	PC might hunt the time slot even though it is still being used.
Corrective Action Taken:	The nibble bus time slot in PCdfnibble is busied (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCdfnibble entry.
<i>Bad Data</i> =	Value of PCdfnibble entry.
<i>Good Data</i> =	PCdfnibble with time slot busied.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump 1</i> =	PCdfnblink entry.
<i>Dump 2</i> =	rIPTSB tuple.

Table 13 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The nibble bus time slot has been marked busy, but no rIPTSB tuple has been set up in PCdfnblink.
Possible Error Effect:	One of the nibble bus time slots is being tied up needlessly.
Corrective Action Taken:	The nibble bus time slot is idled both in software (PCdfnibble) and in hardware (with a disconnect in the DF hardware)
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of busy PCdfnibble entry.
<i>Bad Data</i> =	Value of busy PCdfnibble entry.
<i>Good Data</i> =	PCdfnibble entry with time slot idled.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	PCdfnblink.

Table 14 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The audit was not able to read the rISMEST relation for one side of the PSUCOM CC's.
Possible Error Effect:	The audit cannot determine the configuration of the PSUCOM. Consequently

	the PC global data cannot be audited.
Corrective Action Taken:	The CKTDATA audit is scheduled from this audit to have the rISMEST tuple populated from the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name of PSUCOM.
<i>Dump =</i>	NA.

Table 15 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	One of the nibble bus time slots has been held for maintenance for over one hour.
Possible Error Effect:	One of the nibble bus time slots is being tied up needlessly.
Corrective Action Taken:	The nibble bus and PDB time slots are released. PCmnbstat is idled, PCdfnlink and PCdfnblink are set to null, and PCdfnibble is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	PCmnbstat structure.
<i>Dump 2 =</i>	The current time.

Table 16 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	The nibble bus time slot is being used for maintenance, but PCdfnibble has it marked as being idle.
Possible Error Effect:	PC might try to use the nibble bus time slot when it is already being used for maintenance.
Corrective Action Taken:	The nibble bus time slot is marked busy in PCdfnibble.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad PCdfnibble entry.
<i>Bad Data =</i>	Value of bad PCdfnibble entry.
<i>Good Data =</i>	PCdfnibble entry with time slot busied.
<i>Logical Index =</i>	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump =</i>	NA.

Table 17 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The rIPTSB tuple and the PCdfnblink entries do not pass the point-to-point-back check. The PSU shelf, PDB time slot, and nibble bus time slot found in the rIPTSB tuple does not match the indices of PCdfnblink.
Possible Error Effect:	A nibble bus time slot might be tied up needlessly.
Corrective Action Taken:	The time slot is idled. PCdfnblink is set to null (0) and PCdfnibble is idled (0) for the nibble bus time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCdfnblink entry.
<i>Bad Data =</i>	Value of the PCdfnblink entry.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Three bytes in the form H'xyyzz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	riPTSB tuple.

Table 18 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or ASSERTS.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

DFNIBBLE-B

Software Release: 5E12 and later

1. DATA FANOUT NIBBLE AUDIT (DFNIBBLE)

The data fanout nibble (DFNIBBLE) audit monitors data fanout (DF) flexible time slot assignment. The DF flexible time slot assignment allows the assignment of up to 128 D-channels to a single protocol handler.

2. DATA STRUCTURES

The data structures contain the following information:

- The bit maps are considered as slave data when point-to-point-back (PTPB) checks pass between rIPTSB and the link maps.
- The DF idle nibble bus time slot count are considered slave data to the bit map.
- The DF nibble bus time slot summaries are treated as slave data to the DF nibble bus time slot count.
- Audit control mechanisms, which schedule related audits to run immediately after an audit has found an error, is utilized to maintain consistency across all of the data structures.

3. AUDIT CHECKS

A function does all of the error detection and error recovery in-line.

- The current configuration of the PSUCOM determines how to interpret the data found in the arrays.
- Conditional time breaks are taken by calling AUCTB() after looping on every eight nibble bus time slots.
- The error detection parameters are range checked after every time break.
- The DT bits are read after every time break because of the possibility of a reconfiguration during the time break. Consequently the data structures could have one interpretation before the time break and a different interpretation after the time break.
- A count of the number of nibble bus time slots found idle a packet data bus (PDB) time slot will be kept. The data structures are updated if they do not agree with the count taken while traversing through all eight nibble bus time slots.
- The PTPB check between rIPTSB and the link is performed in the following manner. The rIPTSB tuple is derived by taking the physical key found in the link and calling the macro DAGETPTSB(key) to get a pointer to the rIPTSB tuple. From the physical PH circuit name the shelf can be derived by using the macro SMGSLFPPH(). Now all of the information in the rIPTSB has been translated into a form that will be compared with the error detection parameters.

4. ERROR HANDLING

All error detection and recovery is built into the semantic check function. What follows is the discussion on the actions which are taken for the different values found in the new data. Table 1 provides a summary of the following paragraphs.

Table 1 Audit Recovery Actions

PCdfnblink	PCdfnibble	rIPTSB PTPB ^a	Error Check	Recovery Action Found ^b
null	idle	not done	no	
null	busy	not done	yes	idle
key	idle	success	yes	busy
key	idle	failure	yes	null
key	busy	success	no	
key	busy	failure	yes	idle
				null
mcte ^c	idle	not done	yes	busy
mcte ^c	busy	not done	no	
Notes: a. point-to-point-back b. Audit reports an error and schedules PTSB and DFMAP audits when an error is found. c. Special value is used to indicate diagnostics is performing maintenance.				

If the PSUCOM is found to be in duplex failure, the DFNIBBLE audit expects to find the functions set properly. If any entry in any of these arrays is incorrect, it is changed to the expected value.

When the PSUCOM is in the normal mode of operation (active/standby), the audit takes the following actions based upon what is found in the link.

null = The corresponding function entry is expected to be idle and is recovered to this value when in error.

The idle count is incremented by one.

key = A PTPB check with rIPTSB is performed as described in the previous section. If the check passes, a function is busy and is recovered to this value if it is not. If the PTPB check fails, the appropriate functions are restored to their correct values. The idle count variable is incremented by one if the PTPB check fails.

maintenance flag = The special value PCDMYPTSB is placed into the link indicating that diagnostics is performing maintenance. At this time no rIPTSB tuple is allocated. Also, it is expected that a particular function is marked busy and the audit enforces this.

After every eight nibble bus time slots are looped through, the functions are checked against the idle count variable and updated to reflect what is calculated with the idle count variable.

Note that some of the recovery actions described here result in inconsistencies in other data structures. These are resolved by the related audit mechanisms. When the DFNIBBLE audit finds an error, the PTSB, DFMAP, and PHDSLGP audits are scheduled to run in the elevated mode.

Table 2 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The maintenance status structure appears to be idle, but some of the data is not the correct for idle.
Possible Error Effect:	The data in PCmnbstat structure cannot be believed.
Corrective Action Taken:	The maintenance nibble bus status structure (PCmnbstat) is set to the correct idle values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCmnbstat.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	PCmnbstat structure.

Table 3 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	The count of idle nibble bus time slots (PCdfnbcnt) does not agree with what is in PCdfnibble.
Possible Error Effect:	PC might be distributing the nibble bus time slots unequally which increases the probability of blocking.
Corrective Action Taken:	The count is corrected to how many are marked busy in PCdfnibble.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCdfnbcnt entry.
<i>Bad Data</i> =	Value of the PCdfnbcnt entry.
<i>Good Data</i> =	Count of idle time slots in PCdfnibble.
<i>Logical Index</i> =	Two bytes in the form H'xyy where: xx = PSU shelf yy = PDB time slot.
<i>Dump</i> =	NA.

Table 4 Error Code: BADTSSUM

Error Code:	BADTSSUM
Error Description:	The summary of idle nibble bus time slots does not agree with the count of idle nibble bus time slots.
Possible Error Effect:	PC might be distributing the nibble bus time slots unequally which increases the probability of blocking.
Corrective Action Taken:	The summary arrays (PCdfnbidle and PCdf1nbidle) are made consistent with PCdfnbcnt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Count of idle time slots in PCdfnbcnt.
<i>Logical Index</i> =	Two bytes in the form H'xyy where: xx = PSU shelf. yy = PDB time slot.
<i>Dump 1</i> =	Summary of idle NB time slots (PCdfnbidle).
<i>Dump 2</i> =	Critical summary of idle NB time slots (PCdf1nbidle).

Table 5 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The PCdfnblink entry points to a rIPTSB tuple which does not have the proper timeslot use. The rIPTSB tuple is probably corrupted.
Possible Error Effect:	The data within the rIPTSB tuple cannot be used. There appears to be some internal inconsistencies within the rIPTSB tuple.
Corrective Action Taken:	The bad use is reported as an error and the PTSB audit is scheduled to run. The nibble bus time slot is idled in PCdfnibble and set to null in PCdfnblink.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple.
<i>Bad Data</i> =	Use field of the rIPTSB tuple.
<i>Good Data</i> =	The tsuse field in the rIPTSB tuple.
<i>Logical Index</i> =	Three bytes in the form H'xyyzz where:

	xx = PSU shelf.
	yy = PSU shelf.
	zz = Nibble bus time slot.
<i>Dump 1 =</i>	rIPTSB tuple.
<i>Dump 2 =</i>	PCdfnibble entry.
<i>Dump 3 =</i>	PCdfnblink entry.

Table 6 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The PCmnbstat, PCdflink, and PCdfnblink structures failed a point-to-point-back check. For the PDB and nibble bus time slots found in PCmnbstat, PCdflink and PCdfnblink should have had the dummy rIPTSB key which indicates maintenance.
Possible Error Effect:	One of the PDB and/or nibble bus time slots might be tied up needlessly.
Corrective Action Taken:	The PCmnbstat structure is idled. The remaining portion of the DFNIBBLE audit and the DFMAP audit will clean up PCdfnblink and PCdflink if necessary.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Four bytes in the form H'wwxxyzz where: ww = PSU shelf. xx = PSU shelf. yy = PSU shelf. zz = Nibble bus time slot.
<i>Dump 1 =</i>	PCmnbstat global structure.
<i>Dump 2 =</i>	PCdflink entry.
<i>Dump 3 =</i>	PCdfnblink entry.

Table 7 Error Code: DTREAD

Error Code:	DTREAD
Error Description:	The DT key DTACTPBS was being read to determine if the PSUCOM was in duplex failure. The DT read returned an invalid value.
Possible Error Effect:	The configuration of the PSUCOM cannot be determined. Thus the audit is terminated.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value returned from DT read.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	DTACTPBS.
<i>Dump =</i>	NA.

Table 8 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The time found in PCmnbstat is invalid.
Possible Error Effect:	The PCmnbstat structure contains useless data.
Corrective Action Taken:	The PCmnbstat structure is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	PCmnbstat structure.
<i>Dump 2</i> =	The current time.

Table 9 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The PCdfnblink entry is marked as being used for maintenance, but the global maintenance nibble bus status structure PCmnbstat, which keeps track of which nibble bus time slot is being used, does not agree.
Possible Error Effect:	One of the nibble bus time slots is being tied up needlessly.
Corrective Action Taken:	The PCdfnblink entry is set to null and the PCdfnibble time slot is set to idle (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCdfnblink entry.
<i>Bad Data</i> =	Bad PCdfnblink entry.
<i>Good Data</i> =	Null value for PCdfnblink.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	PCmnbstat structure.

Table 10 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	The PSUCOM is in duplex failure according to DT bits or the given PSU shelf is not operational. However, some of the bits in PCdfnibble are idle.
Possible Error Effect:	PC could hunt time slots on the DF nibble bus.
Corrective Action Taken:	All bits in PCdfnibble for the given shelf and PDB time slot are marked busy (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCdfnibble entry.
<i>Bad Data</i> =	Bad PCdfnibble entry.
<i>Good Data</i> =	The value of the local traffic variable.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = First nibble bus time slot in error.
<i>Dump</i> =	NA.

Table 11 Error Code: MKNULL

Error Code:	MKNULL
Error Description:	The PSUCOM is in duplex failure according to DT bits or the given PSU shelf is not operational. However, one of the entries in PCdfnblink appears to have a rIPTSB key.
Possible Error Effect:	PC could hunt time slots on the DF nibble bus.
Corrective Action Taken:	The PCdfnblink entry for the given PSU shelf, PDB time slot, and nibble bus time slot is marked with null (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCdfnblink entry.
<i>Bad Data</i> =	Bad PCdfnblink entry.

<i>Good Data</i> =	The value of the local traffic variable.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	NA.

Table 12 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIPTSB tuple and PCdfnblink entry pass the point-to-point-back check, but the time slot in PCdfnibble is marked idle.
Possible Error Effect:	PC might hunt the time slot even though it is still being used.
Corrective Action Taken:	The nibble bus time slot in PCdfnibble is busied (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCdfnibble entry.
<i>Bad Data</i> =	Value of PCdfnibble entry.
<i>Good Data</i> =	PCdfnibble with time slot busied.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump 1</i> =	PCdfnblink entry.
<i>Dump 2</i> =	rIPTSB tuple.

Table 13 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The nibble bus time slot has been marked busy, but no rIPTSB tuple has been set up in PCdfnblink.
Possible Error Effect:	One of the nibble bus time slots is being tied up needlessly.
Corrective Action Taken:	The nibble bus time slot is idled both in software (PCdfnibble) and in hardware (with a disconnect in the DF hardware)
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of busy PCdfnibble entry.
<i>Bad Data</i> =	Value of busy PCdfnibble entry.
<i>Good Data</i> =	PCdfnibble entry with time slot idled.
<i>Logical Index</i> =	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	PCdfnblink.

Table 14 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The audit was not able to read the rISMEST relation for one side of the PSUCOM CC's.
Possible Error Effect:	The audit cannot determine the configuration of the PSUCOM. Consequently

	the PC global data cannot be audited.
Corrective Action Taken:	The CKTDATA audit is scheduled from this audit to have the rISMEST tuple populated from the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name of PSUCOM.
<i>Dump =</i>	NA.

Table 15 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	One of the nibble bus time slots has been held for maintenance for over one hour.
Possible Error Effect:	One of the nibble bus time slots is being tied up needlessly.
Corrective Action Taken:	The nibble bus and PDB time slots are released. PCmnbstat is idled, PCdfnlink and PCdfnblink are set to null, and PCdfnibble is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	PCmnbstat structure.
<i>Dump 2 =</i>	The current time.

Table 16 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	The nibble bus time slot is being used for maintenance, but PCdfnibble has it marked as being idle.
Possible Error Effect:	PC might try to use the nibble bus time slot when it is already being used for maintenance.
Corrective Action Taken:	The nibble bus time slot is marked busy in PCdfnibble.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad PCdfnibble entry.
<i>Bad Data =</i>	Value of bad PCdfnibble entry.
<i>Good Data =</i>	PCdfnibble entry with time slot busied.
<i>Logical Index =</i>	Three bytes in the form H'xyyz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump =</i>	NA.

Table 17 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The rIPTSB tuple and the PCdfnblink entries do not pass the point-to-point-back check. The PSU shelf, PDB time slot, and nibble bus time slot found in the rIPTSB tuple does not match the indices of PCdfnblink.
Possible Error Effect:	A nibble bus time slot might be tied up needlessly.
Corrective Action Taken:	The time slot is idled. PCdfnblink is set to null (0) and PCdfnibble is idled (0) for the nibble bus time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCdfnblink entry.
<i>Bad Data =</i>	Value of the PCdfnblink entry.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Three bytes in the form H'xyzz where: xx = PSU shelf. yy = PDB time slot. zz = Nibble bus time slot.
<i>Dump</i> =	riPTSB tuple.

Table 18 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or ASSERTS.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

14. DI - DZ

DISPDB

Software Release: 5E10 and later

1. DISPLAY DATA BLOCK (DISPDB) AUDIT

The display data block (DISPDB) audit schedules audits and handles additional recovery actions such as asserts, single process purges (SSPs) and escalations. The dynamic relation RLDISPDB is used by the display capability to store data before displaying it on customer premises equipment (CPE), in a forty alphanumeric character string. This relation has a one-to-one linkage to the rIBRCSDB relation. When a display feature is activated, a rDISPDB tuple is removed from the rIDBHEADC idle list and linked to a rIBRCSDB tuple and is then used to store display data. The DISPDB audit must verify the linkages associated with a display data block, for each IDSN call that has a display feature.

The rDISPDB tuples are only needed by terminals with display features. In the case when all allocated tuples have been used, the CPE display will always display a default message (INCOMING CALL or OUTGOING CALL), depending on the terminal process.

The DISPDB audit is part of the Feature Control audits subgroup. It runs in the initialization and segmented mode and operates in the OSDS-M environment.

2. DATA STRUCTURES

The DISPDB audit fields consist of the key, display data, a member linkage to the RLBRSCDB relation, and a linkage to the rIDBHEADC idle list. The display data can be associated with a call or can be data for the number of calls on queue. When used to store call data, the RLDISPDB is associated with the originating or terminating terminal process; when used to store queue data, it is associated with the display terminal process (DISPTP).

3. AUDIT CHECKS

The DISPDB audit calls the DRINASEK driver. This driver ensures that the tuple is either linked to a rIBRCSDB tuple or to an rIDBHEADC idle list, checks the key, and calls the semantic check function, which verifies the data stored in the tuple. The DISPDB audit consists of an entry, semantic, recovery and an idle function. All linkage and semantic checks must pass before the audit proceeds to the next tuple. If they do not, the data cannot be assumed to be valid and the tuple will be linked to the rIDBHEADC idle list.

4. ERROR HANDLING

If the logical and physical keys are not consistent the tuple is zapped and placed on the idle list. If a linkage is semantic error is found the recovery function checks the process associated with the RLDISPDB relation. If it is a DISPTP the tuple will be recovered to the idle list. If the process is on OTP or TTP the linkage to the rIBRCSDB will be zeroed and the rDISPDB recovered to the idle list. If all is well, the audit gets the next tuple.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	This tuple is not on the idle list, the owner linkage is zero but the member linkage is also zero.
Possible Error Effect:	The rDISPDB tuple has been garbaged, you cannot trust the data in this tuple.
Corrective Action Taken:	Report the error on the ROP and return to the driver to put the tuple on the idle list.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rDISPDB.
<i>Bad Data</i> =	Member linkage to rIBRCSDB.
<i>Good Data</i> =	Owner linkage.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rDISPDB tuple.

Table 2 Error Code: DISP

Error Code:	DISP
Error Description:	A rDISPDB with an invalid linkage has been found.
Possible Error Effect:	The one-to-one linkage to the BRCSDB tuple is not zero and the call reference field is zero. This means that this DISPDB tuple has an associated Display Terminal process, which means that it is not associated with a call. Data will not be display, the screen will be blank.
Corrective Action Taken:	The audit idles the rDISPDB tuple and zeros the linkage to the BRCSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rDISPDB.
<i>Bad Data</i> =	Non-zero value of linkage.
<i>Good Data</i> =	Call reference value in the rDALB.
<i>Logical Index</i> =	Key of rDISPDB.
<i>Dump</i> =	rDISPDB.

Table 3 Error Code: OWN_LINK

Error Code:	OWN_LINK
Error Description:	A rDISPDB is supposed to be idle but is not linked to the rDISPDB idle list.
Possible Error Effect:	The rDISPDB is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rDISPDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rDISPDB.
<i>Bad Data</i> =	Owner link of rDISPDB.
<i>Good Data</i> =	Physical key of idle head cell.
<i>Logical Index</i> =	Key of rDISPDB.
<i>Dump</i> =	NA.

Table 4 Error Code: RELPTR

Error Code:	RELPTR
Error Description:	This pointer to the rDALB tuple is invalid.
Possible Error Effect:	The correct rDALB tuple will not be gotten, you cannot trust the data in this tuple.
Corrective Action Taken:	Report the error on the ROP and get the next tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rDISPDB.
<i>Bad Data</i> =	One-to-one linkage to rIBRCSDB.
<i>Good Data</i> =	Key to rDALB.
<i>Logical Index</i> =	Key to rDISPDB.
<i>Dump</i> =	rDISPDB tuple.

Table 5 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	This pointer to the rIBRCSDB tuple is invalid.
Possible Error Effect:	The correct rIBRCSDB tuple will not be gotten, you cannot trust the data in this tuple.
Corrective Action Taken:	Report the error on the ROP and get the next tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rDISPDB.
<i>Bad Data</i> =	One-to-one linkage to rIBRCSDB.
<i>Good Data</i> =	Key to rIBRCSDB.
<i>Logical Index</i> =	Key to rDISPDB.

<i>Dump</i> =	rDISPDB tuple.

Table 6 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rDISPDB is in the idle state but some of its linkages not involved in the idle list mapping are nonzero.
Possible Error Effect:	Some resource is linked to a rDISPDB on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rDISPDB tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rDISPDB.
<i>Bad Data</i> =	Non-zero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rDISPDB.
<i>Dump</i> =	rDISPDB.

DLI

Software Release: 5E10 and later

1. DUAL LINK INTERFACE (DLI) AUDIT

The dual link interface circuits (DLIs) reside in the switching modules but they are considered parts of the office network and timing complex (ONTC). Each local switching module contains two DLIs. The DLIs are connected to the time-multiplexed switch (TMS) in the communication module (CM) by means of fiber optic cables. The fiber-optic cable link between the DLIs and the TMS are the time-multiplexed switch links (TMSLNKs). Figure 1 shows these circuits. Status information about these circuits is kept in the peripheral configuration database (PCD). Several audits check the PCD, including the communication link audit (CLNK), the dual link interface audit (DLI), the PCD audit, and the time-multiplexed switch link audit (TMSLNK).

The DLI audit is part of the switch maintenance audits subgroup. It runs in the SMKP environment in the segmented mode.

2. DATA STRUCTURES

The DLI audit checks the SMddli array in the peripheral configuration database. The status of the DLIs is kept in the array. The audit also checks the data delivery data that is set up for each DLI. These data structures are depicted in Figure 2 .

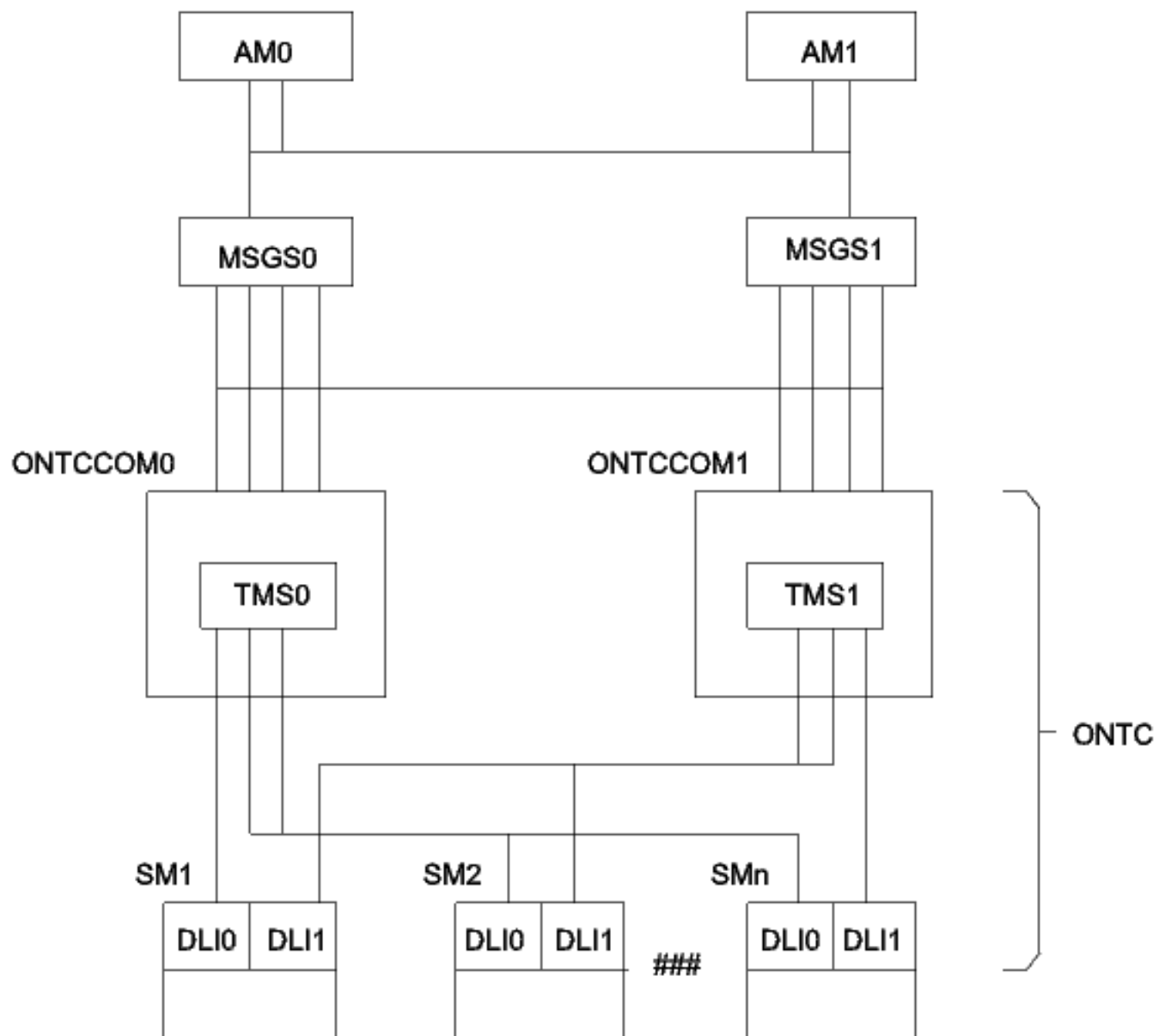


Figure 1 (DLI-1) Dual Link Interface Circuits

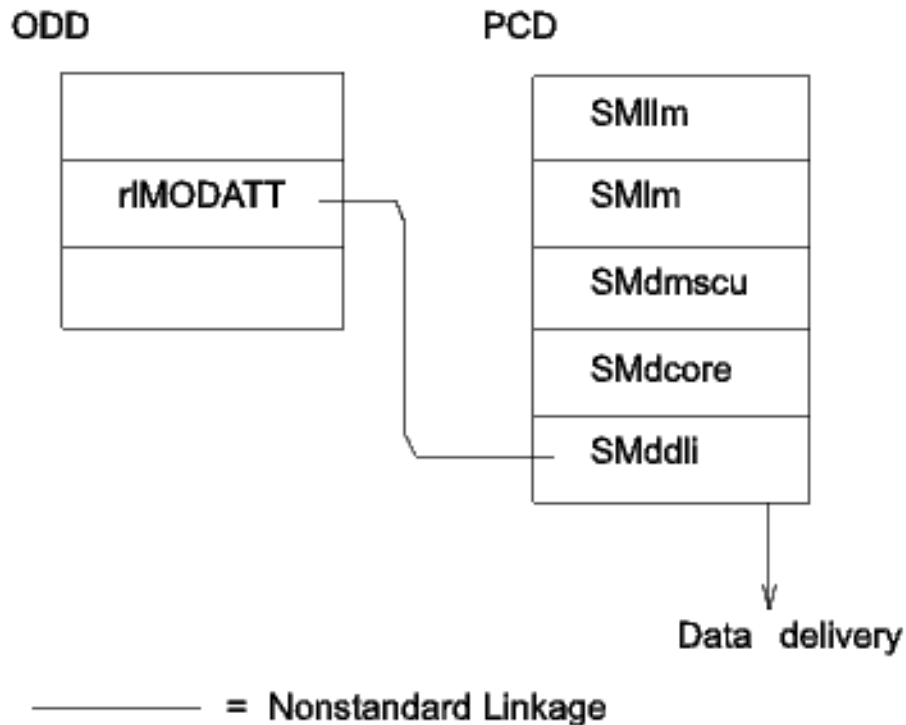


Figure 2 (DLI-2) Data Structures of the DLI Audit

3. AUDIT CHECKS

The audit makes software checks, vertical checks, external checks, and deferred maintenance administrator monitor (DMAM) checks. .AL a .LI The audit checks for the internal consistency within the dynamic peripheral configuration data base for the DLI and checks the consistency between the office dependent data (ODD) and the PCD. The DLI must be in an allowable state. Allowable states include active (ACT) if the SM is operational, out-of-service unequipped (OOS UNEQ) if the SM is unequipped or not a local SM, and out-of-service manual remove (OOS MAN RMV) if the SM is in a special growth state (SGRO). .LI The vertical check ensures the consistency of the DLI state in the PCD with the state of its controlling hardware unit, the office network and timing complex common (ONTCCOM). .LI The audit makes external checks on the SMddli array and special data such as data delivery bits. For out-of-service units in transient states, the audit ensures that some activity is scheduled as pending on DMAM's queue. Before taking corrective action, the audit ensures that the DLI is not in a transient state by checking the switch maintenance and recovery software (SMARS) queue. The SMARS queue contains a record of all maintenance activities. If fault recovery is in progress, the audit is rescheduled. If fault recovery is not in progress, an error is reported.

4. ERROR HANDLING

The type of error determines the actions the audit takes.

If a software internal check or a vertical check fails, the audit places the DLI in a valid state.

If the external check fails, the audit assumes that the DLI state is correct and updates any corresponding data to agree with the state of the DLI.

If the DMAM check fails, the DLI is in a stuck state. The audit places it in a valid state and requests a conditional restoration of the DLI.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not

found.

The audit produces the following error code(s).

Table 1 Error Code: CSADLI

Error Code:	CSADLI
Error Description:	The dual link interface (DLI) is in an active state but both time-multiplexed switch links (TMSLNKs) are out of service trouble.
Possible Error Effect:	The DLI is active when it should not be.
Corrective Action Taken:	The audit puts the DLI in an out of service state and sends a conditional restore for the DLI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the DLI which was found in error.
<i>Logical Index =</i>	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 2 Error Code: DDSVCDGRD

Error Code:	DDSVCDGRD
Error Description:	The service degrade data for this dual link interface (DLI) was found in error.
Possible Error Effect:	The service degrade state for this DLI and switching module (SM) is not correct.
Corrective Action Taken:	The audit corrects the service degrade data for the DLI and SM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the DLI which was found in error.
<i>Logical Index =</i>	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 3 Error Code: DLI_DD_CLI

Error Code:	DLI_DD_CLI
Error Description:	The vintage in data delivery data for the dual link interface (DLI) was found to be in error.
Possible Error Effect:	The DLI may be configured wrong.
Corrective Action Taken:	The audit updates the data delivery data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the DLI which was found in error.
<i>Logical Index =</i>	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 4 Error Code: DLI_DD_UNIQUE

Error Code:	DLI_DD_UNIQUE
Error Description:	The unique part of the data delivery data for the dual link interface (DLI) was found to be in error.
Possible Error Effect:	The DLI may be configured wrong.
Corrective Action Taken:	The audit updates the data delivery data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the DLI which was found in error.
<i>Logical Index =</i>	Office network and timing complex (ONTC) side the error was found on (that is,

	0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 5 Error Code: DLI_DD_UNIVERSAL

Error Code:	DLI_DD_UNIVERSAL
Error Description:	The universal part of the data delivery data for the dual link interface (DLI) was found to be in error.
Possible Error Effect:	The DLI may be configured wrong.
Corrective Action Taken:	The audit updates the data delivery data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the DLI which was found in error.
<i>Logical Index</i> =	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 6 Error Code: DLS_DMAM

Error Code:	DLS_DMAM
Error Description:	The dual link interface (DLI) is in a out of service state where there should be some deferred work being done on it but Deferred Maintenance Administrator and Monitor (DMAM) does not have an entry in its queue for this unit.
Possible Error Effect:	The DLI is out-of-service with no automatic action pending to restore it to service.
Corrective Action Taken:	Audit requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the DLI which was found in error.
<i>Logical Index</i> =	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 7 Error Code: DLI_FE

Error Code:	DLI_FE
Error Description:	The dual link interface (DLI) was in an out of service family state but the parent office network and timing complex (ONTC) is in service.
Possible Error Effect:	The DLI is out of service when it should not be.
Corrective Action Taken:	The audit restores the DLI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the DLI which was found in error.
<i>Logical Index</i> =	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 8 Error Code: DLI_PCD

Error Code:	DLI_PCD
Error Description:	A dual link interface (DLI) has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the DLI which was found in error.

<i>Logical Index</i> =	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 9 Error Code: INDEX

Error Code:	INDEX
Error Description:	The audit's cursor is out-of-range.
Possible Error Effect:	Audit could assert because it's trying to read information for a unit which does not exist.
Corrective Action Taken:	The cursor is reset to a valid value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Audit's cursor value found to be bad.
<i>Dump</i> =	NA.

Table 10 Error Code: MODATT

Error Code:	MODATT
Error Description:	The audit has found a problem with the rIMODATT relation in the office dependent data base (ODD).
Possible Error Effect:	A switching module (SM) may be having communication problems.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats check the rIMODATT relation tuple whose key is the SM number given in GOOD DATA.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	SM number used to read the rIMODATT relation.
<i>Logical Index</i> =	Number of the DLI which was found in error.
<i>Dump 1</i> =	Local buffer used in DBfrdtup(
<i>Dump 2</i> =	Audit's copy of the data read.

Table 11 Error Code: PCD

Error Code:	PCD
Error Description:	The audit has found an error which it cannot correct this may be due to a bad office dependent data (ODD), bad static data or the audit recovery table needs to be changed.
Possible Error Effect:	The audit will not recover this error.
Corrective Action Taken:	The audit will not recover this error.
Manual Action Required:	Check if the ODD is correct for the Peripheral Configuration Data (PCD). The included relations are: rISAUTISS For automatic time-slot switching. rIMODATT For the SMs.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the DLI which was found in error.
<i>Logical Index</i> =	Office network and timing complex (ONTC) side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 12 Error Code: SAITISS

Error Code:	SAITISS
Error Description:	The audit has found a problem with the rISAUTISS relation in the office dependent data base (ODD).

Possible Error Effect:	Automatic time-slot switching (AUTISS) may not be in the right state.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the rISAUTISS relation.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DBfrdtup return code.
<i>Good Data =</i>	SM number used to read the rIMODATT relation.
<i>Logical Index =</i>	Number of the DLI which was found in error.
<i>Dump 1 =</i>	Local buffer used in DBfrdtup()
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 13 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets, and or asserts.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	DLI audit cursor.
<i>Dump =</i>	NA.

DLTDATA

Software Release: 5E10 and later

1. DATA LINK TERMINAL DATA (DLTDATA) AUDIT

The DLTDATA audit is part of the Terminal Maintenance audits subgroup. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DLTDATA audit uses the terminal maintenance data link terminal data array (TMdlt_data), the integrated services inter-SM path relation (RLISMPATH), and the digital subscriber line equipage relation (RLDSLEQUIP).

3. AUDIT CHECKS

The audit checks the TMdlt_data array against the RLISMPATH and RLDSLEQUIP relations.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LNK_SM_NUM

Error Code:	LNK_SM_NUM
Error Description:	The mate SM number stored in TMdlt_data[] does not agree with the mate SM number found in the corresponding rLISMPATH tuple.
Possible Error Effect:	An inter-SM nailup could be out of order.
Corrective Action Taken:	The audit updates the TMdlt_data[] entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Mate SM number found in TMdlt_data[].
<i>Good Data =</i>	Mate SM number found in rLISMPATH.
<i>Logical Index =</i>	The index to the TMdlt_data[] array.
<i>Dump 1 =</i>	The TMdlt_data[] array element.
<i>Dump 2 =</i>	The rLDSLEQUIP tuple.
<i>Dump 3 =</i>	The rLISMPATH tuple.

Table 2 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The PH channel number stored in TMdlt_data[] does not agree with the DSL MEMBER found in the corresponding rLDSLEQUIP tuple.
Possible Error Effect:	An inter-SM nailup could be out of order.
Corrective Action Taken:	The audit updates the TMdlt_data[] entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PH channel number found in TMdlt_data[].
<i>Good Data =</i>	DSL MEMBER found in rLDSLEQUIP.
<i>Logical Index =</i>	The index to the TMdlt_data[] array.
<i>Dump 1 =</i>	The TMdlt_data[] array element.
<i>Dump 2 =</i>	The rLDSLEQUIP tuple.
<i>Dump 3 =</i>	The rLISMPATH tuple.

Table 3 Error Code: LOST_DSLDATA

Error Code:	LOST_DSLDATA

Error Description:	The audit has found a tuple in the rIISMPATH relation but the corresponding tuple in the rIDSLEQUIP relation is missing.
Possible Error Effect:	There is an ODD inconsistency between the two relations.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel must correct the ODD inconsistency between the RLISMPATH relation and the RLDSLEQUIP relation. The key to both relations is printed in <i>Good Data</i> .
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return value from DBfrdtup().
<i>Good Data</i> =	The key used to read the relations.
<i>Logical Index</i> =	The index to the TMdlt_data[] array.
<i>Dump 1</i> =	The TMdlt_data[] array element.
<i>Dump 2</i> =	The rIDSLEQUIP tuple.
<i>Dump 3</i> =	The rIISMPATH tuple.

Table 4 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The master SM indicator stored in TMdlt_data[] does not agree with the one found in the corresponding rIISMPATH tuple.
Possible Error Effect:	An inter-SM nailup could be out of order.
Corrective Action Taken:	The audit updates the TMdlt_data[] entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The master SM indicator found in TMdlt_data[].
<i>Good Data</i> =	The master SM indicator found in rIISMPATH.
<i>Logical Index</i> =	The index to the TMdlt_data[] array.
<i>Dump 1</i> =	The TMdlt_data[] array element.
<i>Dump 2</i> =	The rIDSLEQUIP tuple.
<i>Dump 3</i> =	The rIISMPATH tuple.

Table 5 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	The mate DLT number stored in TMdlt_data[] does not agree with the mate DLT number found in the corresponding rIISMPATH tuple.
Possible Error Effect:	An inter-SM nailup could be out of order.
Corrective Action Taken:	The audit updates the TMdlt_data[] entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Mate DLT number found in TMdlt_data[].
<i>Good Data</i> =	Mate DLT found in rIISMPATH.
<i>Logical Index</i> =	The index to the TMdlt_data[] array.
<i>Dump 1</i> =	The TMdlt_data[] array element.
<i>Dump 2</i> =	The rIDSLEQUIP tuple.
<i>Dump 3</i> =	The rIISMPATH tuple.

Table 6 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has gotten a bad return from DBfrdtup() when reading the rIISMPATH relation.
Possible Error Effect:	The ODD could be bad.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should check the RLISMPATH relation. The key to the relation is printed in <i>Good Data</i> .
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return value from DBfrdtup().
<i>Good Data</i> =	The key used to read the relations.
<i>Logical Index</i> =	The index to the TMdlt_data[] array.
<i>Dump 1</i> =	The TMdlt_data[] array element.
<i>Dump 2</i> =	The rIDSLEQUIP tuple.

<i>Dump 3 =</i>	The rISMPATH tuple.

Table 7 Error Code: PHDSLGP

Error Code:	PHDSLGP
Error Description:	The PH number stored in TMDlt_data[] does not agree with the DSL GROUP found in the corresponding rIDSLEQUIP tuple.
Possible Error Effect:	An inter-SM nailup could be out of order.
Corrective Action Taken:	The audit updates the TMDlt_data[] entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PH number found in TMDlt_data[].
<i>Good Data =</i>	DSL GROUP found in rIDSLEQUIP.
<i>Logical Index =</i>	The index to the TMDlt_data[] array.
<i>Dump 1 =</i>	The TMDlt_data[] array element.
<i>Dump 2 =</i>	The rIDSLEQUIP tuple.
<i>Dump 3 =</i>	The rISMPATH tuple.

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit has been entered by audit control to clean up after a reset.
Possible Error Effect:	Causes resets and or asserts.
Corrective Action Taken:	The audit is restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

DLTSBDB

Software Release: 5E10 and later

1. LINE TIME SLOT BRIDGING DATA BLOCK (DLTSBDB) AUDIT

The DLTSBDB audit verifies internal consistency of the RLDLTSBDB relation by performing linkages and semantic checks. The RLDLTSBDB relation contains per call information for a line assigned to have the bridging capability. The audit runs segmented in the operating system for distributed switching-switching module (OSDS-SM) environment, and during a selective init.

2. DATA STRUCTURES

The RLDLTSBDB relation is a dynamic, indexed, sequential relation. When idle, the rIDLTSBDB tuple is linked to an idle list owned by a tuple in the RLDB_HEADC relation. When it is busy, the rIDLTSBDB tuple is linked to an rCLDB tuple, which is also linked to an rPCBLA tuple. (Refer to Figure 1.)

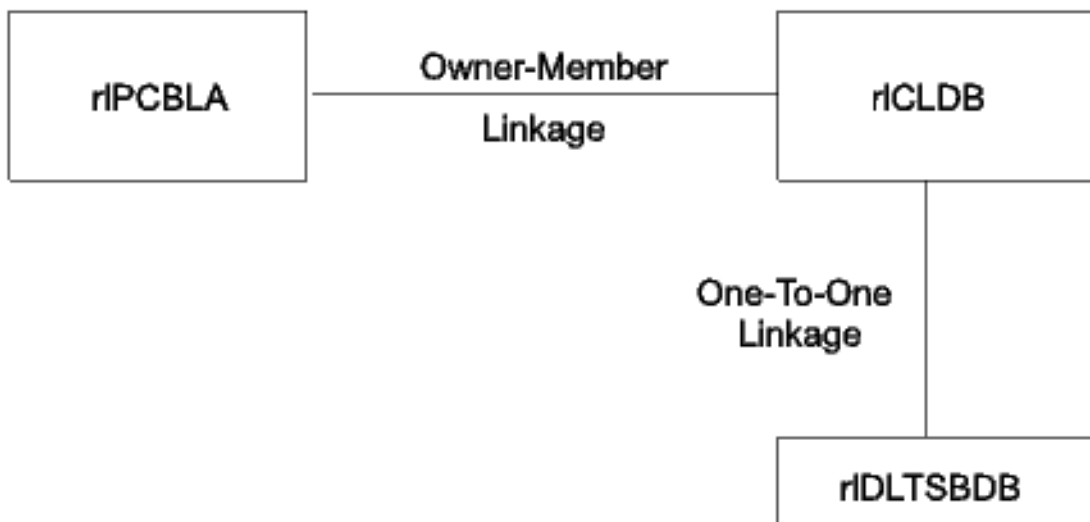


Figure 1 (DLTSBDB-1) Line Time Slot Bridging Data Block Busy Linkage

3. AUDIT CHECKS

The DLTSBDB audit entry function calls the audit driver when the audit is invoked. The driver checks key, quarantine state, and linkage of an rIDLTSBDB tuple. When all tests pass, the driver calls the audit semantic function to verify data consistency of that tuple. The following checks are performed if a specific attribute indicates that the current call is null:

- (a) The owner field of the rIDLTSBDB tuple should be the headcell.
- (b) The linkage zero check should return a zero, given the relation information.

The current call otherwise will not be null and the following checks are performed:

- (a) The rIDLTSBDB tuple linkage to the rCLDB tuple should not be zero.
- (b) The rCLDB tuple logical key should be equal to the linkage from the rIDLTSBDB tuple minus the rCLDB tuple key shift.
- (c) The rCLDB tuple linkage to the rPCBLA tuple should not be zero.

- (d) The flag indicating the rIPCBLA tuple linkage to the rCLDB tuple should be DBYES, and the rIPCBLA tuple logical key should be equal to the linkage from rCLDB tuple minus the rIPCBLA tuple key shift.

4. ERROR HANDLING

If an error occurs, the DLTSBDB audit prints an error message and returns a failure back to the driver. For recovery, the rDLTSBDB tuple is placed on the idle list in the audit recovery function. The DLTSBDB audit schedules the CLDB, PCBLA and DB_HEADC audits as related audits. The DLTSBDB audit produces these error codes.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	A dynamic tuple of line time slot bridging relation is linked to an incorrect list.
Possible Error Effect:	Lost resources vital to call processing.
Corrective Action Taken:	The dynamic tuple is recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDLTSBDB tuple.
<i>Bad Data =</i>	Owner field of rDLTSBDB tuple.
<i>Good Data =</i>	Physical key of correct idle list.
<i>Logical Index =</i>	Key of rDLTSBDB tuple.
<i>Dump =</i>	rDLTSBDB tuple.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	When LTSB process exists, there must be a linkage between rDLTSBDB tuple and rCLDB tuple. A check in the linkage field to rCLDB tuple indicates null.
Possible Error Effect:	Related audits will run and the corresponding DLTSBDB calls will be torn down.
Corrective Action Taken:	The dynamic LTSB tuple is zeroed and placed on the idle list. LTSB process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rDLTSBDB tuple.
<i>Bad Data =</i>	rDLTSBDB linkage field to rCLDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rDLTSBDB tuple.
<i>Dump =</i>	rDLTSBDB tuple.

Table 3 Error Code: BLINK

Error Code:	BLINK
Error Description:	When LTSB process exists, there must be a linkage between member rCLDB tuple and owner rIPCBLA tuple. However, the owner field of rCLDB indicates null.
Possible Error Effect:	Related audits will run and the corresponding DLTSBDB calls will be torn down.
Corrective Action Taken:	The dynamic LTSB tuple is zeroed and placed on the idle list. LTSB process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCLDB tuple.
<i>Bad Data =</i>	Owner field of rCLDB tuple.
<i>Good Data =</i>	rDLTSBDB linkage field to rCLDB.
<i>Logical Index =</i>	Key of rCLDB tuple.
<i>Dump =</i>	NA.

Table 4 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	When LTSB process exists, the ltsb_alwd bit in rIPCBLA tuple should set to one. All linkage checks passed but the bit indicates zero.
Possible Error Effect:	Related audits will run and the corresponding DLTSBDB calls will be torn down.
Corrective Action Taken:	The dynamic LTSB tuple is zeroed and placed on the idle list. LTSB process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDLTSBDB tuple.
<i>Bad Data =</i>	rIPCBLA bit.
<i>Good Data =</i>	Correct value of rIPCBLA bit.
<i>Logical Index =</i>	Key of rDLTSBDB tuple.
<i>Dump =</i>	rDLTSBDB tuple.

Table 5 Error Code: FBKDPTR

Error Code:	FBKDPTR
Error Description:	When LTSB process exists, owner rIPCBLA tuple should link to member rCLDB tuple and also the pcbld_alwd bit should set to one. Either the bit is in error, or invalid pointer is returned which results in key error.
Possible Error Effect:	Related audits will run and the corresponding DLTSBDB calls will be torn down.
Corrective Action Taken:	The dynamic LTSB tuple is zeroed and placed on the idle list. LTSB process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDLTSBDB tuple.
<i>Bad Data =</i>	cldb_flag bit in rIPCBLA tuple.
<i>Good Data =</i>	Owner logical key of rCLDB tuple.
<i>Logical Index =</i>	Key of rDLTSBDB tuple.
<i>Dump 1 =</i>	rDLTSBDB tuple.
<i>Dump 2 =</i>	rIPCBLA tuple.

Table 6 Error Code: FFWDPTR

Error Code:	FFWDPTR
Error Description:	Invalid key in rCLDB tuple.
Possible Error Effect:	Related audits will run and the corresponding DLTSBDB calls will be torn down.
Corrective Action Taken:	The dynamic LTSB tuple is zeroed and placed on the idle list. LTSB process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDLTSBDB tuple.
<i>Bad Data =</i>	Key of rCLDB tuple.
<i>Good Data =</i>	Key of rDLTSBDB tuple link.
<i>Logical Index =</i>	Key of rDLTSBDB tuple.
<i>Dump 1 =</i>	rDLTSBDB tuple.
<i>Dump 2 =</i>	Corresponding rCLDB tuple.

Table 7 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A dynamic tuple of the line time slot bridging relation is linked to an idle list, but its linkage fields are non-zero.
Possible Error Effect:	Lost resources vital to call processing.
Corrective Action Taken:	The dynamic tuple is zeroed and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rDLTSBDB tuple.
<i>Bad Data =</i>	Non-zero value of linkage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rDLTSBDB tuple.
<i>Dump =</i>	rDLTSBDB tuple.

DNGSDB

Software Release: 5E12 and later

1. DNU-S GROUP SCAN DATA BLOCK (DNGSDB) AUDIT

A group scan data block is linked to an active concentrated buffer headcell when a line origination is buffered. Each group scan data block represents a group of eight lines. Each line in a group is represented by one bit. If the bit is set, there is an origination on the line waiting to be processed. The DNU-S group scan data block audit (DNGSDB) ensures that group scan data block tuples (rIDNGSDBs) are properly linked to one of the concentrated buffer headcell tuples (rICBHC). The rICBHC tuples represent the idle list or one of several active lists. The audit also checks the name of the first port in the scan group that is stored in an rIDNGSDB tuple.

2. DATA STRUCTURES

For details on the fields contained in the RLDNGSDB relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	An rIDNGSDB tuple has an incorrect unit type stored in it.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit corrects the invalid unit type designation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDNGSDB tuple.
<i>Bad Data =</i>	rIDNGSDB tuple's unit type.
<i>Good Data =</i>	Derived unit type.
<i>Logical Index =</i>	Key of rIDNGSDB tuple.
<i>Dump 1 =</i>	rUNITDATA tuple.
<i>Dump 2 =</i>	rIDNGSDB tuple.

Table 2 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The first rIDNGSDB tuple should be empty (except for the key) but is not.
Possible Error Effect:	Garbage left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The tuple is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDNGSDB.
<i>Bad Data =</i>	Key of rIDNGSDB.
<i>Good Data =</i>	group name.
<i>Logical Index =</i>	Key of rIDNGSDB.
<i>Dump =</i>	rIDNGSDB tuple.

Table 3 Error Code: MODE

Error Code:	MODE
Error Description:	An rIDNGSDB tuple contains an incorrect shfmode value.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit reinitializes the shfmode attribute.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDNGSDB.
<i>Bad Data =</i>	Shfmode value.
<i>Good Data =</i>	Correct shfmode value.
<i>Logical Index =</i>	Key of rIDNGSDB.
<i>Dump 1 =</i>	rUNITDATA tuple.
<i>Dump 2 =</i>	rIDNGSDB tuple.

Table 4 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	A non-idle rIDNGSDB tuple is not linked to an rIGSLK tuple.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit recovers the rIDNGSDB tuple to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDNGSDB.
<i>Bad Data =</i>	gsdbm.st. link value.
<i>Good Data =</i>	DALNKNUL.
<i>Logical Index =</i>	Key of rIDNGSDB.
<i>Dump =</i>	rIDNGSDB tuple.

Table 5 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A non-idle rIDNGSDB tuple is not linked to valid owner.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit recovers the rIDNGSDB tuple to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDNGSDB.
<i>Bad Data =</i>	gsdbm.st. ownk value.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Key of rIDNGSDB.
<i>Dump =</i>	rIDNGSDB tuple.

Table 6 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	An rIDNGSDB tuple does not have a correct port value.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit recovers the invalid port attribute.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDNGSDB.
<i>Bad Data =</i>	Incorrect port value.
<i>Good Data =</i>	Correct port value.
<i>Logical Index =</i>	Key of rIDNGSDB.
<i>Dump 1 =</i>	rUNITDATA tuple.
<i>Dump 2 =</i>	rICKT_PORT tuple.
<i>Dump 3 =</i>	rIDNGSDB tuple.

Table 7 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An rIDNGSDB tuple is on the idle list, but contains other invalid linkages.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit recovers the tuple.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rIDNGSDB.
<i>Bad Data</i> =	Incorrect linkage value.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rIDNGSDB.
<i>Dump</i> =	rIDNGSDB tuple.

DOC

Software Release: 5E10 and later

1. DYNAMIC OVERLOAD CONTROL (DOC) AUDIT

The common channel signaling (CCS) system includes network management functions. The network management functions include dynamic overload control (DOC), provision of data on CCS controls, and reports of switching congestion conditions. The dynamic overload control relation (RLDOC) keeps track of dynamic overload controls. It is audited periodically to ensure its accuracy so that controls can be applied to the affected trunk groups. The DOC audit is part of the Routing and Terminal Allocation audits subgroup. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The DOC audit uses the dynamic overload control relation (RLDOC), the group to trunk group controls relation (RLGRP_TGC), and the database headcells relation (RLDB_HEADC). These data structures are shown in Figure 1.

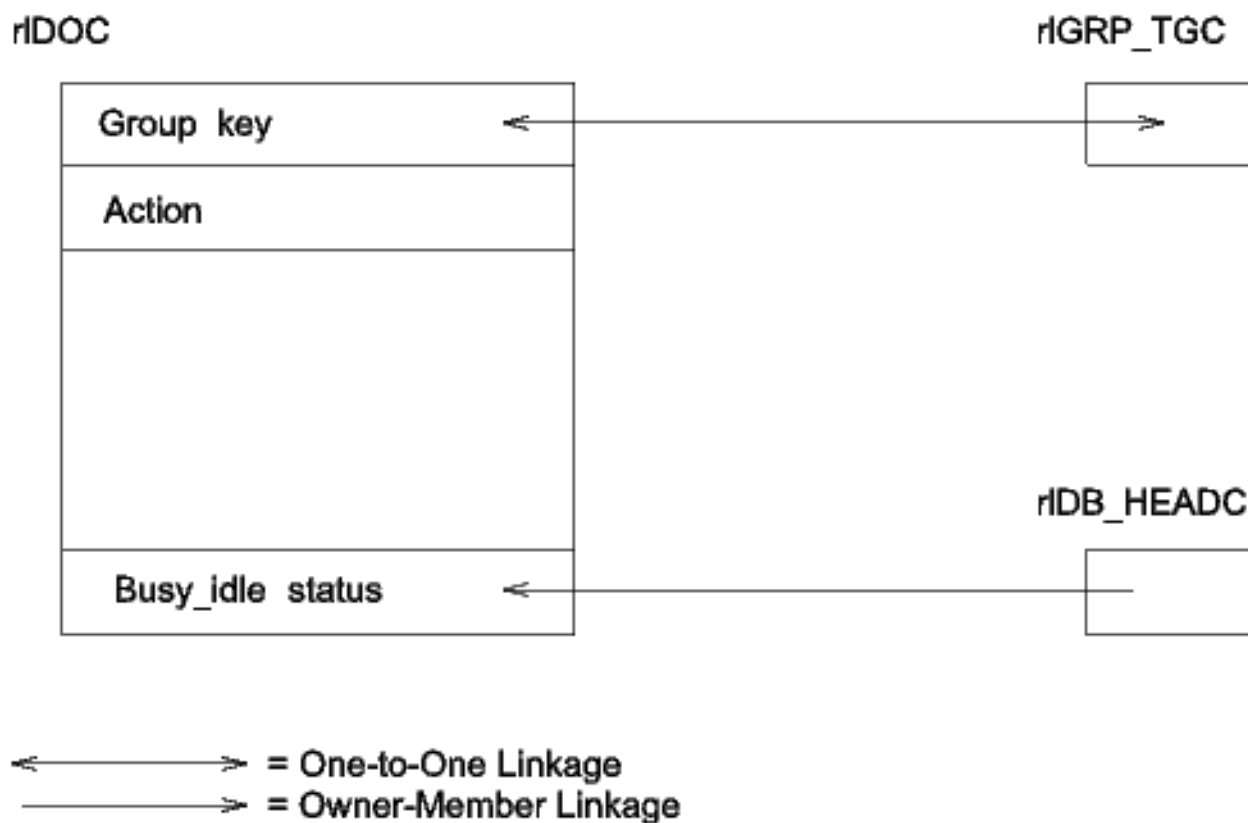


Figure 1 (DOC-1) Data Structures of the DOC Audit

3. AUDIT CHECKS

To verify the linkages in the RLDOC relation, the audit invokes the linkage driver (DRLINK) once for each audit segment. The linkage driver ensures that each tuple in RLDOC has a valid one-to-one linkage to the RLGRP_TGC relation. The linkage driver also verifies the linkage to the RLDB_HEADC relation. The linkage driver calls the semantic check function after verifying the linkage of each rlDOC tuple. The audit performs the following checks:

- (a) The audit verifies the all active rIDOC tuples have a non-null linkage back to the rIGRP_TGC relation.
- (b) The audit verifies the consistency between the control attribute and the control id field of rIGRP_TGC.
- (c) The audit verifies that the tuples are either busy or idle and that they are linked to the busy/idle lists in rIDB_HEADC.

If a control tuple is removed, the automatic control change (NMTG_AUTOCHG) 30-second discrete is set.

4. ERROR HANDLING

The audit reports errors when any check fails. It zeroes the linkage areas of the erroneous tuples and links them to the appropriate idle list.

The DOC audit schedules the GRPTGC and DBHEADC audits as related audits when it finds errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A tuple with valid linkages and data to qualify as busy is not on the busy list.
Possible Error Effect:	If this tuple is on the idle list, the data could be lost when the supposedly idle tuple is made active.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDOC tuple.
<i>Bad Data =</i>	Contents of st_ownc field.
<i>Good Data =</i>	Busy list headcell.
<i>Logical Index =</i>	Key to rIDOC tuple.
<i>Dump =</i>	Contents of rIDOC tuple.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A tuple that has no link into rIGRP_TGC is not fully idled.
Possible Error Effect:	When the tuple is made active again, some fields may be carried over.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDOC tuple.
<i>Bad Data =</i>	Contents of st_ownc field.
<i>Good Data =</i>	Idle list headcell.
<i>Logical Index =</i>	Key of rIDOC tuple.
<i>Dump =</i>	Contents of rIDOC tuple.

Table 3 Error Code: NULLCNTL

Error Code:	NULLCNTL
Error Description:	A tuple on the busy list has a non-active control type.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDOC tuple.
<i>Bad Data =</i>	Contents of action field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of tuple.

<i>Dump</i> =	Contents of rIDOC tuple.

DPB-A

Software Release: 5E10 - 5E11

1. D-CHANNEL PROTOCOL BLOCK (DPB) AUDIT

and the D-channel protocol block (DPB) are used to handle communications between the integrated services digital network (ISDN) periphery and the processes that control the ISDN ports. The DPB corresponds to the D-channel port, or the permanent packet of the channel, and has one DALB linked to it for each process that needs to communicate with the ISDN customer premises equipment (CPE). Most (but not all) of these processes control B-port served by the D-port.

For the integrated digital carrier unit (IDCU) the rIDPB tuple is used to represent the embedded operation channel (EOC) and the timeslot management channel (TMC) channels. A shared linkage is used to link these rIDPB tuples to the IDCU remote terminal linkage block (rIRTLB) instead of a rIDALB linkage and to the remote terminal identifier linkage area (rIRTEILA) instead of the rTEILA tuple.

In addition to the DALB linkages, the DPB contains pointers to the path through the switch network that connects the D-channel line card (T-card or U-card) or the inter-switch packet trunk to the protocol handler (PH), and the information required to control the handshaking and data transmission over the D-channel to the ISDN CPE.

2. DATA STRUCTURES

DPB is used for all ISDN applications which connect a port to a PH. Examples of PH ports are:

- Digital subscriber line (DSL) D-channel.
- DSL B-channel for packet switching.
- Interswitch packet trunk.
- Inter-switching module (SM) port.
- Data link terminal (DLT) port.
- Extended digital subscriber line (EDSL) D-channel.
- DSL B-channel for on-demand packet switching.
- Integrated digital carrier unit (IDCU) EOC/TMC channels.
- CDMA packet pipe trunk.
- CDMA speech handler (PHV) trunk.

The RLDPB, an indexed relation in the switching modules (SMs), contains the D-channel port name and peripheral interface data bus (PIDB) time slot block keys for the integrated services line unit (ISLU) and packet switching units (PSUs). The audit also uses the D-channel application linkage block relation (RLDALB), the PIDB time slot block relation (RLPTSB), the port type relation (RLPORTTYP), the digital subscriber line (SDL) equipage relation (RLDSLEQUIP), and the port linkage area relation (RLPORTLA).

For an ISDN or ISDN attendant D-channel DPB, services subscribed by the users (represented by TEDB) are initialized and linked to the DPB during full initialization.

When the CPE is plugged in and terminal end point identifier (TEI) is generated, the TEI linkage area (TEILA) is linked to the DPB.

This only applies to ISDN or ISDN ATTENDANT D-channels. The other channels, such as operator services position system (OSPS), CDMA, EDSL or PRI, have no TEDB or TEILA linked to the DPB. For non-OSPS PRI ISDN calls or CDMA packet pipe or speech handler trunks, no DALB is linked to the DPB. Please refer to the data diagrams for the DPB and related data structure. Refer to Figure 1 and Figure 2.

3. AUDIT CHECKS

- (1) The audit first verifies that the DPB belongs to a valid D-channel or packet B-channel.
- (2) The audit verifies that the key in DPB is correct.
- (3) The audit verifies that all linkages between the DALB and the DPB are correct.

This linkages will be null when the rDALB tuple is involved with primary rate interface (PRI) calls for a non-facility associated signaling (NFAS) and D-channel back up (DCBU) features excluding OSPS PRI calls.

- (4) The audit verifies that all linkages between the DPB and the following are correct:
 - Terminal equipment data block (TEDB).
 - Remote terminal linkage block (rRTLb).
 - Remote terminal endpoint identifier linkage area (rRTEILA).
 - TEI linkage area (TEILA).
 - Peripheral interface data bus time slot backs (PTSBs).
- (5) The audit verifies that data transmission rates and protocol data in the DPB are correct.
- (6) If channel is being monitored for protocol transmission, the audit verifies that the protocol monitor job tables are consistent each other.
- (7) The audit verifies that B-channels management data stored in the PDB is consistent with the corresponding B-channel status in PORTLA and B-channel configuration in rDSLEQUIP.
- (8) The audit verifies the path between the DPB and the PTSB is set per the routine port condition table.
- (9) For CDMA packet pipe trunks, the audit verifies that the number of PTSB's that should be associated with the packet pipe (according to the count in rIPP_MEM) are all there.
- (10) For ISDN or ISDN attendant lines, the D-channel DPB should link to the terminal equipment data block (TEDB). Other channels such as OSPS, extended digital subscriber lines (EDSLs) should not have linkages to TEDBs.
- (11) The audit verifies the number of TEILAs linked to DPB is the same as the specified in the DPB and is not greater than the maximum number configured in the rDSLEQUIP.
- (12) The audit verifies that the basic rate interface (BRI) class and binding data stored in the DPB is derived from the corresponding rDSLEQUIP tuple.
- (13) The audit provides existence checks on B-channel DPBs used for on demand B-channel package services (ODB). The audit also verifies the consistency of B-channel management data related to ODB.
- (14) The audit verifies the associated group information (rDPB.agi1_bchan and rDPB.agi2_bchan).

- (15) The audit verifies all the fields related to the NFAS/DCBU features for PRI calls, namely, interf_id, prigrp, pri_type and srwait.

4. ERROR HANDLING

If any errors are detected in a DPB tuple, the audit determines if a process currently owns the D-channel. If a process is found, the audit recovers the data by requesting a single process purge of the owning process.

If the DPB does not have a living process, the audit reinitializes the protocol and data rate information. The audit attempts to preserve the DALB, TEDB, and TEILA lists and the physical path through the PTSB's, since changes to either of these could result in lost data. The audit recovers these linkages if they do not pass the point to point back checks. (NON-ODB calls only. For ODB calls, the tuple and all linkages are idled.)

During recovery the rIDPB field is cleared to DBNO.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ACCPTR

Error Code:	ACCPTR
Error Description:	The ODB B-channel rIDPB tuple corresponding to a given rIORTLA tuple associated with the B-channel's port could not be accessed by the audit. For every rIORTLA tuple associated with an ODB B-channel port there should exist a corresponding B-channel rIDPB tuple where rIORTLA.port = rIDPB.port.
Possible Error Effect:	ODB calls cannot be set up on one of the B-channels for this DSL.
Corrective Action Taken:	The B-channel management data in the D-channel rIDPB tuple is idled out to indicate that no ODB calls are up on the B-channel (b1use or b2use marked to DMIDDLE_USE and b1pty_resb or b2pty_resv marked to 0 depending on which B-channel was using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple for the B-channel's port.
<i>Bad Data =</i>	The logical key of the D-channel rIDPB tuple (port).
<i>Good Data =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Logical Index =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Dump 1 =</i>	The rIORTLA tuple for the B-channel's port.
<i>Dump 2 =</i>	The rIDPB tuple for the D-channel.

Table 2 Error Code: ASSOC_PTSB

Error Code:	ASSOC_PTSB
Error Description:	For non-CDMA calls, the owner linkage field to the rIPTSb tuple dppto must be null as those PTSBs are linked through the semantic linkages, islu_ptsb and psiu_ptsb. The DPB audit found this field to be non null.
Possible Error Effect:	Possible failure of IDCU and other non-CDMA calls.
Corrective Action Taken:	The dppto linkage is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The st_headk field of the dppto field of the rIDPB tuple.
<i>Bad Data =</i>	The st_talk field of the dppto field of the rIDPB tuple.
<i>Good Data =</i>	The phctyp field of the rIDPB tuple.
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 3 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP

Error Description:	The corresponding rIODPSOP static tuple for a given ODB DSL and party could not be accessed. For every D-channel rIDPB tuple where the B-channel management data indicates ODB is in use on the DSL (rIDPB.b1use or b2use = DMOB_USE) there should exist a corresponding rIODPSOP static tuple where (key) rIDPB.port = (key) rIODPSOP.port and (key) rIODPSOP.party = (rIDPB.X >> rIODPSOP.party & (DMUCHAR)1) where X = b1pty_resv or b2pty_resv in the D-channel rIDPB tuple.
Possible Error Effect:	An unauthorized subscriber may be using the ODB feature.
Corrective Action Taken:	The B-channel port is quarantined (rIORTLA.quarstate set to AUAUSET) and the B-channel management data in the associated D-channel rIDPB tuple is recovered to an idle state (b1use or b2use marked DMIDLE_USE and b1pty_resv or b2pty_resv marked to 0 depending on which B-channel was using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple for the B-channel's port.
<i>Bad Data =</i>	The logical key of the D-channel rIDPB tuple (port).
<i>Good Data =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Logical Index =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Dump 1 =</i>	The rIORTLA tuple for the B-channel's port.
<i>Dump 2 =</i>	The rIDPB tuple for the D-channel.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	Either the blkcall, pktcampon or the srwait field of the rIDPB tuple is set but the D-channel port is not owned as indicated by the state of the rIORTLA tuple of this D port (DBPSIDLE) which is an error.
Possible Error Effect:	• If the blkcall field is set, then one can not originate or terminate to this DSL as the call is blocked. • If the pktcampon field is set, the D port indicates it is camped on when it is not. • If the srwait field is set, the D port will time out waiting for SERV and SERV ACK messages.
Corrective Action Taken:	Audit will force the recovery of the rIDPB tuple being audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The blkcall field of the rIDPB tuple.
<i>Bad Data =</i>	The pktcampon field of the rIDPB tuple.
<i>Good Data =</i>	The srwait field of the rIDPB tuple.
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump 1 =</i>	The rIDPB tuple being audited.
<i>Dump 2 =</i>	The rIORTLA tuple of the D-channel involved.

Table 5 Error Code: BADPTSB

Error Code:	BADPTSB
Error Description:	One of the PTSB's pointed to by the D-channel protocol block did not point back to the D-channel.
Possible Error Effect:	Potential for breaking the D-channel signaling path.
Corrective Action Taken:	PTSB pointers are cleared. Process owning DPB is purged if any. D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The key of the rIPTSB tuple that should have been linked to the rIDBP tuple under audit.
<i>Bad Data =</i>	PTSB number.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Port number of D-port.
<i>Dump 1</i> =	The rIDPB tuple.
<i>Dump 2</i> =	The rIPTSB tuple.

Table 6 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	B2-channel usage field is not consistent with DSLEQUIP.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B2-channel usage field is reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The error code sequence number: 1 = DMNO_SERV. 2 = DMPPB_USE. 3 = DMIDLE_USE.
<i>Good Data</i> =	B2-channel usage in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 7 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The pri_type field of the rIDPB tuple did not match the same field from the static rIPRIGRP tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	The pri_type field of the rIDPB will be corrected with the value found in the same field of the static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The prigrp field of the rIDPB tuple.
<i>Bad Data</i> =	The pri_type field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIPRIGRP tuple.
<i>Logical Index</i> =	The port field, key, of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	B-channel management data in DPB is not correct. Either B-channel usage is wrong or party reservation bits are not correctly set.
Possible Error Effect:	B-channel could be blocked for call processing.
Corrective Action Taken:	B-channel data are reinitialized correctly per DSLEQUIP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The error indicator. 1 = B-channel is configured as NO_SERV, but B-channel management data is not set to initialized value. 2 = B-channel is configured as X_25, but B-channel management data is not set to DMPB_USE or party reservations are not zero. 3 = B-channel is used for default service or operator use but party reservation is not zero or no TEDB is linked to DPB. 4 = B-channel is used for TM-USE but party reservation is not zero.

	5 = B-channel is used for CSV, CSD or ODB use, but party reservation is not set or no TEDB is linked to PDB. 6 = B-channel is used for CSV, CSD or ODB use, but party reservation bit is not consistent with PCBLA (process). 7 = If B-channel is owned by process, B-channel usage should not be IDLE_USE, VRSV-USE, PPB_USE or NO_SERV. 8 = If B-channel is not owned by process and usage field is indicated as IDLE-USE, then party reservation bit should not be set. 9 = When B-channel is not owned by process but used by shared DN (usage field is DMVRSV_USE), party reservation bits should not be zero and TEDB, DALB should link to DPB. 10 = The B-channel usage field is not defined in the system.
<i>Good Data</i> =	Valid values are: 1 = B1 channel has problem. 2 = B2 channel has problem.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	rIDPB tuple.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The value in rIPP_MEM.num_mem indicates that a certain number of rIPTSB tuples should be linked to this rIDPB when it is stable the number of tuples is wrong.
Possible Error Effect:	This packet pipe is not correctly set up for CDMA calls and will cause a loss of data.
Corrective Action Taken:	Audits will force recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the port (rIDPB.phctyp).
<i>Bad Data</i> =	The number of rIPTSB tuples that are currently linked (local cnt).
<i>Good Data</i> =	The number of rIPTSB tuples that should be linked (rIPP_MEM.num_mem).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump 1</i> =	The contents of the rIDPB tuple.
<i>Dump 2</i> =	The contents of the rIPP_MEM tuple.
<i>Dump 3</i> =	The contents of the rIFC_PORTTYP tuple.

Table 10 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The binding related data such as blport_set are not set correctly in DPB.
Possible Error Effect:	This DSL may have signaling and binding problems.
Corrective Action Taken:	blport_set in DPB is correctly set to be consistent with DSLEQUIP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	blport_set in DPB.
<i>Good Data</i> =	bind_lport in DSLEQUIP. (if bind_lport in DSLEQUIP is not zero, then blport_set in DPB should be set to DBYES).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 11 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The audible alert party bits should not be set when DSL is idle.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The audible alert party bits will be reset to zero.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The audible party in DPB.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The rIDPB key - port name.
<i>Dump =</i>	The rIDPB tuple.

Table 12 Error Code: BLINK

Error Code:	BLINK
Error Description:	There should be no linkage between a B-channel rIDPB tuple used for the ODB feature and TEDB, and the B-channel rIDPB tuple's tedb_flag should be DBNO. Either one or both of these conditions was found to be in error.
Possible Error Effect:	The B-channel rIDPB tuple, and related data are in an invalid state resulting in wasted resources.
Corrective Action Taken:	The B-channel rIDPB tuple, and related ODB call processing data are recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the B-DPB.
<i>Bad Data =</i>	The tedb link flag.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the B-DPB tuple.
<i>Dump =</i>	The B-channel rIDPB (which will contain the status of the linkage to the TEDB).

Table 13 Error Code: BOTHCHAN

Error Code:	BOTHCHAN
Error Description:	An associated group (rIDPB.agi1_bchan or rIDPB.agi2_bchan) is using both B-channels, when only one is allowed. Terminals identified in a group by the bit map rIDSLEQUIP.agipty# coincide with both B-channels identified in bit maps rIDPB.b1pty_resv and rIDPB.b2pty_resv.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	None.
Manual Action Required:	Check assignment of associated groups and multi-line hunt groups (MLHG) on RC/V Views 23.2 (DIGITAL SUBSCRIBER LINE) and 23.3 (ASSOCIATED GROUP). Check for overlapping groups, and MLHGs which occur in both groups.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	rIDPB.agi1_bchan.
<i>Good Data =</i>	rIDPB.agi2_bchan.
<i>Logical Index =</i>	Key of DPB (rIDPB.port).
<i>Dump 1 =</i>	rIDPB tuple.
<i>Dump 2 =</i>	rIDSLEQUIP tuple.

Table 14 Error Code: CHAN_ZERO

Error Code:	CHAN_ZERO
Error Description:	B-channel management data are not set to initialized value for non-ISDN line.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B-channel management data are reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDPB key - port name.
<i>Dump =</i>	The rIDPB tuple.

Table 15 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of TEILA linked to DPB is not equal to those specified in DPB (tot_terms field).
Possible Error Effect:	The error may cause binding and signaling problem.
Corrective Action Taken:	tot_terms field in DPB is reset to correct number.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	tot_terms in DPB.
<i>Good Data</i> =	The actual TEILA linked to DPB.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 16 Error Code: DBCHAN

Error Code:	DBCHAN
Error Description:	B1-channel usage field is not consistent with DSLEQUIP.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B1-channel usage field is reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The error code sequence number: 1 = Should be DMNO_SERV. 2 = Should be DMPPB_USE. 3 = Should be DMIDLE_USE.
<i>Good Data</i> =	B1-channel usage in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 17 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	An invalid linkage is found between a B-channel rIDPB tuple used for the ODB feature and an rIPTSB tuple.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	The ODB call will be torn down and related data recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The invalid linkage between the B-channel DPB and the PTSB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DPB key.
<i>Dump 1</i> =	The rIDPB tuple.
<i>Dump 2</i> =	The rIPTSB tuple.

Table 18 Error Code: DSL_RATE

Error Code:	DSL_RATE
Error Description:	Data rate in DPB differs from ODD data rate.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Process owning DPB is purged if any D-PORT is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data rate found in DPB.
<i>Good Data</i> =	Data rate according to ODD.
<i>Logical Index</i> =	Port number of D-port.

<i>Dump</i> =	NA.

Table 19 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The logical PH number and physical PH channel data in a B-channel rIDPB tuple used for ODB did not match the data contained in the corresponding rIODPSOP static tuple. For B-channel rIDPB tuples used for ODB calls, the ph_number and ph_cn attributes must be represented in rIODPSOP.ph_number and rIODPSOP.ph_cn or ph_cn2 where, for the associated D-channel, (key) rIDPB.port = (key) rIODPSOP.port and (key) rIDPSOP.party = (rIDPB.X >> rIODPSOP.party & (DMUCHAR)1) where X = b1pty_resv or b2pty_resv in the D-channel rIDPB tuple associated with the B-channel rIDPB tuple under audit.
Possible Error Effect:	An unauthorized subscriber may be using the ODB feature.
Corrective Action Taken:	The ODB call process is purged and the ODB call owning the B-channel is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-channel rIDPB tuple under audit.
<i>Bad Data</i> =	The PH number the rIDPB tuple (ph_number)
<i>Good Data</i> =	The PH number found in the rIODPSOP static tuple (ph_number).
<i>Logical Index</i> =	The key of the rIDPB tuple under audit (port).
<i>Dump 1</i> =	The B-channel rIDPB tuple under audit.
<i>Dump 2</i> =	The associated D-channel rIDPB tuple for the DSL.
<i>Dump 3</i> =	The associated rIODPSOP static tuple.

Table 20 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rIDPB dynamic tuple without a corresponding static tuple where: - the DPB does not have the static rIDSLEQUIP tuple with rIDPB.port = rIDSLEQUIP.dsl OR - there is no user subscribed to the on-demand-packet feature with rIDPB.port = rIODPSOP.port
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key value.
<i>Dump</i> =	The rIDPB tuple.

Table 21 Error Code: GPORT

Error Code:	GPORT
Error Description:	The specified B-channel port is not populated or key or B-channel PORTLA is wrong.
Possible Error Effect:	Application code may assert when accessing the data and B-channel cannot be used.
Corrective Action Taken:	PORTLA or FCPORTTYPE audit will be scheduled as related audit to detect this error and correct it.
Manual Action Required:	If FC_PORTTYPE is missing, then manual action is required to insert the B-channel FC PORTTYPE tuple. (Look good data in the dump.)
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.

<i>Bad Data</i> =	The key of B-PORTLA in the system (maybe 0xffff).
<i>Good Data</i> =	The key of B-PORTLA which should be in the system.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 22 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Based on the bit maps rIDSLEQUIP.agipty1 and rIDPB.b#pty_resv the associated group information rIDPB.agi1_bchan is set to the incorrect value. The range of values is FCNO_BCHAN, FCB1_CHAN, and FCB2_CHAN.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	rIDPB.agi1_bchan is set to the correct value listed below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan.
<i>Good Data</i> =	Correct value of rIDPB.agi1_bchan.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 23 Error Code: GRP2INFO

Error Code:	GRP2INFO
Error Description:	Based on the bit maps rIDSLEQUIP.agipty2 and rIDPB.b#pty_resv the associated group information rIDPB.agi2_bchan is set to the incorrect value. The range of values is FCNO_BCHAN, FCB1_CHAN, and FCB2_CHAN.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	rIDPB.agi2_bchan is set to the correct value listed below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi2_bchan.
<i>Good Data</i> =	Correct value of rIDPB.agi2_bchan.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 24 Error Code: IL_PTYE

Error Code:	IL_PTYE
Error Description:	The lp_type in rIFC_PORTTYP does not indicate that the port B-channel rIDPB is associated with a B-channel port.
Possible Error Effect:	Wasted resources since this tuple cannot be used for anything.
Corrective Action Taken:	The B-channel rIDPB tuple and related data, if any, will be recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The lp_type in rIFCPORTTYP.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the DPB tuple.
<i>Dump</i> =	The DPB tuple.

Table 25 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	EOC/TMC ports on an IDCU TR303 have a invalid link between the rIDPB and the rRTLb tuple.
Possible Error Effect:	Tasks for this RT may be lost and not completed.
Corrective Action Taken:	The EOC/TMC port will be recovered and the link restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The contents of the rIDPB tuple.

Table 26 Error Code: INV

Error Code:	INV
Error Description:	Either the interf_id or the prigrp field of the rIDPB tuple did not match the same field from the static rIDCHNPRI tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Both the interf_id and the prigrp fields of the rIDPB tuple will be corrected with the values found in the same fields of the static rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The interf_id field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDCHNPRI tuple.
<i>Good Data</i> =	The interf_id field of the rIDCHNPRI tuple.
<i>Logical Index</i> =	The prigrp field of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 27 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	BRI class of service in DPB is not consistent with DSLEQUIP.
Possible Error Effect:	This DSL may have signaling and binding problems.
Corrective Action Taken:	briclass in DPB is correctly set to be consistent with DSLEQUIP and the port is put on the to be idled list (TMLTTBI) to be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	briclass in DPB.
<i>Good Data</i> =	briclass in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 28 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The default service TEDB should link to the DSL which has subscribed default service feature but it links to other DSL.
Possible Error Effect:	No default service is provided in this DSL.
Corrective Action Taken:	The DSL will be recovered and all the TEDBs will be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The DSL which TEDB links to.
<i>Good Data</i> =	The DSL which TEDB should link to.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 29 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The audit received an unrecognized or invalid return from TMdpbprt().
Possible Error Effect:	The rIDPB tuple cannot be audited.
Corrective Action Taken:	The audit skips the rIDPB tuple since it cannot access any related data required for auditing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIFC_PORTTYP static tuple passed to TMdpbprt().
<i>Bad Data</i> =	The return received from TMdpbprt().
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The rIFC_PORTTYP static tuple.

Table 30 Error Code: L2MON

Error Code:	L2MON
Error Description:	The port is requested to be monitored for protocol but the data used is inconsistent.
Possible Error Effect:	Switch may waste resources to monitor the unrequested port.
Corrective Action Taken:	The port is marked as not being monitored.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The pmon field in DPB.
<i>Good Data</i> =	DBNO (0).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 31 Error Code: LEVEL_1

Error Code:	LEVEL_1
Error Description:	Level two profile in the ODD does not match profile in the DPB.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Process owning DPB is purged if any D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Level 2 profile found in DPB.
<i>Good Data</i> =	Level 2 profile according to ODD.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	NA.

Table 32 Error Code: LEVEL_3

Error Code:	LEVEL_3
Error Description:	The level three state in the DPB did not match the level three state derived from the circuit type information in the PORTTYP and DSLEQUIP tuple.
Possible Error Effect:	Potential for breaking the D-channel signaling path.
Corrective Action Taken:	Level three state is updated to match the static ODD data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Level three state from the DPB.
<i>Good Data</i> =	Level three state derived from the ODD.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	DPB is dumped.

Table 33 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	DSL should have default service feature but there is no default service TEDB linked to DPB or TEDB key is wrong.
Possible Error Effect:	No default service is provided in this DSL.
Corrective Action Taken:	Print the error and request related audits to insert TEDB or correct the TEDB errors.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The TEDB key in the system.
<i>Good Data</i> =	The TEDB key which should be.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 34 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The servclass and lp_type fields of the static rIFC_PORTTYPE tuple indicates that this D channel is not used for NFAS and DCBU PRI D-channel (AUportbuf.servclass not != FCEDSL and AUportbuf.lp_type != PCTDCHAN).

	And yet audit found that one of the three fields related to NFAS and DCBU D channel, namely, interf_id, prigrp, and pri_type has a non null value.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	All the three fields will be set to the null value (interf_id->PSNO_INTID, prigrp->TMNULLPRI and pri_type->TMNULL_TYPE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The interf_id field of the rIDPB tuple.
<i>Bad Data =</i>	The prigrp field of the rIDPB tuple.
<i>Good Data =</i>	The pri_type field of the rIDPB tuple.
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 35 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	For an NFAS/DCBU PRI calls, the linkage field to the rIDALB tuple dpdao must be null as no DALBs used in such type of PRI calls is linked to DPBs. Audit found this field to be non null.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Audit will force recovery of this rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The st_headk field of the dpdao field of the rIDPB tuple.
<i>Bad Data =</i>	The st_tailk field of the dpdao field of the rIDPB tuple.
<i>Good Data =</i>	The prigrp field of the rIDPB tuple.
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 36 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	While looking for a process on the port associated with the rIDPB tuple during selective initialization, the DA function call returned an error saying that database corruption exists. The audit error report dumps rIORTLA because that is what it has access to.
Possible Error Effect:	The port would normally be taken down unless a barge-in call pickup process exists for this port. Since database error was returned, the port will be taken down.
Corrective Action Taken:	The tuple is quarantined, so that the audit forces recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The pointer to the rIORTLA tuple.
<i>Bad Data =</i>	The value returned from the DA function call.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIORTLA tuple's key, port field.
<i>Dump =</i>	The rIORTLA tuple being audited.

Table 37 Error Code: NOTIDLE2

Error Code:	NOTIDLE2
Error Description:	A rIDPB tuple not currently used for ODB was found to have its logical PH number (ph_number) and physical channel number (ph_cn) attributes set to invalid values. For D-channel rIDPB tuples, or rIDPB B-channel tuples which are not associated with B-channels used for the ODB feature, the PH and channel attributes are unused and should be set to idle values (PSPHNULL and zero respectively).
Possible Error Effect:	Call processing will never be able to use the associated B-channel for ODB since the PH and channel attributes make it appear that the B-channel already has a PH and channel assigned and in use.
Corrective Action Taken:	The ph_number and ph_cn attributes are recovered to their idle values (PSPHNULL and 0 respectively).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The contents of the ph_number and ph_cn attributes. (rIDPB.ph_number << 4 rIDPB.ph_cn)
<i>Good Data</i> =	The null values for ph_number and ph_cn (PSPHNULL << 4 (DMPHCN) 0).
<i>Logical Index</i> =	The rIDPB tuple's key (port).
<i>Dump</i> =	The rIDPB tuple.

Table 38 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The PH channel type indicates that this is a packet pipe or speech handler trunk so the owner linkage to the PTSB should be non-zero. Instead, the semantic linkage (islu_ptsb or psiu_ptsb) to the PTSB is non-zero.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	Audits will force recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The islu_ptsb field of the rIDPB tuple.
<i>Bad Data</i> =	The psiu_ptsb field of the rIDPB tuple.
<i>Good Data</i> =	The phctyp field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 39 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The PORTTYP relation indicates that the DSL for this port must be equipped, but the DSLEQUIP tuple is missing from the static ODD. If the GOOD DATA is 2, the DPB audit was unable to obtain the static relation tuple (RLPP_MEM) needed to audit the dynamic rIDPB tuple for the CDMA packet pipe either there is no packet pipe member tuple with the same port as the DPB or PP_MEM key is wrong.
Possible Error Effect:	If the <i>Good Data</i> is 1, the audit cannot verify any data in the DPB. If it is bad, the audit has no way to fix it. If the <i>Good Data</i> is 2, packet pipe members cannot be located for packet pipe call processing.
Corrective Action Taken:	None.
Manual Action Required:	If the <i>Good Data</i> is 1, office personnel will need to use ODBE to put the missing DSLEQUIP tuple back into the ODD, since ODBE does not permit removing the PORTTYP tuple under these circumstances. If the <i>Good Data</i> is 2, office personnel will need to use ODBE to put the missing PP_MEM tuple into the ODD. Find out the value of the port.member from the dump of the rIDPB tuple. The port.module will be the processor receiving the audit. Using these as the key, insert the static rIPP_MEM tuple. For the other fields of the tuple, refer to the terminal maintenance and trigger area.
Dump Description:	
<i>Error Address</i> =	If the <i>Good Data</i> is 1, the address of the rDSLEQUIP tuple. If the <i>Good Data</i> is 2, the address of the rIDPB tuple.
<i>Bad Data</i> =	The logical port type (rIFC_PORTTYP.lp_type).
<i>Good Data</i> =	Valid values are:
	1 = Could not read the rDSLEQUIP tuple.
	2 = Could not read the rIPP_MEM tuple.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump 1</i> =	The contents of the rIDPB tuple.
<i>Dump 2</i> =	The contents of the rIFC_PORTTYP tuple.

Table 40 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	DPB tuples that are not used by IDCU or CDMA must be linked to TEDB, DALB, and TEILA tuples. This DPB tuple is not linked to one of them. If <i>Good Data</i> is 1, it is not linked to TEDB. If <i>Good Data</i> is 2, it is not linked to DALB. If <i>Good Data</i> is 3, it is not linked to TEILA.
Possible Error Effect:	Problem in call processing.
Corrective Action Taken:	The rIDPB tuple is quarantined so that the tuple is recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The linkage, either TEDB, DALB, or TEILA. (See under GOOD DATA.)
<i>Good Data</i> =	Valid values are: 1 = The bad linkage is to a TEDB. 2 = The bad linkage is to a DALB. 3 = The bad linkage is to a TEILA.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 41 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	The PH channel circuit type stored in an rIDPB tuple, which is associated with a B-channel port used for the ODB feature, was found inconsistent from the type derived from the B-channel management data stored in static ODD (rIFC_PORTTYPE).
Possible Error Effect:	The PORT could be OOS.
Corrective Action Taken:	The "phctype" field is updated in the DPB to agree with the B-channel management data stored in static ODD (rIFC_PORTTYP).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DPB tuple.
<i>Bad Data</i> =	Value of the "phctype" field in the DPB.
<i>Good Data</i> =	Value of "phctyp" computed from the ODD.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The DPB tuple.

Table 42 Error Code: PHDSLGP

Error Code:	PHDSLGP
Error Description:	There are currently two ODB calls up on this DSL. However, the logical PH and physical PH channel numbers stored in the rIDPB tuples for each B-channel are identical. When ODB calls are up on each B-channel of a DSL, the ph_number and ph_cn attributes in each ODB B-channel rIDPB tuple must be unique to each other. Since it is physically impossible to set up ODB calls on a DSL under these conditions, this error is an indication that the ph_number and ph_cn attributes in one or both rIDPB tuples became corrupted after both ODB calls were set up on the DSL. The occurrence of this error is an indication that the PH and channel data does not reflect the actual PH and channel(s) being used for ODB call activities.
Possible Error Effect:	The ODB connection(s) on this DSL cannot be cleanly torn down. Attempts by call processing to do so may result in asserts and/or audit errors.
Corrective Action Taken:	Both B-channel rIPORTLA tuples are quarantined for recovery (rIPORTLA.quarstate marked AUAUSET). The B-channel management data in the D-channel rIDPB tuple (b1use, b2use, b1pty_resv and b2pty_resv) are marked to idle states (DMIDDLE_USE and 0 respectively). Lastly, the logical PH number (ph_number) and physical channel number (ph_cn) attributes in the B1

	and B2 rIDPB tuples are marked to an idle state (PSPHNULL and 0 respectively). Any ODB call processes up on the DSL will then be purged and any other related call data will be idled out by related audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the B-channel rIDPB ODB B-channel tuple under audit.
<i>Bad Data =</i>	The logical PH number (ph_number) in the B-channel rIDPB ODB B-channel tuple under audit.
<i>Good Data =</i>	The physical PH channel number (ph_cn) in the B-channel rIDPB ODB B-channel tuple under audit.
<i>Logical Index =</i>	The key of the B-channel rIDPB ODB B-channel tuple (port).
<i>Dump 1 =</i>	The contents of the B-channel rIDPB ODB B-channel tuple under audit.
<i>Dump 2 =</i>	The contents of the "other" B-channel rIDPB ODB B-channel tuple.
<i>Dump 3 =</i>	The contents of the D-channel rIDPB tuple on the DSL.

Table 43 Error Code: PHNUNEQUIP

Error Code:	PHNUNEQUIP
Error Description:	The dsl_group and/or member obtain from the static tuple rIDSLEQUIP do not match the ph_number and/or ph_cn stored in the rIDPB tuple.
Possible Error Effect:	Problems in call processing.
Corrective Action Taken:	The rIDPB tuple is recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	PH number (rIDPB.ph_number).
<i>Good Data =</i>	DSL_group (rIDSLEQUIP.dsl_group).
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump 1 =</i>	The rIDPB tuple being audited.
<i>Dump 2 =</i>	The rIDSLEQUIP tuple being audited.

Table 44 Error Code: PLE_STATE

Error Code:	PLE_STATE
Error Description:	The plestate attribute is incorrect.
Possible Error Effect:	D-channel will be idled.
Corrective Action Taken:	The rIDPB tuple will be initialized and made idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPB.
<i>Bad Data =</i>	Incorrect plestate.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDPB.
<i>Dump =</i>	rIDPB.

Table 45 Error Code: PORT

Error Code:	PORT
Error Description:	Using port field of the rIDPB tuple as the member field and GI_pcrd as the module field of the dport key field of rIDCHNPRI tuple, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it is inserted by office personnel. Look at the manual action required field below as to how to insert the tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Both the interf_id and the prigrp fields of the rIDPB tuple will be reset to their null values and audit will retun.
Manual Action Required:	Using port field of the rIDPB tuple as the member field and GI_pcrd as the module field of the dport key field of rIDCHNPRI tuple, insert the static rIDCHNPRI tuple. As for the values of the other fields, refer to the terminal maintenance and triggers area.
Dump Description:	
<i>Error Address =</i>	The prigrp field of the rIDPB tuple.
<i>Bad Data =</i>	The interf_id field of the rIDPB tuple.
<i>Good Data =</i>	The pri_type field of the rIDPB tuple.

<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 46 Error Code: PROC_TO

Error Code:	PROC_TO
Error Description:	The process owning B-channel port does not link properly with the B-channel PORTLA.
Possible Error Effect:	Application code may assert when accessing the data and B-channel cannot be used.
Corrective Action Taken:	The PORTLA will be quarantined and process will be purged.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The process number of PCBLA.
<i>Good Data</i> =	The expected process number from PORTLA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 47 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The level 3 profile obtained from rFC_PORTTYP and rIODPSOP static ODD did not match the level 3 profile stored in a B-channel rIDPB tuple being used for the ODB feature.
Possible Error Effect:	The B-channel signaling path may be corrupted.
Corrective Action Taken:	The level 3 data will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-DPB tuple.
<i>Bad Data</i> =	The level 3 profile found in the B-DPB tuple.
<i>Good Data</i> =	The level 3 profile derived from static data.
<i>Logical Index</i> =	The key of the B-DPB tuple.
<i>Dump</i> =	The B-channel DPB tuple.

Table 48 Error Code: PTR3

Error Code:	PTR3
Error Description:	According to the D-channel rIDPB tuple for a DSL with two ODB B-channels assigned, both B-channels currently have ODB calls up, however, only one rIDPB B-channel tuple could be accessed by the audit. For each B-channel "use" field in the D-channel rIDPB tuple which is marked to indicate ODB is in use on the respective B-channel (b1use and/or b2use = DMODB_USE) there should exist a corresponding ODB B-channel rIDPB tuple.
Possible Error Effect:	If the B-channel is assigned the ODB feature, then ODB calls can never be originated on this B-channel.
Corrective Action Taken:	The B-channel management data in the associated D-channel rIDPB tuple which corresponds to the B-channel under audit will be recovered to an idle state (b1use or b2use set to DMIDDLE_USE and b1pty_resv or b2pty_resv set to 0 depending on which B-channel was marked as using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the D-channel rIDPB tuple for the DSL.
<i>Bad Data</i> =	Informational: The b1use field in the D-channel rIDPB tuple.
<i>Good Data</i> =	Informational: The b2use field in the D-channel rIDPB tuple.
<i>Logical Index</i> =	The key of the D-channel rIDPB tuple (port)
<i>Dump</i> =	The D-channel rIDPB tuple.

Table 49 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The routine port conditioning table indicates that a call path should be up (TMportcond[rIORTLA.list where rIORTLA.port = rIDPB.port].ispathup =

	TMPCYES) but the rIDPB tuple is not linked to an rIPTSB tuple (rIDPB.islu_ptsb = 0 and rIDPB.psiu_ptsb = 0) and no process owns the port (rIORTLA.proc_cplr.st_headk = 0 where rIORTLA.port = rIDPB.port). The rIDPB tuple should be linked to an rIPTSB tuple if the routine port conditioning table indicates a path should be up for a given port list except if the port is owned by a process.
Possible Error Effect:	The time slots associated with the rIPTSB tuple will be lost to the system, thus reducing the total number of available time slots.
Corrective Action Taken:	The rIDPB tuple is recovered to an idle state, and the rIORTLA tuple is placed on the to-be-idled list (rIORTLA.list = TMLTTBI), and the terminal maintenance unloader is awakened to process the rIORTLA tuple so that it is eventually set to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The list the port associated with the RLDPB is on (rIORTLA.list where rIORTLA.port = rIDPB.port).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key (rIDPB.port) of the rIDPB tuple.
<i>Dump 1 =</i>	The contents of the rIDPB tuple.
<i>Dump 2 =</i>	The contents of the associated rIORTLA tuple.

Table 50 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The port is not on a recovery list, so the quarantine state should not be in a race condition, but it is.
Possible Error Effect:	Lost resources since this port is out-of-service.
Corrective Action Taken:	The rIDPB tuple is quarantined so it can be recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The quarantine state (rIDPB.quarstat).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump =</i>	NA.

Table 51 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The number of TEILA linked to DPB is greater than what user subscribes (specify in DSLEQUIP).
Possible Error Effect:	The error may cause binding and signaling problem.
Corrective Action Taken:	The DSL will be reinitialized and recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	Number of TEILA linked to DPB.
<i>Good Data =</i>	Number of TEILA allowed in DSLEQUIP.
<i>Logical Index =</i>	The rIDPB key - port name.
<i>Dump =</i>	The rIDPB tuple.

Table 52 Error Code: SGNOOD

Error Code:	SGNOOD
Error Description:	Using prigrp field of the rIDPB tuple as the key field of rIPRIGRP tuple, audit tried to read the static rIPRIGRP tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it is inserted by office personnel. Look at the manual action required field below as to how to insert the tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	The pri_type field of the rIDPB tuple will be reset to its null value (TMNULL TYPE) and audit will retun.
Manual Action Required:	Find out the value of the prigrp field from the dump of the rIDPB tuple. Use it as

	the key field insert the static rIPRGRP tuple. As for the other fields of the tuple, refer to the terminal maintenance and trigger area.
Dump Description:	
<i>Error Address</i> =	The pri_type field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDPB tuple.
<i>Good Data</i> =	The interf_id field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 53 Error Code: TEDBFLAG

Error Code:	TEDBFLAG
Error Description:	The tedb_flag attribute in the DPB tuple is incorrectly set.
Possible Error Effect:	An incorrect linkage may exist between the DPB and either a TEDB tuple or a DALB tuple. This may effect call processing.
Corrective Action Taken:	The tedb_flag will be correctly set according to the servclass found at FC_PORTTYPE. If it is ISDN or ISDN attendant DSL, then tedb_flag will be set to DBYES. If it is an EOC/TMC port, then the tedb_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Valid values are: 0 = If the tedb_flag should be set to DBYES. 1 = If the tedb_flag should be set to DBNO.
<i>Good Data</i> =	Valid values are: 0 = If not ISDN and ISDN attendant or if EOC/TMC port. 1 = If it is ISDN or ISDN attendant.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	RLDPB.

Table 54 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	An invalid linkage is found between the rIDPB tuple and a rTEDB tuple.
Possible Error Effect:	Switch cannot communicate with customer premise equipment. EOC/TMC port on IDCUI TR303 have lost communication with the switch.
Corrective Action Taken:	Port will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Linkage value from rIDPB tuple to rTEDB (rIDPB.tedb_own.st_headk).
<i>Good Data</i> =	0. There should not be a linkage between the tuples.
<i>Logical Index</i> =	The rIDPB key.
<i>Dump</i> =	The contents of rIDPB tuple.

Table 55 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The fixed terminal endpoint identifier (FTEI) flag indicator in the rIDPB tuple is incorrect. It does not match the same indicator in the corresponding rIDSLEQUIP tuple. This flag is used by call processing to determine if there is a FTEI resource on this digital subscriber line (DSL).
Possible Error Effect:	FTEI resources on this DSL may not function correctly on this DSL.
Corrective Action Taken:	The flag indicator (pfitei) is recovered to the value specified in the corresponding rIDSLEQUIP flag.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.

<i>Bad Data</i> =	The value of pfixtei in the rIDPB before correction.
<i>Good Data</i> =	The value of pfixtei in the corresponding rIDSLEQUIP tuple.
<i>Logical Index</i> =	The rIDPB key.
<i>Dump 1</i> =	The contents of rIDPB tuple.
<i>Dump 2</i> =	The contents of rIDSLEQUIP tuple.

Table 56 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	For non-ISDN line. TEILA should not link to DPB.
Possible Error Effect:	Service cannot be received from this port.
Corrective Action Taken:	The port will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The port which TEILA links to.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 57 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	The tpwait field associated with this DPB is not set correctly according to data in the PORTLA.
Possible Error Effect:	"tpwait" should be on for EOC/TMC ports plus X.75, X.75\, and IP trunks when ISPA terminal process (TP) is waiting for a level 2 setup message and off for all others.
Corrective Action Taken:	The tpwait field will be updated according to the information present in the PORTLA if this is the only error detected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the dpb tuple.
<i>Bad Data</i> =	Value of "tpwait" field in the DPB.
<i>Good Data</i> =	"tpwait" value determined from PORTLA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	The DPB tuple.
<i>Dump 2</i> =	The PORTLA tuple.

Table 58 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	A broken linkage is found when traversing the rTEILA tuple through this rIDPB tuple. A broken linkage to rRTEILA tuple is found on an IDCU TR303 EOC/TMC port.
Possible Error Effect:	Application codes may assert and some of CPEs may not receive any service.
Corrective Action Taken:	The DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 59 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PH channel type stored in the DPB does not agree with that which was computed from the ODD.
Possible Error Effect:	The PORT could be OOS.
Corrective Action Taken:	The "phctyp" field is updated in the DPB to agree with the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DPB tuple.

<i>Bad Data</i> =	Value of the "phctyp" field in the DPB.
<i>Good Data</i> =	Value of "phctyp" computer from the ODD.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The rIDPB tuple.

Table 60 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	rIDPB.agi1_bchan and rIDPB.agi2_bchan are equal. This means that both groups are using the same B-channel, which is not allowed.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	None.
Manual Action Required:	Check assignment of associated groups and MLHGs on RC/V Views 23.2 (DIGITAL SUBSCRIBER LINE) and 23.3 (ASSOCIATED GROUP). Check for overlapping groups, and MLHGs which occur in both groups.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan
<i>Good Data</i> =	rIDPB.agi2_bchan
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 61 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	All rIDPB tuples for EOC/TMC ports on an IDCUC TR303 should have a linkage to an rIRTEILA tuple.
Possible Error Effect:	Message flow between the SM and IDCUC RT303 is interrupted.
Corrective Action Taken:	EOC/TMC port is recovered and communication restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The contents of the rIDPB tuple.

DPB-B

Software Release: 5E12 only

1. D-CHANNEL PROTOCOL BLOCK (DPB) AUDIT

The D-channel application linkage block (DALB) and the D-channel protocol block (DPB) are used to handle communications between the integrated services digital network (ISDN) periphery and the processes that control the ISDN ports. The DPB corresponds to the D-channel port, or the permanent packet of the channel, and has one DALB linked to it for each process that needs to communicate with the ISDN customer premises equipment (CPE). Most (but not all) of these processes control B-port served by the D-port.

For the integrated digital carrier unit (IDCU) the rIDPB tuple is used to represent the embedded operation channel (EOC) and the timeslot management channel (TMC) channels. A shared linkage is used to link these rIDPB tuples to the IDCU remote terminal linkage block (rIRTLB) instead of a rIDALB linkage and to the remote terminal identifier linkage area (rIRTEILA) instead of the rTEILA tuple.

In addition to the DALB linkages, the DPB contains pointers to the path through the switch network that connects the D-channel line card (T-card or U-card) or the inter-switch packet trunk to the protocol handler (PH), and the information required to control the handshaking and data transmission over the D-channel to the ISDN CPE.

2. DATA STRUCTURES

DPB is used for all ISDN applications which connect a port to a PH. Examples of PH ports are:

- Digital subscriber line (DSL) D-channel.
- DSL B-channel for packet switching.
- Interswitch packet trunk.
- Inter-switching module (SM) port.
- Data link terminal (DLT) port.
- Extended digital subscriber line (EDSL) D-channel.
- DSL B-channel for on-demand packet switching.
- Integrated digital carrier unit (IDCU) EOC/TMC channels.
- CDMA packet pipe trunk.
- CDMA speech handler (PHV) trunk.

The RLDPB, an indexed relation in the switching modules (SMs), contains the D-channel port name and peripheral interface data bus (PIDB) time slot block keys for the integrated services line unit (ISLU) and packet switching units (PSUs). The audit also uses the D-channel application linkage block relation (RLDALB), the PIDB time slot block relation (RLPTSB), the port type relation (RLPORTTYP), the digital subscriber line (SDL) equipage relation (RLDSLEQUIP), and the port linkage area relation (RLPORTLA).

For an ISDN or ISDN attendant D-channel DPB, services subscribed by the users (represented by TEDB) are initialized and linked to the DPB during full initialization.

When the CPE is plugged in and terminal end point identifier (TEI) is generated, the TEI linkage area (TEILA) is linked to the DPB.

This only applies to ISDN or ISDN ATTENDANT D-channels. The other channels, such as operator services position system (OSPS), CDMA, EDSL or PRI, have no TEDB or TEILA linked to the DPB. For non-OSPS PRI ISDN calls or CDMA packet pipe or speech handler trunks, no DALB is linked to the DPB. Please refer to the data diagrams for the DPB and related data structure. Refer to Figure 1 and Figure 2.

3. AUDIT CHECKS

- (a) The audit first verifies that the DPB belongs to a valid D-channel or packet B-channel.
- (b) The audit verifies that the key in DPB is correct.
- (c) The audit verifies that all linkages between the DALB and the DPB are correct. This linkage will be null when the rDALB tuple is involved with primary rate interface (PRI) calls for a non-facility associated signaling (NFAS) and D-channel back up (DCBU) features excluding OSPS PRI calls.
- (d) The audit verifies that all linkages between the DPB and the following are correct:
 - Terminal equipment data block (TEDB).
 - Remote terminal linkage block (rIRTLB).
 - Remote terminal endpoint identifier linkage area (rIRTEILA).
 - TEI linkage area (TEILA).
 - Peripheral interface data bus time slot backs (PTSBs).
- (e) The audit verifies that data transmission rates and protocol data in the DPB are correct.
- (f) If channel is being monitored for protocol transmission, the audit verifies that the protocol monitor job tables are consistent each other.
- (g) The audit verifies that B-channels management data stored in the PDB is consistent with the corresponding B-channel status in PORTLA and B-channel configuration in rIDSLEQUIP.
- (h) The audit verifies the path between the DPB and the PTSB is set per the routine port condition table.
- (i) For CDMA packet pipe trunks, the audit verifies that the number of PTSB's that should be associated with the packet pipe (according to the count in rIPP_MEM) are all there.
- (j) For ISDN or ISDN attendant lines, the D-channel DPB should link to the terminal equipment data block (TEDB). Other channels such as OSPS, extended digital subscriber lines (EDSLs) should not have linkages to TEDBs.
- (k) The audit verifies the number of TEILAs linked to DPB is the same as the specified in the DPB and is not greater than the maximum number configured in the rIDSLEQUIP.
- (l) The audit verifies that the basic rate interface (BRI) class and binding data stored in the DPB is derived from the corresponding rIDSLEQUIP tuple.
- (m) The audit provides existence checks on B-channel DPBs used for on demand B-channel package services (ODB). The audit also verifies the consistency of B-channel management data related to ODB.
- (n) The audit verifies the associated group information (rIDPB.agi1_bchan and rIDPB.agi2_bchan).
- (o) The audit verifies all the fields related to the NFAS/DCBU features for PRI calls, namely, interf_id, prigrp,

pri_type and srwait.

4. ERROR HANDLING

If any errors are detected in a DPB tuple, the audit determines if a process currently owns the D-channel. If a process is found, the audit recovers the data by requesting a single process purge of the owning process.

If the DPB does not have a living process, the audit reinitializes the protocol and data rate information. The audit attempts to preserve the DALB, TEDB, and TEILA lists and the physical path through the PTSB's, since changes to either of these could result in lost data. The audit recovers these linkages if they do not pass the point to point back checks. (NON-ODB calls only. For ODB calls, the tuple and all linkages are idled.)

During recovery the rIDPB field is cleared to DBNO.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ACCPTR

Error Code:	ACCPTR
Error Description:	The ODB B-channel rIDPB tuple corresponding to a given rIORTLA tuple associated with the B-channel's port could not be accessed by the audit. For every rIORTLA tuple associated with an ODB B-channel port there should exist a corresponding B-channel rIDPB tuple where rIORTLA.port = rIDPB.port.
Possible Error Effect:	ODB calls cannot be set up on one of the B-channels for this DSL.
Corrective Action Taken:	The B-channel management data in the D-channel rIDPB tuple is idled out to indicate that no ODB calls are up on the B-channel (b1use or b2use marked to DMIDDLE_USE and b1pty_resb or b2pty_resv marked to 0 depending on which B-channel was using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple for the B-channel's port.
<i>Bad Data =</i>	The logical key of the D-channel rIDPB tuple (port).
<i>Good Data =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Logical Index =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Dump 1 =</i>	The rIORTLA tuple for the B-channel's port.
<i>Dump 2 =</i>	The rIDPB tuple for the D-channel.

Table 2 Error Code: ASSOC_PTSB

Error Code:	ASSOC_PTSB
Error Description:	For ADT and non-CDMA calls, the owner linkage field to the rIPTSB tuple dppto must be null, as non-CDMA PTSBs are linked through the semantic linkages, islu_ptsb and psiu_ptsb and ADT trunks have no associations with rIPTSB's. The DPB audit found this field to be non null.
Possible Error Effect:	Possible failure of ADT, IDCUs and other non-CDMA calls.
Corrective Action Taken:	The dppto linkage is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The st_headk field of the dppto field of the rIDPB tuple.
<i>Bad Data =</i>	The st_tailk field of the dppto field of the rIDPB tuple.
<i>Good Data =</i>	The phctyp field of the rIDPB tuple.
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 3 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP

Error Description:	The corresponding rIODPSOP static tuple for a given ODB DSL and party could not be accessed. For every D-channel rIDPB tuple where the B-channel management data indicates ODB is in use on the DSL (rIDPB.b1use or b2use = DMODB_USE) there should exist a corresponding rIODPSOP static tuple where (key) rIDPB.port = (key) rIODPSOP.port and (key) rIODPSOP.party = (rIDPB.X >> rIODPSOP.party & (DMUCHAR)1) where X = b1pty_resv or b2pty_resv in the D-channel rIDPB tuple.
Possible Error Effect:	An unauthorized subscriber may be using the ODB feature.
Corrective Action Taken:	The B-channel port is quarantined (rIORTLA.quarstate set to AUAUSET) and the B-channel management data in the associated D-channel rIDPB tuple is recovered to an idle state (b1use or b2use marked DMIDLE_USE and b1pty_resv or b2pty_resv marked to 0 depending on which B-channel was using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple for the B-channel's port.
<i>Bad Data =</i>	The logical key of the D-channel rIDPB tuple (port).
<i>Good Data =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Logical Index =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Dump 1 =</i>	The rIORTLA tuple for the B-channel's port.
<i>Dump 2 =</i>	The rIDPB tuple for the D-channel.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	Either the blkcall, pktcampon or the srwait field of the rIDPB tuple is set but the D-channel port is not owned as indicated by the state of the rIORTLA tuple of this non-FRBC port (DBPSIDLE), or the port is an FRBC trunk and both the SVC jobid and the SVC job's process ID are null.
Possible Error Effect:	Possible error effects are: • If the blkcall field is set, then one can not originate or terminate to this DSL as the call is blocked. • If the pktcampon field is set, the D port indicates it is camped on when it is not. • If the srwait field is set, the D port will time out waiting for SERV and SERV ACK messages.
Corrective Action Taken:	Audit will force the recovery of the rIDPB tuple being audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The blkcall field of the rIDPB tuple.
<i>Bad Data =</i>	The pktcampon field and the srwait field of the rIDPB tuple (rIDPB.pktcampon rIDPB.srwait).
<i>Good Data =</i>	The logical port type (rIFC_PORTTYP.lp_type).
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump 1 =</i>	The rIDPB tuple being audited.
<i>Dump 2 =</i>	The rIORTLA tuple involved.

Table 5 Error Code: BADPTSB

Error Code:	BADPTSB
Error Description:	One of the PTSB's pointed to by the D-channel protocol block did not point back to the D-channel.
Possible Error Effect:	Potential for breaking the D-channel signaling path.
Corrective Action Taken:	PTSB pointers are cleared. Process owning DPB is purged if any. D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The key of the rIPTSB tuple that should have been linked to the rIDBP tuple under audit.
<i>Bad Data</i> =	PTSB number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	PORT number of D-port.
<i>Dump 1</i> =	The rIDPB tuple.
<i>Dump 2</i> =	The rIPTSB tuple.

Table 6 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	B2-channel usage field is not consistent with DSLEQUIP.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B2-channel usage field is reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The error code sequence number: 1 = Should be DMNO_SERV. 2 = Should be DMPPB_USE. 3 = Should be DMIDLE_USE.
<i>Good Data</i> =	B2-channel usage in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 7 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The pri_type field of the rIDPB tuple did not match the same field from the static rIPRIGRP tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	The pri_type field of the rIDPB will be corrected with the value found in the same field of the static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The prigrp field of the rIDPB tuple.
<i>Bad Data</i> =	The pri_type field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIPRIGRP tuple.
<i>Logical Index</i> =	The port field, key, of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	B-channel management data in DPB is not correct. Either B-channel usage is wrong or party reservation bits are not correctly set.
Possible Error Effect:	B-channel could be blocked for call processing.
Corrective Action Taken:	B-channel data are reinitialized correctly per DSLEQUIP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The error indicator. 1 = B-channel is configured as NO_SERV, but B-channel management data is not set to initialized value. 2 = B-channel is configured as X_25, but B-channel management data is not set to DMPB_USE or party reservations are not zero. 3 = B-channel is used for default service or operator use but party reservation

	is not zero or no TEDB is linked to DPB. 4 = B-channel is used for TM-USE but party reservation is not zero. 5 = B-channel is used for CSV, CSD or ODB use, but party reservation is not set or no TEDB is linked to PDB. 6 = B-channel is used for CSV, CSD or ODB use, but party reservation bit is not consistent with PCBLA (process). 7 = If B-channel is owned by process, B-channel usage should not be IDLE_USE, VRSV-USE, PPB_USE or NO_SERV. 8 = If B-channel is not owned by process and usage field is indicated as IDLE-USE, then party reservation bit should not be set. 9 = When B-channel is not owned by process but used by shared DN (usage field is DMVRSV_USE), party reservation bits should not be zero and TEDB, DALB should link to DPB. 10 = The B-channel usage field is not defined in the system.
<i>Good Data</i> =	Valid values are: 1 = B1 channel has problem. 2 = B2 channel has problem.
<i>Logical Index</i> =	PORT number of D-port.
<i>Dump</i> =	rIDPB tuple.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The value in rIPP_MEM.num_mem indicates that a certain number of rIPTSB tuples should be linked to this rIDPB when it is stable the number of tuples is wrong.
Possible Error Effect:	This packet pipe is not correctly set up for CDMA calls and will cause a loss of data.
Corrective Action Taken:	Audits will force recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the port (rIDPB.phctyp).
<i>Bad Data</i> =	The number of rIPTSB tuples that are currently linked (local cnt).
<i>Good Data</i> =	The number of rIPTSB tuples that should be linked (rIPP_MEM.num_mem).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump 1</i> =	The contents of the rIDPB tuple.
<i>Dump 2</i> =	The contents of the rIPP_MEM tuple.
<i>Dump 3</i> =	The contents of the rIFC_PORTTYP tuple.

Table 10 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The binding related data such as blport_set are not set correctly in DPB.
Possible Error Effect:	This DSL may have signaling and binding problems.
Corrective Action Taken:	blport_set in DPB is correctly set to be consistent with DSLEQUIP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	blport_set in DPB.
<i>Good Data</i> =	bind_lport in DSLEQUIP. (if bind_lport in DSLEQUIP is not zero, then blport_set in DPB should be set to DBYES).

<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 11 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The audible alert party bits should not be set when DSL is idle.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The audible alert party bits will be reset to zero.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The audible party in DPB.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 12 Error Code: BLINK

Error Code:	BLINK
Error Description:	There should be no linkage between a B-channel rIDPB tuple used for the ODB feature and TEDB, and the B-channel rIDPB tuple's tedb_flag should be DBNO. Either one or both of these conditions was found to be in error.
Possible Error Effect:	The B-channel rIDPB tuple, and related data are in an invalid state resulting in wasted resources.
Corrective Action Taken:	The B-channel rIDPB tuple, and related ODB call processing data are recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-DPB.
<i>Bad Data</i> =	The tedb_link flag.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the B-DPB tuple.
<i>Dump</i> =	The B-channel rIDPB (which will contain the status of the linkage to the TEDB).

Table 13 Error Code: BOTHCHAN

Error Code:	BOTHCHAN
Error Description:	An associated group (rIDPB.agi1_bchan or rIDPB.agi2_bchan) is using both B-channels, when only one is allowed. Terminals identified in a group by the bit map rIDSLEQUIP.agipty# coincide with both B-channels identified in bit maps rIDPB.b1pty_resv and rIDPB.b2pty_resv.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	None.
Manual Action Required:	Check assignment of associated groups and multi-line hunt groups (MLHG) on RC/V Views 23.2 (DIGITAL SUBSCRIBER LINE) and 23.3 (ASSOCIATED GROUP). Check for overlapping groups, and MLHGs which occur in both groups.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan.
<i>Good Data</i> =	rIDPB.agi2_bchan.
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump 1</i> =	rIDPB tuple.
<i>Dump 2</i> =	rIDSLEQUIP tuple.

Table 14 Error Code: CHAN_ZERO

Error Code:	CHAN_ZERO
Error Description:	B-channel management data are not set to initialized value for non-ISDN line.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B-channel management data are reset to initialized value.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 15 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of TEILA linked to DPB is not equal to those specified in DPB(<i>tot_terms</i> field).
Possible Error Effect:	The error may cause binding and signaling problem.
Corrective Action Taken:	<i>tot_terms</i> field in DPB is reset to correct number.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	<i>tot_terms</i> in DPB.
<i>Good Data</i> =	The actual TEILA linked to DPB.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 16 Error Code: DBCHAN

Error Code:	DBCHAN
Error Description:	B1-channel usage field is not consistent with DSLEQUIP.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B1-channel usage field is reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The error code sequence number: 1 = Should be DMNO_SERV. 2 = Should be DMPPB_USE. 3 = Should be DMIDLE_USE.
<i>Good Data</i> =	B1-channel usage in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 17 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	There is a non-null call activity check process ID, but the process no longer exists.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The call activity check process ID is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the rIDPB tuple (<i>rIDPB.phctype</i>).
<i>Bad Data</i> =	The process number of the call activity check pid (<i>rIDPB.cac_pid.procno</i>).
<i>Good Data</i> =	The uniqueness of the call activity check pid (<i>rIDPB.cac_pid.uniq</i>).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 18 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	An invalid linkage is found between a B-channel rIDPB tuple used for the ODB feature and an rIPTSB tuple.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	The ODB call will be torn down and related data recovered to an idle condition.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The invalid linkage between the B-channel DPB and the PTSB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DPB key.
<i>Dump 1</i> =	The rIDPB tuple.
<i>Dump 2</i> =	The rIPTSB tuple.

Table 19 Error Code: DSL_RATE

Error Code:	DSL_RATE
Error Description:	Data rate in DPB differs from ODD data rate.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Process owning DPB is purged if any D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data rate found in DPB.
<i>Good Data</i> =	Data rate according to ODD.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	NA.

Table 20 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The logical PH number and physical PH channel data in a B-channel rIDPB tuple used for ODB did not match the data contained in the corresponding rIODPSOP static tuple. For B-channel rIDPB tuples used for ODB calls, the ph_number and ph_cn attributes must be represented in rIODPSOP.ph_number and rIODPSOP.ph_cn or ph_cn2 where, for the associated D-channel, (key) rIDPB.port = (key) rIODPSOP.port and (key) rIODPSOP.party = (rIDPB.X >> rIODPSOP.party & (DMUCHAR)1) where X = b1pty_resv or b2pty_resv in the D-channel rIDPB tuple associated with the B-channel rIDPB tuple under audit.
Possible Error Effect:	An unauthorized subscriber may be using the ODB feature.
Corrective Action Taken:	The ODB call process is purged and the ODB call owning the B-channel is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-channel rIDPB tuple under audit.
<i>Bad Data</i> =	The PH number the rIDPB tuple (ph_number)
<i>Good Data</i> =	The PH number found in the rIODPSOP static tuple (ph_number)
<i>Logical Index</i> =	The key of the rIDPB tuple under audit (port)
<i>Dump 1</i> =	The B-channel rIDPB tuple under audit.
<i>Dump 2</i> =	The associated D-channel rIDPB tuple for the DSL.
<i>Dump 3</i> =	The associated rIODPSOP static tuple.

Table 21 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rIDPB dynamic tuple without a corresponding static tuple where: - the DPB does not have the static rIDSLEQUIP tuple with rIDPB.port = rIDSLEQUIP.dsl or - there is no user subscribed to the on-demand-packet feature with rIDPB.port = rIODPSOP.port
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key value.
<i>Dump</i> =	The rIDPB tuple.

Table 22 Error Code: GPORT

Error Code:	GPORT
Error Description:	The specified B-channel port is not populated or key or B-channel PORTLA is wrong.
Possible Error Effect:	Application code may assert when accessing the data and B-channel cannot be used.
Corrective Action Taken:	PORTLA or FCPORTTYPE audit will be scheduled as related audit to detect this error and correct it.
Manual Action Required:	If FC_PORTTYPE is missing, then manual action is required to insert the B-channel FC_PORTTYPE tuple. (Look good data in the dump.)
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The key of B-PORTLA in the system (maybe 0xffff).
<i>Good Data</i> =	The key of B-PORTLA which should be in the system.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 23 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Based on the bit maps rIDSLEQUIP.agipty1 and rIDPB.b#pty_resv the associated group information rIDPB.agi1_bchan is set to the incorrect value. The range of values is FCNO_BCHAN, FCB1_CHAN, and FCB2_CHAN.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	rIDPB.agi1_bchan is set to the correct value listed below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan.
<i>Good Data</i> =	Correct value of rIDPB.agi1_bchan.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 24 Error Code: GRP2INFO

Error Code:	GRP2INFO
Error Description:	Based on the bit maps rIDSLEQUIP.agipty2 and rIDPB.b#pty_resv the associated group information rIDPB.agi2_bchan is set to the incorrect value. The range of values is FCNO_BCHAN, FCB1_CHAN, and FCB2_CHAN.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	rIDPB.agi2_bchan is set to the correct value listed below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi2_bchan.
<i>Good Data</i> =	Correct value of rIDPB.agi2_bchan.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 25 Error Code: IL_PTYE

Error Code:	IL_PTYE
Error Description:	The lp_type in rIFC_PORTTYP does not indicate that the port B-channel rIDPB is associated with a B-channel port.
Possible Error Effect:	Wasted resources since this tuple cannot be used for anything.
Corrective Action Taken:	The B-channel rIDPB tuple and related data, if any, will be recovered to an idle condition.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The lp_type in rIFCPORTTYP.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the DPB tuple.
<i>Dump</i> =	The DPB tuple.

Table 26 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	EOC/TMC ports on an IDCU TR303 have a invalid link between the rIDPB and the rRTLb tuple.
Possible Error Effect:	Tasks for this RT may be lost and not completed.
Corrective Action Taken:	The EOC/TMC port will be recovered and the link restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The contents of the rIDPB tuple.

Table 27 Error Code: INV

Error Code:	INV
Error Description:	Either the interf_id or the prigrp field of the rIDPB tuple did not match the same field from the static rIDCHNPRI tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Both the interf_id and the prigrp fields of the rIDPB tuple will be corrected with the values found in the same fields of the static rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The interf_id field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDCHNPRI tuple.
<i>Good Data</i> =	The interf_id field of the rIDCHNPRI tuple.
<i>Logical Index</i> =	The prigrp field of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 28 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	BRI class of service in DPB is not consistent with DSLEQUIP.
Possible Error Effect:	This DSL may have signaling and binding problems.
Corrective Action Taken:	briclass in DPB is correctly set to be consistent with DSLEQUIP and the port is put on the to be idled list (TMLTTBI) to be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	briclass in DPB.
<i>Good Data</i> =	briclass in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 29 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The default service TEDB should link to the DSL which has subscribed default service feature but it links to other DSL.
Possible Error Effect:	No default service is provided in this DSL.
Corrective Action Taken:	The DSL will be recovered and all the TEDBs will be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The DSL which TEDB links to.
<i>Good Data</i> =	The DSL which TEDB should link to.

<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 30 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	Non-FRBC trunks must have a null SVC job identifier and null SVC job process ID this trunk has a non-null SVC attribute.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job identifier and SVC job process ID is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job identifier of the rIDPB tuple (rIDPB.svcjob_id).
<i>Good Data</i> =	The SVC job process ID of the rIDPB tuple (rIDPB.svcjob_pid).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 31 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The audit received an unrecognized or invalid return from TMdppbprt().
Possible Error Effect:	The rIDPB tuple cannot be audited.
Corrective Action Taken:	The audit skips the rIDPB tuple since it cannot access any related data required for auditing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIFC_PORTTYP static tuple passed to TMdppbprt().
<i>Bad Data</i> =	The return received from TMdppbprt().
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The rIFC_PORTTYP static tuple.

Table 32 Error Code: L2MON

Error Code:	L2MON
Error Description:	The port is requested to be monitored for protocol but the data used is inconsistent.
Possible Error Effect:	Switch may waste resources to monitor the unrequested port.
Corrective Action Taken:	The port is marked as not being monitored.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The pmon field in DPB.
<i>Good Data</i> =	DBNO(0).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 33 Error Code: LEVEL_1

Error Code:	LEVEL_1
Error Description:	Level two profile in the ODD does not match profile in the DPB.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Process owning DPB is purged if any D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Level 2 profile found in DPB.
<i>Good Data</i> =	Level 2 profile according to ODD.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	NA.

Table 34 Error Code: LEVEL_3

Error Code:	LEVEL_3

Error Description:	The level three state in the DPB did not match the level three state derived from the circuit type information in the PORTTYP and DSLEQUIP tuple.
Possible Error Effect:	Potential for breaking the D-channel signaling path.
Corrective Action Taken:	Level three state is updated to match the static ODD data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Level three state from the DPB.
<i>Good Data</i> =	Level three state derived from the ODD.
<i>Logical Index</i> =	PORT number of D-port.
<i>Dump</i> =	DPB is dumped.

Table 35 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	DSL should have default service feature but there is no default service TEDB linked to DPB or TEDB key is wrong.
Possible Error Effect:	No default service is provided in this DSL.
Corrective Action Taken:	Print the error and request related audits to insert TEDB or correct the TEDB errors.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The TEDB key in the system.
<i>Good Data</i> =	The TEDB key which should be.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 36 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The servclass and lp_type fields of the static rIFC_PORTTYPE tuple indicates that this D channel is not used for NFAS and DCBU PRI D channel(AUportbuf.servclass not != FCEDSL and AUportbuf.lp_type != PCTDCHAN). And yet audit found that one of the three fields related to NFAS and DCBU D channel, namely, interf_id, prigrp, and pri_type has a non null value.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	All the three fields will be set to the null value (interf_id->PSNO_INTID, prigrp->TMNULLPRI and pri_type->TMNULL_TYPE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The interf_id field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 37 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	For an NFAS/DCBU PRI calls, the linkage field to the rIDALB tuple dpdao must be null as no DALBs used in such type of PRI calls is linked to DPBs. Audit found this field to be non null.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Audit will force recovery of this rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The st_headk field of the dpdao field of the rIDPB tuple.
<i>Bad Data</i> =	The st_tailk field of the dpdao field of the rIDPB tuple.
<i>Good Data</i> =	The prigrp field of the rIDPB tuple.
<i>Logical Index</i> =	The port field, key, of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 38 Error Code: NOACCBK

Error Code:	NOACCBK
Error Description:	For FRBC calls, when either the SVC job ID or the SVC job PID is non-null, future SVC calls must be blocked. If the block-call bit is set to 0 (DBNO), calls are not blocked.
Possible Error Effect:	SVC calls are attempted and fail later.
Corrective Action Taken:	Audit will set the block-call bit to 1 (DBYES).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The SVC job identifier (rIDPB.svcjob_id).
<i>Bad Data =</i>	The SVC job process identifier (rIDPB.svcjob_pid.procno).
<i>Good Data =</i>	The SVC job process ID's unique bit (rIDPB.svcjob_pid.uniq).
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 39 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	While looking for a process on the port associated with the rIDPB tuple during selective initialization, the DA function call returned an error saying that database corruption exists. The audit error report dumps rPORTLA because that is what it has access to.
Possible Error Effect:	The port would normally be taken down unless a barge-in call pickup process exists for this port. Since database error was returned, the port will be taken down.
Corrective Action Taken:	The tuple is quarantined, so that the audit forces recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The pointer to the rPORTLA tuple.
<i>Bad Data =</i>	The value returned from the DA function call.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rPORTLA tuple's key, port field.
<i>Dump =</i>	The rPORTLA tuple being audited.

Table 40 Error Code: NOTIDLE2

Error Code:	NOTIDLE2
Error Description:	A rIDPB tuple not currently used for ODB was found to have its logical PH number (ph_number) and physical channel number (ph_cn) attributes set to invalid values. For D-channel rIDPB tuples, or rIDPB B-channel tuples which are not associated with B-channels used for the ODB feature, the PH and channel attributes are unused and should be set to idle values (PSPHNULL and zero respectively).
Possible Error Effect:	Call processing will never be able to use the associated B-channel for ODB since the PH and channel attributes make it appear that the B-channel already has a PH and channel assigned and in use.
Corrective Action Taken:	The ph_number and ph_cn attributes are recovered to their idle values (PSPHNULL and 0 respectively).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The contents of the ph_number and ph_cn attributes. (rIDPB.ph_number << 4 rIDPB.ph_cn)
<i>Good Data =</i>	The null values for ph_number and ph_cn (PSPHNULL << 4 (DMPHCN) 0).
<i>Logical Index =</i>	The rIDPB tuple's key (port).
<i>Dump =</i>	The rIDPB tuple.

Table 41 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The PH channel type indicates that this is a ADT, packet pipe or speech handler trunk so the semantic linkage to the rIPTSB (islu_ptsb or psiu_ptsb) should be

	zero, but instead is non-zero.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	The semantic linkage to the rIPTSB is nulled out (rIDPB.islu_ptsb and rIDPB.psiu_ptsb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The islu_ptsb field of the rIDPB tuple.
<i>Bad Data =</i>	The psiu_ptsb field of the rIDPB tuple.
<i>Good Data =</i>	The phctyp field of the rIDPB tuple.
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 42 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The RLPORRTYP relation indicates that the DSL for this port must be equipped, but the rIDSLEQUIP tuple is missing from the static ODD. If the <i>Good Data</i> is 2, the DPB audit was unable to obtain the static relation tuple (rIPP_MEM) needed to audit the dynamic rIDPB tuple for the CDMA packet pipe either there is no packet pipe member tuple with the same port as the DPB or PP_MEM key is wrong.
Possible Error Effect:	If the <i>Good Data</i> is 1, the audit cannot verify any data in the DPB. If it is bad, the audit has no way to fix it. If the <i>Good Data</i> is 2, packet pipe members cannot be located for packet pipe call processing.
Corrective Action Taken:	None.
Manual Action Required:	If the <i>Good Data</i> is 1, the craft will need to use ODBE to put the missing DSLEQUIP tuple back into the ODD, since ODBE does not permit removing the PORTTYP tuple under these circumstances. If the <i>GOOD DATA</i> is 2, the craft will need to use ODBE to put the missing PP_MEM tuple into the ODD. Find out the value of the port.member from the dump of the rIDPB tuple. The port.module will be the processor receiving the audit. Using these as the key, insert the static rIPP_MEM tuple. For the other fields of the tuple, refer to the terminal maintenance and trigger area.
Dump Description:	
<i>Error Address =</i>	If the <i>Good Data</i> is 1, the address of the rIDSLEQUIP tuple. If the <i>Good Data</i> is 2, the address of the rIDPB tuple.
<i>Bad Data =</i>	The logical port type (rIFC_PORTTYP.lp_type).
<i>Good Data =</i>	Valid values are: 1 = Could not read the rIDSLEQUIP tuple. 2 = Could not read the rIPP_MEM tuple.
<i>Logical Index =</i>	The rIDPB key (rIDPB.port).
<i>Dump 1 =</i>	The contents of the rIDPB tuple.
<i>Dump 2 =</i>	The contents of the rIFC_PORTTYP tuple.

Table 43 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An FRBC trunk has a non-null SVC job identifier, but it is not within the range of valid values for such a field.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job identifier is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data =</i>	The SVC job identifier of the rIDPB tuple (rIDPB.svcjob_id).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDPB key - port name.

<i>Dump</i> =	The rIDPB tuple.

Table 44 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	DPB tuples that are not used by IDCU or certain types of CDMA must be linked to TEDB, DALB, and/or TEILA tuples. This DPB tuple is not correctly linked to one of them. If the <i>Good Data</i> is 1, it is incorrectly linked to TEDB 2, DALB or 3, TEILA.
Possible Error Effect:	Problem in call processing.
Corrective Action Taken:	The rIDPB tuple is quarantined so that the tuple is recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The linkage, either TEDB, DALB, or TEILA. (Refer to <i>Good Data</i> .)
<i>Good Data</i> =	Valid values are: 1 = The bad linkage is to a TEDB. 2 = The bad linkage is to a DALB. 3 = The bad linkage is to a TEILA.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 45 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	The PH channel circuit type stored in an rIDPB tuple, which is associated with a B-channel port used for the ODB feature, was found inconsistent from the type derived from the B-channel management data stored in static ODD (rIFC_PORTTYPE).
Possible Error Effect:	The PORT could be OOS.
Corrective Action Taken:	The "phctype" field is updated in the DPB to agree with the B-channel management data stored in static ODD (rIFC_PORTTYP).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DPB tuple.
<i>Bad Data</i> =	Value of the "phctype" field in the DPB.
<i>Good Data</i> =	Value of "phctyp" computed from the ODD.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The DPB tuple.

Table 46 Error Code: PHDSLGP

Error Code:	PHDSLGP
Error Description:	There are currently two ODB calls up on this DSL. However, the logical PH and physical PH channel numbers stored in the rIDPB tuples for each B-channel are identical. When ODB calls are up on each B-channel of a DSL, the ph_number and ph_cn attributes in each ODB B-channel rIDPB tuple must be unique to each other. Since it is physically impossible to set up ODB calls on a DSL under these conditions, this error is an indication that the ph_number and ph_cn attributes in one or both rIDPB tuples became corrupted after both ODB calls were set up on the DSL. The occurrence of this error is an indication that the PH and channel data does not reflect the actual PH and channel(s) being used for ODB call activities.
Possible Error Effect:	The ODB connection(s) on this DSL cannot be cleanly torn down. Attempts by call processing to do so may result in asserts and/or audit errors.
Corrective Action Taken:	Both B-channel rIORTLA tuples are quarantined for recovery (rIORTLA.quarstate marked AUAUSET). The B-channel management data in

	the D-channel rIDPB tuple (b1use, b2use, b1pty_resv and b2pty_resv) are marked to idle states (DMIDLE_USE and 0 respectively). Lastly, the logical PH number (ph_number) and physical channel number (ph_cn) attributes in the B1 and B2 rIDPB tuples are marked to an idle state (PSPHNULL and 0 respectively). Any ODB call processes up on the DSL will then be purged and any other related call data will be idled out by related audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the B-channel rIDPB ODB B-channel tuple under audit.
<i>Bad Data =</i>	The logical PH number (ph_number) in the B-channel rIDPB ODB B-channel tuple under audit.
<i>Good Data =</i>	The physical PH channel number (ph_cn) in the B-channel rIDPB ODB B-channel tuple under audit.
<i>Logical Index =</i>	The key of the B-channel rIDPB ODB B-channel tuple (port).
<i>Dump 1 =</i>	The contents of the B-channel rIDPB ODB B-channel tuple under audit.
<i>Dump 2 =</i>	The contents of the "other" B-channel rIDPB ODB B-channel tuple.
<i>Dump 3 =</i>	The contents of the D-channel rIDPB tuple on the DSL.

Table 47 Error Code: PHNUNEQUIP

Error Code:	PHNUNEQUIP
Error Description:	The dsl_group and/or member obtain from the static tuple rIDSLEQUIP do not match the ph_number and/or ph_cn stored in the rIDPB tuple.
Possible Error Effect:	Problems in Call Processing.
Corrective Action Taken:	The rIDPB tuple is recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	PH number (rIDPB.ph_number).
<i>Good Data =</i>	DSL_group (rIDSLEQUIP.dsl_group).
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump 1 =</i>	The rIDPB tuple being audited.
<i>Dump 2 =</i>	The rIDSLEQUIP tuple being audited.

Table 48 Error Code: PLE_STATE

Error Code:	PLE_STATE
Error Description:	The plestate attribute is incorrect.
Possible Error Effect:	D-channel will be idled.
Corrective Action Taken:	The rIDPB tuple will be initialized and made idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPB.
<i>Bad Data =</i>	Incorrect plestate.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIDPB.
<i>Dump =</i>	rIDPB.

Table 49 Error Code: PORT

Error Code:	PORT
Error Description:	Using port field of the rIDPB tuple as the member field and GI_pcrd as the module field of the dport key field of rIDCHNPRI tuple, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it is inserted by office personnel. Refer to the Manual Action Required field as to how to insert the tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Both the interf_id and the prigrp fields of the rIDPB tuple will be reset to their null values and audit will retun.
Manual Action Required:	Using port field of the rIDPB tuple as the member field and GI_pcrd as the module field of the dport key field of rIDCHNPRI tuple, insert the static rIDCHNPRI tuple. As for the values of the other fields, refer to the terminal maintenance and triggers area.

Dump Description:	
<i>Error Address</i> =	The prigrp field of the rIDPB tuple.
<i>Bad Data</i> =	The interf_id field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 50 Error Code: PROC_LOST

Error Code:	PROC_LOST
Error Description:	An FRBC trunk has a non-null SVC job process ID, but the process is non-existent.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job process ID is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job process ID's process number (rIDPB.svcjob_pid.procn).
<i>Good Data</i> =	The SVC job process ID's uniqueness (rIDPB.svcjob_pid.uniq).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 51 Error Code: PROC_TO

Error Code:	PROC_TO
Error Description:	The process owning B-channel port does not link properly with the B-channel PORTLA.
Possible Error Effect:	Application code may assert when accessing the data and B-channel cannot be used.
Corrective Action Taken:	The PORTLA will be quarantined and process will be purged.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The process number of PCBLA.
<i>Good Data</i> =	The expected process number from PORTLA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 52 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The level 3 profile obtained from rIFC_PORTTYP and rIODPSOP static ODD did not match the level 3 profile stored in a B-channel rIDPB tuple being used for the ODB feature.
Possible Error Effect:	The B-channel signaling path may be corrupted.
Corrective Action Taken:	The level 3 data will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-DPB tuple.
<i>Bad Data</i> =	The level 3 profile found in the B-DPB tuple.
<i>Good Data</i> =	The level 3 profile derived from static data.
<i>Logical Index</i> =	The key of the B-DPB tuple.
<i>Dump</i> =	The B-channel DPB tuple.

Table 53 Error Code: PTR3

Error Code:	PTR3
Error Description:	According to the D-channel rIDPB tuple for a DSL with two ODB B-channels assigned, both B-channels currently have ODB calls up, however, only one rIDPB B-channel tuple could be accessed by the audit. For each B-channel "use" field in the D-channel rIDPB tuple which is marked to indicate ODB is in use on the respective B-channel (b1use and/or b2use = DMOB_USE) there should exist a corresponding ODB B-channel rIDPB tuple.
Possible Error Effect:	If the B-channel is assigned the ODB feature, then ODB calls can never be

	originated on this B-channel.
Corrective Action Taken:	The B-channel management data in the associated D-channel rIDPB tuple which corresponds to the B-channel under audit will be recovered to an idle state (b1use or b2use set to DMIDLE_USE and b1pty_resv or b2pty_resv set to 0 depending on which B-channel was marked as using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the D-channel rIDPB tuple for the DSL.
<i>Bad Data =</i>	Informational: The b1use field in the D-channel rIDPB tuple.
<i>Good Data =</i>	Informational: The b2use field in the D-channel rIDPB tuple.
<i>Logical Index =</i>	The key of the D-channel rIDPB tuple (port)
<i>Dump =</i>	The D-channel rIDPB tuple.

Table 54 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The routine port conditioning table indicates that a call path should be up (TMportcond[rIORTLA.list where rIORTLA.port = rIDPB.port].ispathup = TMPCYES) but the rIDPB tuple is not linked to an rIPTSB tuple (rIDPB.islu_ptsb = 0 and rIDPB.psiu_ptsb = 0) and no process owns the port (rIORTLA.proc_cplr.st_headk = 0 where rIORTLA.port = rIDPB.port). The rIDPB tuple should be linked to an rIPTSB tuple if the routine port conditioning table indicates a path should be up for a given port list except if the port is owned by a process.
Possible Error Effect:	The time slots associated with the rIPTSB tuple will be lost to the system, thus reducing the total number of available time slots.
Corrective Action Taken:	The rIDPB tuple is recovered to an idle state, and the rIORTLA tuple is placed on the to-be-idled list (rIORTLA.list = TMLTTBI), and the terminal maintenance unloader is awakened to process the rIORTLA tuple so that it is eventually set to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The list the port associated with the RLDPB is on (rIORTLA.list where rIORTLA.port = rIDPB.port).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key (rIDPB.port) of the rIDPB tuple.
<i>Dump 1 =</i>	The contents of the rIDPB tuple.
<i>Dump 2 =</i>	The contents of the associated rIORTLA tuple.

Table 55 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The port is not on a recovery list, so the quarantine state should not be in a race condition, but it is.
Possible Error Effect:	Lost resources since this port is out-of-service.
Corrective Action Taken:	The rIDPB tuple is quarantined so it can be recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The quarantine state (rIDPB.quarstat).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump =</i>	NA.

Table 56 Error Code: QUE

Error Code:	QUE
Error Description:	An FRBC trunk has a non-null SVC job identifier, but the ISPA job is no longer on the queue.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job identifier is nulled out.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job identifier of the rIDPB tuple (rIDPB.svcjob_id).
<i>Good Data</i> =	The return value from the function call, TMisqjobid.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 57 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The number of TEILA linked to DPB is greater than what user subscribes (specify in DSLEQUIP).
Possible Error Effect:	The error may cause binding and signaling problem.
Corrective Action Taken:	The DSL will be reinitialized and recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	Number of TEILA linked to DPB.
<i>Good Data</i> =	Number of TEILA allowed in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 58 Error Code: SGNOOD

Error Code:	SGNOOD
Error Description:	Using prigrp field of the rIDPB tuple as the key field of rIPRIGRP tuple, audit tried to read the static rIPRIGRP tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it is inserted by office personnel. Refer to the manual action required field as to how to insert the tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	The pri_type field of the rIDPB tuple will be reset to its null value(TMNULL_TYPE) and audit will return.
Manual Action Required:	Find out the value of the prigrp field from the dump of the rIDPB tuple. Use it as the key field insert the static rIPRIGRP tuple. As for the other fields of the tuple, refer to the terminal maintenance and trigger area.
Dump Description:	
<i>Error Address</i> =	The pri_type field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDPB tuple.
<i>Good Data</i> =	The interf_id field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 59 Error Code: TEDBFLAG

Error Code:	TEDBFLAG
Error Description:	The tedb_flag attribute in the DPB tuple is incorrectly set.
Possible Error Effect:	An incorrect linkage may exist between the DPB and either a TEDB tuple or a DALB tuple. This may effect call processing.
Corrective Action Taken:	The tedb_flag will be correctly set according to the servclass found at FC_PORTTYPE. If it is ISDN or ISDN attendant DSL, then tedb_flag will be set to DBYES. If it is an EOC/TMC port, then the tedb_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Valid values are: 0 = If the tedb_flag should be set to DBYES. 1 = If the tedb_flag should be set to DBNO.
<i>Good Data</i> =	Valid values are: 0 = If not ISDN and ISDN attendant or if EOC/TMC port.

	1 = If it is ISDN or ISDN attendant.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	RLDPB.

Table 60 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	An invalid linkage is found between the rIDPB tuple and a rTEDB tuple.
Possible Error Effect:	Switch cannot communicate with customer premise equipment. EOC/TMC port on IDCU TR303 have lost communication with the switch.
Corrective Action Taken:	Port will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Linkage value from rIDPB tuple to rTEDB (rIDPB.tedb_ own.st_headk).
<i>Good Data</i> =	0. There should not be a linkage between the tuples.
<i>Logical Index</i> =	The rIDPB key.
<i>Dump</i> =	The contents of rIDPB tuple.

Table 61 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The fixed terminal endpoint identifier (FTEI) flag indicator in the rIDPB tuple is incorrect. It does not match the same indicator in the corresponding rIDSLEQUIP tuple. This flag is used by call processing to determine if there is a FTEI resource on this digital subscriber line (DSL).
Possible Error Effect:	FTEI resources on this DSL may not function correctly on this DSL.
Corrective Action Taken:	The flag indicator (pfitei) is recovered to the value specified in the corresponding rIDSLEQUIP flag.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of pfitei in the rIDPB before correction.
<i>Good Data</i> =	The value of pfitei in the corresponding rIDSLEQUIP tuple.
<i>Logical Index</i> =	The rIDPB key.
<i>Dump 1</i> =	The contents of rIDPB tuple.
<i>Dump 2</i> =	The contents of rIDSLEQUIP tuple.

Table 62 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	For non-ISDN line. TEILA should not link to DPB.
Possible Error Effect:	Service cannot be received from this port.
Corrective Action Taken:	The port will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The port which TEILA links to.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 63 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	The tpwait field associated with this DPB is not set correctly according to data in the PORTLA.
Possible Error Effect:	"tpwait" should be on for EOC/TMC ports plus X.75, X.75', FRBC and IP trunks when ISPA terminal process (TP) is waiting for a level 2 setup message, and off for all other ports.
Corrective Action Taken:	The tpwait field will be updated according to the information present in the PORTLA if this is the only error detected.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Service class of the dpb tuple (rIFC_PORTTYP.servclass).
<i>Bad Data</i> =	Value of "tpwait" field in the DPB.
<i>Good Data</i> =	"tpwait" value determined from PORTLA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	The DPB tuple.
<i>Dump 2</i> =	The PORTLA tuple.

Table 64 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	A broken linkage is found when traversing the rTEILA tuple through this rIDPB tuple. A broken linkage to rRTEILA tuple is found on an IDCUC TR303 EOC/TMC port.
Possible Error Effect:	Application codes may assert and some of CPEs may not receive any service.
Corrective Action Taken:	The DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 65 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PH channel type stored in the DPB does not agree with that which was computed from the ODD.
Possible Error Effect:	The PORT could be OOS.
Corrective Action Taken:	The "phctyp" field is updated in the DPB to agree with the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DPB tuple.
<i>Bad Data</i> =	Value of the "phctyp" field in the DPB.
<i>Good Data</i> =	Value of "phctyp" computer from the ODD.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The rIDPB tuple.

Table 66 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	rIDPB.agi1_bchan and rIDPB.agi2_bchan are equal. This means that both groups are using the same B-channel, which is not allowed.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	None.
Manual Action Required:	Check assignment of associated groups and multi-Line hunt groups (MLHG) on RC/V View 23.2 (DIGITAL SUBSCRIBER LINE) and 23.3 (ASSOCIATED GROUP). Check for overlapping groups, and MLHGs which occur in both groups.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan
<i>Good Data</i> =	rIDPB.agi2_bchan
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 67 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	All rIDPB tuples for EOC/TMC ports on an IDCUC TR303 should have a linkage to an rRTEILA tuple.
Possible Error Effect:	Message flow between the SM and IDCUC RT303 is interrupted.
Corrective Action Taken:	EOC/TMC port is recovered and communication restarted.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of DPB (rDPB.port).
<i>Dump</i> =	The contents of the rDPB tuple.

DPB-C

Software Release: 5E13 and later

1. D-CHANNEL PROTOCOL BLOCK (DPB) AUDIT

The D-channel application linkage block (DALB) and the D-channel protocol block (DPB) are used to handle communications between the integrated services digital network (ISDN) periphery and the processes that control the ISDN ports. The DPB corresponds to the D-channel port, or the permanent packet of the channel, and has one DALB linked to it for each process that needs to communicate with the ISDN customer premises equipment (CPE). Most (but not all) of these processes control B-port served by the D-port.

For the integrated digital carrier unit (IDCU) the rIDPB tuple is used to represent the embedded operation channel (EOC) and the timeslot management channel (TMC) channels. A shared linkage is used to link these rIDPB tuples to the IDCU remote terminal linkage block (rIRTLB) instead of a rIDALB linkage and to the remote terminal identifier linkage area (rIRTEILA) instead of the rTEILA tuple.

In addition to the DALB linkages, the DPB contains pointers to the path through the switch network that connects the D-channel line card (T-card or U-card) or the inter-switch packet trunk to the protocol handler (PH), and the information required to control the handshaking and data transmission over the D-channel to the ISDN CPE.

2. DATA STRUCTURES

DPB is used for all ISDN applications which connect a port to a PH. Examples of PH ports are:

- Digital subscriber line (DSL) D-channel.
- DSL B-channel for packet switching.
- Interswitch packet trunk.
- Inter-switching module (SM) port.
- Data link terminal (DLT) port.
- Extended digital subscriber line (EDSL) D-channel.
- DSL B-channel for on-demand packet switching.
- Integrated digital carrier unit (IDCU) EOC/TMC channels.
- CDMA packet pipe trunk.
- CDMA speech handler (PHV) trunk.

The RLDPB, an indexed relation in the switching modules (SMs), contains the D-channel port name and peripheral interface data bus (PIDB) time slot block keys for the integrated services line unit (ISLU) and packet switching units (PSUs). The audit also uses the D-channel application linkage block relation (RLDALB), the PIDB time slot block relation (RLPTSB), the port type relation (RLPORTTYP), the digital subscriber line (SDL) equipage relation (RLDSLEQUIP), and the port linkage area relation (RLPORTLA).

For an ISDN or ISDN attendant D-channel DPB, services subscribed by the users (represented by TEDB) are initialized and linked to the DPB during full initialization.

When the CPE is plugged in and terminal end point identifier (TEI) is generated, the TEI linkage area (TEILA) is linked to the DPB.

This only applies to ISDN or ISDN ATTENDANT D-channels. The other channels, such as operator services position system (OSPS), CDMA, EDSL or PRI, have no TEDB or TEILA linked to the DPB. For non-OSPS PRI ISDN calls or CDMA packet pipe or speech handler trunks, no DALB is linked to the DPB. Please refer to the data diagrams for the DPB and related data structure. Refer to Figure 1 and Figure 2 .

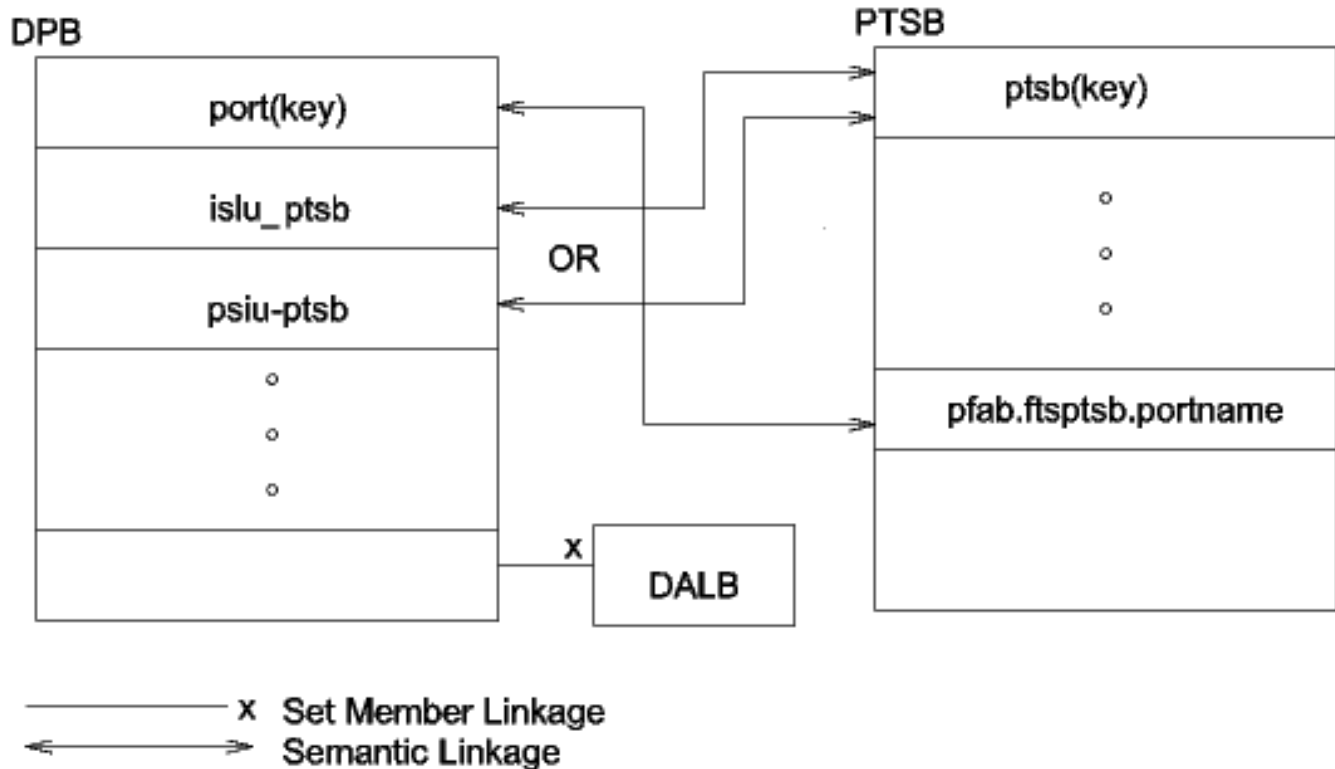


Figure 1 (DPB-1) Non ISDN or ISDN Attendant D-Channel

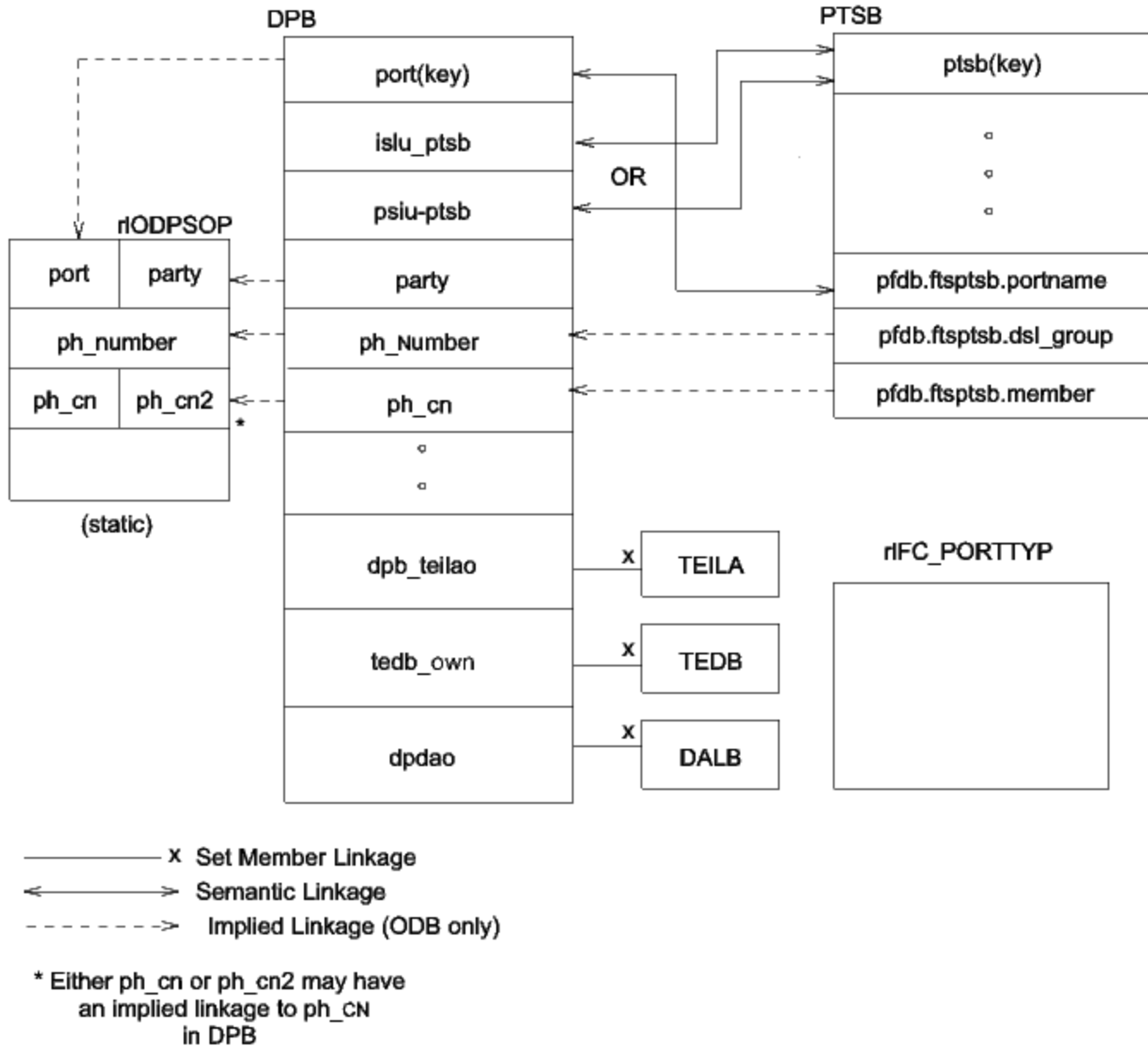


Figure 2 (DPB-2) Non ISDN or ISDN Attendant D-Channel

3. AUDIT CHECKS

- The audit first verifies that the DPB belongs to a valid D-channel or packet B-channel.
- The audit verifies that the key in DPB is correct.
- The audit verifies that all linkages between the DALB and the DPB are correct. This linkages will be null when the rIDALB tuple is involved with primary rate interface (PRI) calls for a non-facility associated signaling (NFAS) and D-channel back up (DCBU) features excluding OSPS PRI calls.
- The audit verifies that all linkages between the DPB and the following are correct:
 - Terminal equipment data block (TEDB).

- Remote terminal linkage block (rRTLb).
 - Remote terminal endpoint identifier linkage area (rRTEILA).
 - TEI linkage area (TEILA).
 - Peripheral interface data bus time slot backs (PTSBs).
- (e) The audit verifies that data transmission rates and protocol data in the DPB are correct.
- (f) If channel is being monitored for protocol transmission, the audit verifies that the protocol monitor job tables are consistent each other.
- (g) The audit verifies that B-channels management data stored in the PDB is consistent with the corresponding B-channel status in PORTLA and B-channel configuration in rIDSLEQUIP.
- (h) The audit verifies the path between the DPB and the PTSB is set per the routine port condition table.
- (i) For CDMA packet pipe trunks, the audit verifies that the number of PTSB's that should be associated with the packet pipe (according to the count in rIPP_MEM) are all there.
- (j) For ISDN or ISDN attendant lines, the D-channel DPB should link to the terminal equipment data block (TEDB). Other channels such as OSPS, extended digital subscriber lines (EDSLs) should not have linkages to TEDBs.
- (k) The audit verifies the number of TEILAs linked to DPB is the same as the specified in the DPB and is not greater than the maximum number configured in the rIDSLEQUIP.
- (l) The audit verifies that the basic rate interface (BRI) class and binding data stored in the DPB is derived from the corresponding rIDSLEQUIP tuple.
- (m) The audit provides existence checks on B-channel DPBs used for on demand B-channel package services (ODB). The audit also verifies the consistency of B-channel management data related to ODB.
- (n) The audit verifies the associated group information (rIDPB.agi1_bchan and rIDPB.agi2_bchan).
- (o) The audit verifies all the fields related to the NFAS/DCBU features for PRI calls, namely, interf_id, prigrp, pri_type and srvwait.

4. ERROR HANDLING

If any errors are detected in a DPB tuple, the audit determines if a process currently owns the D-channel. If a process is found, the audit recovers the data by requesting a single process purge of the owning process.

If the DPB does not have a living process, the audit reinitializes the protocol and data rate information. The audit attempts to preserve the DALB, TEDB, and TEILA lists and the physical path through the PTSB's, since changes to either of these could result in lost data. The audit recovers these linkages if they do not pass the point to point back checks. (NON-ODB calls only. For ODB calls, the tuple and all linkages are idled.)

During recovery the rIDPB field is cleared to DBNO.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ACCPTR

Error Code:	ACCPTR
Error Description:	The ODB B-channel rIDPB tuple corresponding to a given rIORTLA tuple associated with the B-channel's port could not be accessed by the audit. For every rIORTLA tuple associated with an ODB B-channel port there should exist a corresponding B-channel rIDPB tuple where rIORTLA.port = rIDPB.port.
Possible Error Effect:	ODB calls cannot be set up on one of the B-channels for this DSL.
Corrective Action Taken:	The B-channel management data in the D-channel rIDPB tuple is idled out to indicate that no ODB calls are up on the B-channel (b1use or b2use marked to DMIDDLE_USE and b1pty_resb or b2pty_resv marked to 0 depending on which B-channel was using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple for the B-channel's port.
<i>Bad Data =</i>	The logical key of the D-channel rIDPB tuple (port).
<i>Good Data =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Logical Index =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Dump 1 =</i>	The rIORTLA tuple for the B-channel's port.
<i>Dump 2 =</i>	The rIDPB tuple for the D-channel.

Table 2 Error Code: ASSOC_PTSB

Error Code:	ASSOC_PTSB
Error Description:	For ADT and non-CDMA calls, the owner linkage field to the rIPTSB tuple dppto must be null, as non-CDMA PTSBs are linked through the semantic linkages, islu_ptsb and psiu_ptsb and ADT trunks have no associations with rIPTSB's. The DPB audit found this field to be non null.
Possible Error Effect:	Possible failure of ADT, IDCU and other non-CDMA calls.
Corrective Action Taken:	The dppto linkage is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The st_headk field of the dppto field of the rIDPB tuple.
<i>Bad Data =</i>	The st_talk field of the dppto field of the rIDPB tuple.
<i>Good Data =</i>	The phctyp field of the rIDPB tuple.
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 3 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	The corresponding rIODPSOP static tuple for a given ODB DSL and party could not be accessed. For every D-channel rIDPB tuple where the B-channel management data indicates ODB is in use on the DSL (rIDPB.b1use or b2use = DMODB_USE) there should exist a corresponding rIODPSOP static tuple where (key) rIDPB.port = (key) rIODPSOP.port and (key) rIODPSOP.party = (rIDPB.X >> rIODPSOP.party & (DMUCHAR)1) where X = b1pty_resv or b2pty_resv in the D-channel rIDPB tuple.
Possible Error Effect:	An unauthorized subscriber may be using the ODB feature.
Corrective Action Taken:	The B-channel port is quarantined (rIORTLA.quarstate set to AUAUSET) and the B-channel management data in the associated D-channel rIDPB tuple is recovered to an idle state (b1use or b2use marked DMIDDLE_USE and b1pty_resv or b2pty_resv marked to 0 depending on which B-channel was using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple for the B-channel's port.
<i>Bad Data =</i>	The logical key of the D-channel rIDPB tuple (port).
<i>Good Data =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Logical Index =</i>	The key of the B-channel rIORTLA tuple (port).
<i>Dump 1 =</i>	The rIORTLA tuple for the B-channel's port.
<i>Dump 2 =</i>	The rIDPB tuple for the D-channel.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	Either the blkcall, pktcampon or the srwait field of the rIDPB tuple is set but the D-channel port is not owned as indicated by the state of the rIORTLA tuple of this non-FRBC port (DBPSIDLE), or the port is an FRBC trunk and both the SVC jobid and the SVC job's process ID are null.
Possible Error Effect:	Possible error effects are: • If the blkcall field is set, then one can not originate or terminate to this DSL as the call is blocked. • If the pktcampon field is set, the D port indicates it is camped on when it is not. • If the srwait field is set, the D port will time out waiting for SERV and SERV ACK messages.
Corrective Action Taken:	Audit will force the recovery of the rIDPB tuple being audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The blkcall field of the rIDPB tuple.
<i>Bad Data =</i>	The pktcampon field and the srwait field of the rIDPB tuple (rIDPB.pktcampon rIDPB.srwait).
<i>Good Data =</i>	The logical port type (rIFC_PORTTYP.lp_type).
<i>Logical Index =</i>	The port field, key, of the rIDPB tuple.
<i>Dump 1 =</i>	The rIDPB tuple being audited.
<i>Dump 2 =</i>	The rIORTLA tuple involved.

Table 5 Error Code: BADPTSB

Error Code:	BADPTSB
Error Description:	One of the PTSB's pointed to by the D-channel protocol block did not point back to the D-channel.
Possible Error Effect:	Potential for breaking the D-channel signaling path.
Corrective Action Taken:	PTSB pointers are cleared. Process owning DPB is purged if any. D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The key of the rIPTSB tuple that should have been linked to the rIDBP tuple under audit.
<i>Bad Data =</i>	PTSB number.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	PORT number of D-port.
<i>Dump 1 =</i>	The rIDPB tuple.
<i>Dump 2 =</i>	The rIPTSB tuple.

Table 6 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	B2-channel usage field is not consistent with DSLEQUIP.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B2-channel usage field is reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The error code sequence number: 1 = Should be DMNO_SERV. 2 = Should be DMPPB_USE.

	3 = Should be DMIDLE_USE.
<i>Good Data</i> =	B2-channel usage in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 7 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The pri_type field of the rIDPB tuple did not match the same field from the static rIPRIGRP tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	The pri_type field of the rIDPB will be corrected with the value found in the same field of the static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The prigrp field of the rIDPB tuple.
<i>Bad Data</i> =	The pri_type field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIPRIGRP tuple.
<i>Logical Index</i> =	The port field, key, of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 8 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	B-channel management data in DPB is not correct. Either B-channel usage is wrong or party reservation bits are not correctly set.
Possible Error Effect:	B-channel could be blocked for call processing.
Corrective Action Taken:	B-channel data are reinitialized correctly per DSLEQUIP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The error indicator. 1 = B-channel is configured as NO_SERV, but B-channel management data is not set to initialized value. 2 = B-channel is configured as X_25, but B-channel management data is not set to DMPB_USE or party reservations are not zero. 3 = B-channel is used for default service or operator use but party reservation is not zero or no TEDB is linked to DPB. 4 = B-channel is used for TM-USE but party reservation is not zero. 5 = B-channel is used for CSV, CSD or ODB use, but party reservation is not set or no TEDB is linked to PDB. 6 = B-channel is used for CSV, CSD or ODB use, but party reservation bit is not consistent with PCBLA (process). 7 = If B-channel is owned by process, B-channel usage should not be IDLE_USE, VRSV-USE, PPB_USE or NO_SERV. 8 = If B-channel is not owned by process and usage field is indicated as IDLE-USE, then party reservation bit should not be set. 9 = When B-channel is not owned by process but used by shared DN (usage field is DMVRSV_USE), party reservation bits should not be zero and TEDB, DALB should link to DPB.

	10 = The B-channel usage field is not defined in the system.
<i>Good Data</i> =	Valid values are: 1 = B1 channel has problem. 2 = B2 channel has problem.
<i>Logical Index</i> =	PORT number of D-port.
<i>Dump</i> =	rIDPB tuple.

Table 9 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The value in rIPP_MEM.num_mem indicates that a certain number of rIPTSB tuples should be linked to this rIDPB when it is stable the number of tuples is wrong.
Possible Error Effect:	This packet pipe is not correctly set up for CDMA calls and will cause a loss of data.
Corrective Action Taken:	Audits will force recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the port (rIDPB.phctyp).
<i>Bad Data</i> =	The number of rIPTSB tuples that are currently linked (local cnt).
<i>Good Data</i> =	The number of rIPTSB tuples that should be linked (rIPP_MEM.num_mem).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump 1</i> =	The contents of the rIDPB tuple.
<i>Dump 2</i> =	The contents of the rIPP_MEM tuple.
<i>Dump 3</i> =	The contents of the rIFC_PORTTYP tuple.

Table 10 Error Code: BDPID

Error Code:	BDPID
Error Description:	There is a non-null owner process ID, but the process does not exist any longer.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The owner process ID is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the rIDPB tuple (rIDPB.phctype).
<i>Bad Data</i> =	The process number of the owner process ID (rIDPB.opid.procno).
<i>Good Data</i> =	The uniqueness of the owner pid (rIDPB.opid.uniq).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 11 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The binding related data such as blport_set are not set correctly in DPB.
Possible Error Effect:	This DSL may have signaling and binding problems.
Corrective Action Taken:	blport_set in DPB is correctly set to be consistent with DSLEQUIP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	blport_set in DPB.
<i>Good Data</i> =	bind_lport in DSLEQUIP. (if bind_lport in DSLEQUIP is not zero, then blport_set in DPB should be set to DBYES).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 12 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The audible alert party bits should not be set when DSL is idle.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The audible alert party bits will be reset to zero.

Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The audible party in DPB.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 13 Error Code: BLINK

Error Code:	BLINK
Error Description:	There should be no linkage between a B-channel rIDPB tuple used for the ODB feature and TEDB, and the B-channel rIDPB tuple's tedb_flag should be DBNO. Either one or both of these conditions was found to be in error.
Possible Error Effect:	The B-channel rIDPB tuple, and related data are in an invalid state resulting in wasted resources.
Corrective Action Taken:	The B-channel rIDPB tuple, and related ODB call processing data are recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-DPB.
<i>Bad Data</i> =	The tedb_link flag.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the B-DPB tuple.
<i>Dump</i> =	The B-channel rIDPB (which will contain the status of the linkage to the TEDB).

Table 14 Error Code: BOTHCHAN

Error Code:	BOTHCHAN
Error Description:	An associated group (rIDPB.agi1_bchan or rIDPB.agi2_bchan) is using both B-channels, when only one is allowed. Terminals identified in a group by the bit map rIDSLEQUIP.agipty# coincide with both B-channels identified in bit maps rIDPB.b1pty_resv and rIDPB.b2pty_resv.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	None.
Manual Action Required:	Check assignment of associated groups and multi-line hunt groups (MLHG) on RC/V Views 23.2 (DIGITAL SUBSCRIBER LINE) and 23.3 (ASSOCIATED GROUP). Check for overlapping groups, and MLHGs which occur in both groups.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan.
<i>Good Data</i> =	rIDPB.agi2_bchan.
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump 1</i> =	rIDPB tuple.
<i>Dump 2</i> =	rIDSLEQUIP tuple.

Table 15 Error Code: CHAN_ZERO

Error Code:	CHAN_ZERO
Error Description:	B-channel management data are not set to initialized value for non-ISDN line.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B-channel management data are reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 16 Error Code: COUNTS

Error Code:	COUNTS

Error Description:	The number of TEILA linked to DPB is not equal to those specified in DPB(tot_terms field).
Possible Error Effect:	The error may cause binding and signaling problem.
Corrective Action Taken:	tot_terms field in DPB is reset to correct number.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	tot_terms in DPB.
<i>Good Data</i> =	The actual TEILA linked to DPB.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 17 Error Code: DBCHAN

Error Code:	DBCHAN
Error Description:	B1-channel usage field is not consistent with DSLEQUIP.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	B1-channel usage field is reset to initialized value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The error code sequence number: 1 = Should be DMNO_SERV. 2 = Should be DMPPB_USE. 3 = Should be DMIDLE_USE.
<i>Good Data</i> =	B1-channel usage in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 18 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	There is a non-null call activity check process ID, but the process no longer exists.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The call activity check process ID is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the rIDPB tuple (rIDPB.phctype).
<i>Bad Data</i> =	The process number of the call activity check pid (rIDPB.cac_pid.procno).
<i>Good Data</i> =	The uniqueness of the call activity check pid (rIDPB.cac_pid.uniq).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 19 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	An invalid linkage is found between a B-channel rIDPB tuple used for the ODB feature and an rIPTSB tuple.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	The ODB call will be torn down and related data recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The invalid linkage between the B-channel DPB and the PTSB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DPB key.
<i>Dump 1</i> =	The rIDPB tuple.
<i>Dump 2</i> =	The rIPTSB tuple.

Table 20 Error Code: DSL_RATE

Error Code:	DSL_RATE
Error Description:	Data rate in DPB differs from ODD data rate.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Process owning DPB is purged if any D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Data rate found in DPB.
<i>Good Data</i> =	Data rate according to ODD.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	NA.

Table 21 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The logical PH number and physical PH channel data in a B-channel rIDPB tuple used for ODB did not match the data contained in the corresponding rIODPSOP static tuple. For B-channel rIDPB tuples used for ODB calls, the ph_number and ph_cn attributes must be represented in rIODPSOP.ph_number and rIODPSOP.ph_cn or ph_cn2 where, for the associated D-channel, (key) rIDPB.port = (key) rIODPSOP.port and (key) rIDPSOP.party = (rIDPB.X >> rIODPSOP.party & (DMUCHAR)1) where X = b1pty_resv or b2pty_resv in the D-channel rIDPB tuple associated with the B-channel rIDPB tuple under audit.
Possible Error Effect:	An unauthorized subscriber may be using the ODB feature.
Corrective Action Taken:	The ODB call process is purged and the ODB call owning the B-channel is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-channel rIDPB tuple under audit.
<i>Bad Data</i> =	The PH number the rIDPB tuple (ph_number)
<i>Good Data</i> =	The PH number found in the rIODPSOP static tuple (ph_number)
<i>Logical Index</i> =	The key of the rIDPB tuple under audit (port)
<i>Dump 1</i> =	The B-channel rIDPB tuple under audit.
<i>Dump 2</i> =	The associated D-channel rIDPB tuple for the DSL.
<i>Dump 3</i> =	The associated rIODPSOP static tuple.

Table 22 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rIDPB dynamic tuple without a corresponding static tuple where: - the DPB does not have the static rIDSLEQUIP tuple with rIDPB.port = rIDSLEQUIP.dsl or - there is no user subscribed to the on-demand-packet feature with rIDPB.port = rIODPSOP.port
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key value.
<i>Dump</i> =	The rIDPB tuple.

Table 23 Error Code: GPORT

Error Code:	GPORT

Error Description:	The specified B-channel port is not populated or key or B-channel PORTLA is wrong.
Possible Error Effect:	Application code may assert when accessing the data and B-channel cannot be used.
Corrective Action Taken:	PORTLA or FCPORTTYPE audit will be scheduled as related audit to detect this error and correct it.
Manual Action Required:	If FC_PORTTYPE is missing, then manual action is required to insert the B-channel FC_PORTTYPE tuple. (Look good data in the dump.)
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The key of B-PORTLA in the system (maybe 0xffff).
<i>Good Data =</i>	The key of B-PORTLA which should be in the system.
<i>Logical Index =</i>	The rIDPB key - port name.
<i>Dump =</i>	The rIDPB tuple.

Table 24 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	Based on the bit maps rIDSLEQUIP.agipty1 and rIDPB.b#pty_resv the associated group information rIDPB.agi1_bchan is set to the incorrect value. The range of values is FCNO_BCHAN, FCB1_CHAN, and FCB2_CHAN.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	rIDPB.agi1_bchan is set to the correct value listed below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	rIDPB.agi1_bchan.
<i>Good Data =</i>	Correct value of rIDPB.agi1_bchan.
<i>Logical Index =</i>	The rIDPB key (rIDPB.port).
<i>Dump =</i>	The rIDPB tuple.

Table 25 Error Code: GRP2INFO

Error Code:	GRP2INFO
Error Description:	Based on the bit maps rIDSLEQUIP.agipty2 and rIDPB.b#pty_resv the associated group information rIDPB.agi2_bchan is set to the incorrect value. The range of values is FCNO_BCHAN, FCB1_CHAN, and FCB2_CHAN.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	rIDPB.agi2_bchan is set to the correct value listed below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	rIDPB.agi2_bchan.
<i>Good Data =</i>	Correct value of rIDPB.agi2_bchan.
<i>Logical Index =</i>	The rIDPB key (rIDPB.port).
<i>Dump =</i>	The rIDPB tuple.

Table 26 Error Code: IL_PTYE

Error Code:	IL_PTYE
Error Description:	The lp_type in rIFC_PORTTYP does not indicate that the port B-channel rIDPB is associated with a B-channel port.
Possible Error Effect:	Wasted resources since this tuple cannot be used for anything.
Corrective Action Taken:	The B-channel rIDPB tuple and related data, if any, will be recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The lp_type in rIFCPORTTYP.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the DPB tuple.
<i>Dump =</i>	The DPB tuple.

Table 27 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	EOC/TMC ports on an IDCU TR303 have a invalid link between the rIDPB and the rRTLb tuple.
Possible Error Effect:	Tasks for this RT may be lost and not completed.
Corrective Action Taken:	The EOC/TMC port will be recovered and the link restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The contents of the rIDPB tuple.

Table 28 Error Code: INV

Error Code:	INV
Error Description:	Either the interf_id or the prigrp field of the rIDPB tuple did not match the same field from the static rIDCHNPRI tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Both the interf_id and the prigrp fields of the rIDPB tuple will be corrected with the values found in the same fields of the static rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The interf_id field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDCHNPRI tuple.
<i>Good Data</i> =	The interf_id field of the rIDCHNPRI tuple.
<i>Logical Index</i> =	The prigrp field of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 29 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	BRI class of service in DPB is not consistent with DSLEQUIP.
Possible Error Effect:	This DSL may have signaling and binding problems.
Corrective Action Taken:	briclass in DPB is correctly set to be consistent with DSLEQUIP and the port is put on the to be idled list (TMLTTBI) to be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	briclass in DPB.
<i>Good Data</i> =	briclass in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 30 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The default service TEDB should link to the DSL which has subscribed default service feature but it links to other DSL.
Possible Error Effect:	No default service is provided in this DSL.
Corrective Action Taken:	The DSL will be recovered and all the TEDBs will be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The DSL which TEDB links to.
<i>Good Data</i> =	The DSL which TEDB should link to.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 31 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	Non-FRBC trunks must have a null SVC job identifier and null SVC job process ID this trunk has a non-null SVC attribute.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job identifier and SVC job process ID is nulled out.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job identifier of the rIDPB tuple (rIDPB.svcjob_id).
<i>Good Data</i> =	The SVC job process ID of the rIDPB tuple (rIDPB.svcjob_pid).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 32 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The audit received an unrecognized or invalid return from TMDpbprt().
Possible Error Effect:	The rIDPB tuple cannot be audited.
Corrective Action Taken:	The audit skips the rIDPB tuple since it cannot access any related data required for auditing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIFC_PORTTYP static tuple passed to TMDpbprt().
<i>Bad Data</i> =	The return received from TMDpbprt().
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The rIFC_PORTTYP static tuple.

Table 33 Error Code: L2MON

Error Code:	L2MON
Error Description:	The port is requested to be monitored for protocol but the data used is inconsistent.
Possible Error Effect:	Switch may waste resources to monitor the unrequested port.
Corrective Action Taken:	The port is marked as not being monitored.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The pmon field in DPB.
<i>Good Data</i> =	DBNO(0).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 34 Error Code: LEVEL_1

Error Code:	LEVEL_1
Error Description:	Level two profile in the ODD does not match profile in the DPB.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Process owning DPB is purged if any D-port is turned over to terminal maintenance for recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Level 2 profile found in DPB.
<i>Good Data</i> =	Level 2 profile according to ODD.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	NA.

Table 35 Error Code: LEVEL_3

Error Code:	LEVEL_3
Error Description:	The level three state in the DPB did not match the level three state derived from the circuit type information in the PORTTYP and DSLEQUIP tuple.
Possible Error Effect:	Potential for breaking the D-channel signaling path.
Corrective Action Taken:	Level three state is updated to match the static ODD data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Level three state from the DPB.
<i>Good Data</i> =	Level three state derived from the ODD.
<i>Logical Index</i> =	PORT number of D-port.

<i>Dump</i> =	DPB is dumped.

Table 36 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	DSL should have default service feature but there is no default service TEDB linked to DPB or TEDB key is wrong.
Possible Error Effect:	No default service is provided in this DSL.
Corrective Action Taken:	Print the error and request related audits to insert TEDB or correct the TEDB errors.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The TEDB key in the system.
<i>Good Data</i> =	The TEDB key which should be.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 37 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The servclass and lp_type fields of the static rIFC_PORTTYPE tuple indicates that this D channel is not used for NFAS and DCBU PRI D channel(AUportbuf.servclass not != FCEDSL and AUportbuf.lp_type != PCTDCHAN). And yet audit found that one of the three fields related to NFAS and DCBU D channel, namely, interf_id, prigrp, and pri_type has a non null value.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	All the three fields will be set to the null value (interf_id->PSNO_INTID, prigrp->TMNULLPRI and pri_type->TMNULL_TYPE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The interf_id field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 38 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	For an NFAS/DCBU PRI calls, the linkage field to the rIDALB tuple dpdao must be null as no DALBs used in such type of PRI calls is linked to DPBs. Audit found this field to be non null.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Audit will force recovery of this rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The st_headk field of the dpdao field of the rIDPB tuple.
<i>Bad Data</i> =	The st_tailk field of the dpdao field of the rIDPB tuple.
<i>Good Data</i> =	The prigrp field of the rIDPB tuple.
<i>Logical Index</i> =	The port field, key, of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 39 Error Code: NOACCBLK

Error Code:	NOACCBLK
Error Description:	For FRBC calls, when either the SVC job ID or the SVC job PID is non-null, future SVC calls must be blocked. If the block-call bit is set to 0 (DBNO), calls are not blocked.
Possible Error Effect:	SVC calls are attempted and fail later.
Corrective Action Taken:	Audit will set the block-call bit to 1 (DBYES).
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The SVC job identifier (rIDPB.svcjob_id).
<i>Bad Data</i> =	The SVC job process identifier (rIDPB.svcjob_pid.procno).
<i>Good Data</i> =	The SVC job process ID's unique bit (rIDPB.svcjob_pid.uniq).
<i>Logical Index</i> =	The port field, key, of the rIDPB tuple.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 40 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	While looking for a process on the port associated with the rIDPB tuple during selective initialization, the DA function call returned an error saying that database corruption exists. The audit error report dumps rIORTLA because that is what it has access to.
Possible Error Effect:	The port would normally be taken down unless a barge-in call pickup process exists for this port. Since database error was returned, the port will be taken down.
Corrective Action Taken:	The tuple is quarantined, so that the audit forces recovery of the rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The pointer to the rIORTLA tuple.
<i>Bad Data</i> =	The value returned from the DA function call.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIORTLA tuple's key, port field.
<i>Dump</i> =	The rIORTLA tuple being audited.

Table 41 Error Code: NOTIDLE2

Error Code:	NOTIDLE2
Error Description:	A rIDPB tuple not currently used for ODB was found to have its logical PH number (ph_number) and physical channel number (ph_cn) attributes set to invalid values. For D-channel rIDPB tuples, or rIDPB B-channel tuples which are not associated with B-channels used for the ODB feature, the PH and channel attributes are unused and should be set to idle values (PSPHNULL and zero respectively).
Possible Error Effect:	Call processing will never be able to use the associated B-channel for ODB since the PH and channel attributes make it appear that the B-channel already has a PH and channel assigned and in use.
Corrective Action Taken:	The ph_number and ph_cn attributes are recovered to their idle values (PSPHNULL and 0 respectively).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The contents of the ph_number and ph_cn attributes. (rIDPB.ph_number << 4 rIDPB.ph_cn)
<i>Good Data</i> =	The null values for ph_number and ph_cn (PSPHNULL << 4 (DMPHCN) 0).
<i>Logical Index</i> =	The rIDPB tuple's key (port).
<i>Dump</i> =	The rIDPB tuple.

Table 42 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The PH channel type indicates that this is a ADT, packet pipe or speech handler trunk so the semantic linkage to the rIPTSB (islu_ptsb or psiu_ptsb) should be zero, but instead is non-zero.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	The semantic linkage to the rIPTSB is nulled out (rIDPB.islu_ptsb and rIDPB.psiu_ptsb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The islu_ptsb field of the rIDPB tuple.
<i>Bad Data</i> =	The psiu_ptsb field of the rIDPB tuple.
<i>Good Data</i> =	The phctyp field of the rIDPB tuple.

<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 43 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The RLPORTTYP relation indicates that the DSL for this port must be equipped, but the rIDSLEQUIP tuple is missing from the static ODD. If the <i>Good Data</i> is 2, the DPB audit was unable to obtain the static relation tuple (rIPP_MEM) needed to audit the dynamic rIDPB tuple for the CDMA packet pipe either there is no packet pipe member tuple with the same port as the DPB or PP_MEM key is wrong.
Possible Error Effect:	If the <i>Good Data</i> is 1, the audit cannot verify any data in the DPB. If it is bad, the audit has no way to fix it. If the <i>Good Data</i> is 2, packet pipe members cannot be located for packet pipe call processing.
Corrective Action Taken:	None.
Manual Action Required:	If the <i>Good Data</i> is 1, the craft will need to use ODBE to put the missing DSLEQUIP tuple back into the ODD, since ODBE does not permit removing the PORTTYP tuple under these circumstances. If the <i>GOOD DATA</i> is 2, the craft will need to use ODBE to put the missing PP_MEM tuple into the ODD. Find out the value of the port.member from the dump of the rIDPB tuple. The port.module will be the processor receiving the audit. Using these as the key, insert the static rIPP_MEM tuple. For the other fields of the tuple, refer to the terminal maintenance and trigger area.
Dump Description:	
<i>Error Address</i> =	If the <i>Good Data</i> is 1, the address of the rIDSLEQUIP tuple. If the <i>Good Data</i> is 2, the address of the rIDPB tuple.
<i>Bad Data</i> =	The logical port type (rIFC_PORTTYP.lp_type).
<i>Good Data</i> =	Valid values are:
	1 = Could not read the rIDSLEQUIP tuple.
	2 = Could not read the rIPP_MEM tuple.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump 1</i> =	The contents of the rIDPB tuple.
<i>Dump 2</i> =	The contents of the rIFC_PORTTYP tuple.

Table 44 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An FRBC trunk has a non-null SVC job identifier, but it is not within the range of valid values for such a field.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job identifier is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job identifier of the rIDPB tuple (rIDPB.svcjob_id).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 45 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	DPB tuples that are not used by IDCU or certain types of CDMA must be linked to TEDB, DALB, and/or TEILA tuples. This DPB tuple is not correctly linked to one of them. If the <i>Good Data</i> is 1, it is incorrectly linked to TEDB 2, DALB or 3, TEILA.

Possible Error Effect:	Problem in call processing.
Corrective Action Taken:	The rIDPB tuple is quarantined so that the tuple is recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The linkage, either TEDB, DALB, or TEILA. (Refer to <i>Good Data</i> .)
<i>Good Data =</i>	Valid values are: 1 = The bad linkage is to a TEDB. 2 = The bad linkage is to a DALB. 3 = The bad linkage is to a TEILA.
<i>Logical Index =</i>	The rIDPB tuple's key, port field.
<i>Dump =</i>	The rIDPB tuple being audited.

Table 46 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	The PH channel circuit type stored in an rIDPB tuple, which is associated with a B-channel port used for the ODB feature, was found inconsistent from the type derived from the B-channel management data stored in static ODD (rIFC_PORTTYPE).
Possible Error Effect:	The PORT could be OOS.
Corrective Action Taken:	The "phctype" field is updated in the DPB to agree with the B-channel management data stored in static ODD (rIFC_PORTTYPE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the DPB tuple.
<i>Bad Data =</i>	Value of the "phctype" field in the DPB.
<i>Good Data =</i>	Value of "phctyp" computed from the ODD.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	The DPB tuple.

Table 47 Error Code: PHDSLGP

Error Code:	PHDSLGP
Error Description:	There are currently two ODB calls up on this DSL. However, the logical PH and physical PH channel numbers stored in the rIDPB tuples for each B-channel are identical. When ODB calls are up on each B-channel of a DSL, the ph_number and ph_cn attributes in each ODB B-channel rIDPB tuple must be unique to each other. Since it is physically impossible to set up ODB calls on a DSL under these conditions, this error is an indication that the ph_number and ph_cn attributes in one or both rIDPB tuples became corrupted after both ODB calls were set up on the DSL. The occurrence of this error is an indication that the PH and channel data does not reflect the actual PH and channel(s) being used for ODB call activities.
Possible Error Effect:	The ODB connection(s) on this DSL cannot be cleanly torn down. Attempts by call processing to do so may result in asserts and/or audit errors.
Corrective Action Taken:	Both B-channel rIORTLA tuples are quarantined for recovery (rIORTLA.quarstate marked AUAUSET). The B-channel management data in the D-channel rIDPB tuple (b1use, b2use, b1pty_resv and b2pty_resv) are marked to idle states (DMIDDLE_USE and 0 respectively). Lastly, the logical PH number (ph_number) and physical channel number (ph_cn) attributes in the B1 and B2 rIDPB tuples are marked to an idle state (PSPHNULL and 0 respectively). Any ODB call processes up on the DSL will then be purged and any other related call data will be idled out by related audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the B-channel rIDPB ODB B-channel tuple under audit.

<i>Bad Data</i> =	The logical PH number (ph_number) in the B-channel rIDPB ODB B-channel tuple under audit.
<i>Good Data</i> =	The physical PH channel number (ph_cn) in the B-channel rIDPB ODB B-channel tuple under audit.
<i>Logical Index</i> =	The key of the B-channel rIDPB ODB B-channel tuple (port).
<i>Dump 1</i> =	The contents of the B-channel rIDPB ODB B-channel tuple under audit.
<i>Dump 2</i> =	The contents of the "other" B-channel rIDPB ODB B-channel tuple.
<i>Dump 3</i> =	The contents of the D-channel rIDPB tuple on the DSL.

Table 48 Error Code: PHNUNEQUIP

Error Code:	PHNUNEQUIP
Error Description:	The dsl_group and/or member obtain from the static tuple rIDSLEQUIP do not match the ph_number and/or ph_cn stored in the rIDPB tuple.
Possible Error Effect:	Problems in Call Processing.
Corrective Action Taken:	The rIDPB tuple is recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	PH number (rIDPB.ph_number).
<i>Good Data</i> =	DSL group (rIDSLEQUIP.dsl_group).
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump 1</i> =	The rIDPB tuple being audited.
<i>Dump 2</i> =	The rIDSLEQUIP tuple being audited.

Table 49 Error Code: PLE_STATE

Error Code:	PLE_STATE
Error Description:	The plestate attribute is incorrect.
Possible Error Effect:	D-channel will be idled.
Corrective Action Taken:	The rIDPB tuple will be initialized and made idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDPB.
<i>Bad Data</i> =	Incorrect plestate.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIDPB.
<i>Dump</i> =	rIDPB.

Table 50 Error Code: PORT

Error Code:	PORT
Error Description:	Using port field of the rIDPB tuple as the member field and GI_pcrd as the module field of the dport key field of rIDCHNPRI tuple, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it is inserted by office personnel. Refer to the Manual Action Required field as to how to insert the tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	Both the interf_id and the prigrp fields of the rIDPB tuple will be reset to their null values and audit will return.
Manual Action Required:	Using port field of the rIDPB tuple as the member field and GI_pcrd as the module field of the dport key field of rIDCHNPRI tuple, insert the static rIDCHNPRI tuple. As for the values of the other fields, refer to the terminal maintenance and triggers area.
Dump Description:	
<i>Error Address</i> =	The prigrp field of the rIDPB tuple.
<i>Bad Data</i> =	The interf_id field of the rIDPB tuple.
<i>Good Data</i> =	The pri_type field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 51 Error Code: PROC_LOST

Error Code:	PROC_LOST

Error Description:	An FRBC trunk has a non-null SVC job process ID, but the process is non-existent.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job process ID is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job process ID's process number (rIDPB.svcjob_pid.procno).
<i>Good Data</i> =	The SVC job process ID's uniqueness (rIDPB.svcjob_pid.uniq).
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 52 Error Code: PROC_TO

Error Code:	PROC_TO
Error Description:	The process owning B-channel port does not link properly with the B-channel PORTLA.
Possible Error Effect:	Application code may assert when accessing the data and B-channel cannot be used.
Corrective Action Taken:	The PORTLA will be quarantined and process will be purged.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The process number of PCBLA.
<i>Good Data</i> =	The expected process number from PORTLA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 53 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The level 3 profile obtained from rIFC_PORTTYP and rIODPSOP static ODD did not match the level 3 profile stored in a B-channel rIDPB tuple being used for the ODB feature.
Possible Error Effect:	The B-channel signaling path may be corrupted.
Corrective Action Taken:	The level 3 data will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the B-DPB tuple.
<i>Bad Data</i> =	The level 3 profile found in the B-DPB tuple.
<i>Good Data</i> =	The level 3 profile derived from static data.
<i>Logical Index</i> =	The key of the B-DPB tuple.
<i>Dump</i> =	The B-channel DPB tuple.

Table 54 Error Code: PTR3

Error Code:	PTR3
Error Description:	According to the D-channel rIDPB tuple for a DSL with two ODB B-channels assigned, both B-channels currently have ODB calls up, however, only one rIDPB B-channel tuple could be accessed by the audit. For each B-channel "use" field in the D-channel rIDPB tuple which is marked to indicate ODB is in use on the respective B-channel (b1use and/or b2use = DMOB_USE) there should exist a corresponding ODB B-channel rIDPB tuple.
Possible Error Effect:	If the B-channel is assigned the ODB feature, then ODB calls can never be originated on this B-channel.
Corrective Action Taken:	The B-channel management data in the associated D-channel rIDPB tuple which corresponds to the B-channel under audit will be recovered to an idle state (b1use or b2use set to DMIDLE_USE and b1pty_resv or b2pty_resv set to 0 depending on which B-channel was marked as using ODB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the D-channel rIDPB tuple for the DSL.
<i>Bad Data</i> =	Informational: The b1use field in the D-channel rIDPB tuple.
<i>Good Data</i> =	Informational: The b2use field in the D-channel rIDPB tuple.

<i>Logical Index</i> =	The key of the D-channel rIDPB tuple (port)
<i>Dump</i> =	The D-channel rIDPB tuple.

Table 55 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The routine port conditioning table indicates that a call path should be up (TMportcond[rIORTLA.list where rIDPB.port = rIDPB.port].ispathup = TMPCYES) but the rIDPB tuple is not linked to an rIPTSB tuple (rIDPB.islu_ptsb = 0 and rIDPB.psiu_ptsb = 0) and no process owns the port (rIORTLA.proc_cplr.st_headk = 0 where rIORTLA.port = rIDPB.port). The rIDPB tuple should be linked to an rIPTSB tuple if the routine port conditioning table indicates a path should be up for a given port list except if the port is owned by a process.
Possible Error Effect:	The time slots associated with the rIPTSB tuple will be lost to the system, thus reducing the total number of available time slots.
Corrective Action Taken:	The rIDPB tuple is recovered to an idle state, and the rIORTLA tuple is placed on the to-be-idled list (rIORTLA.list = TMLTTBI), and the terminal maintenance unloader is awakened to process the rIORTLA tuple so that it is eventually set to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The list the port associated with the RLDPB is on (rIORTLA.list where rIORTLA.port = rIDPB.port).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key (rIDPB.port) of the rIDPB tuple.
<i>Dump 1</i> =	The contents of the rIDPB tuple.
<i>Dump 2</i> =	The contents of the associated rIORTLA tuple.

Table 56 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The port is not on a recovery list, so the quarantine state should not be in a race condition, but it is.
Possible Error Effect:	Lost resources since this port is out-of-service.
Corrective Action Taken:	The rIDPB tuple is quarantined so it can be recovered by audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	The quarantine state (rIDPB.quarstat).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	NA.

Table 57 Error Code: QUE

Error Code:	QUE
Error Description:	An FRBC trunk has a non-null SVC job identifier, but the ISPA job is no longer on the queue.
Possible Error Effect:	Application code may assert when accessing the data.
Corrective Action Taken:	The SVC job identifier is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical port type of the rIDPB tuple (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	The SVC job identifier of the rIDPB tuple (rIDPB.svcjob_id).
<i>Good Data</i> =	The return value from the function call, TMisqjobid.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 58 Error Code: R_COUNTS

Error Code:	R_COUNTS

Error Description:	The number of TEILA linked to DPB is greater than what user subscribes (specify in DSLEQUIP).
Possible Error Effect:	The error may cause binding and signaling problem.
Corrective Action Taken:	The DSL will be reinitialized and recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	Number of TEILA linked to DPB.
<i>Good Data</i> =	Number of TEILA allowed in DSLEQUIP.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 59 Error Code: SGNOOD

Error Code:	SGNOOD
Error Description:	Using prigrp field of the rIDPB tuple as the key field of rIPRIGRP tuple, audit tried to read the static rIPRIGRP tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it is inserted by office personnel. Refer to the manual action required field as to how to insert the tuple.
Possible Error Effect:	Possible failure of the NFAS/DCBU PRI calls.
Corrective Action Taken:	The pri_type field of the rIDPB tuple will be reset to its null value (TMNULL TYPE) and audit will retun.
Manual Action Required:	Find out the value of the prigrp field from the dump of the rIDPB tuple. Use it as the key field insert the static rIPRIGRP tuple. As for the other fields of the tuple, refer to the terminal maintenance and trigger area.
Dump Description:	
<i>Error Address</i> =	The pri_type field of the rIDPB tuple.
<i>Bad Data</i> =	The prigrp field of the rIDPB tuple.
<i>Good Data</i> =	The interf_id field of the rIDPB tuple.
<i>Logical Index</i> =	The rIDPB tuple's key, port field.
<i>Dump</i> =	The rIDPB tuple being audited.

Table 60 Error Code: TEDBFLAG

Error Code:	TEDBFLAG
Error Description:	The tedb_flag attribute in the DPB tuple is incorrectly set.
Possible Error Effect:	An incorrect linkage may exist between the DPB and either a TEDB tuple or a DALB tuple. This may effect call processing.
Corrective Action Taken:	The tedb_flag will be correctly set according to the servclass found at FC_PORTTYPE. If it is ISDN or ISDN attendant DSL, then tedb_flag will be set to DBYES. If it is an EOC/TMC port, then the tedb_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Valid values are: 0 = If the tedb_flag should be set to DBYES. 1 = If the tedb_flag should be set to DBNO.
<i>Good Data</i> =	Valid values are: 0 = If not ISDN and ISDN attendant or if EOC/TMC port. 1 = If it is ISDN or ISDN attendant.
<i>Logical Index</i> =	The rIDPB key (rIDPB.port).
<i>Dump</i> =	RLDPB.

Table 61 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	An invalid linkage is found between the rIDPB tuple and a rTEDB tuple.

Possible Error Effect:	Switch cannot communicate with customer premise equipment. EOC/TMC port on IDCU TR303 have lost communication with the switch.
Corrective Action Taken:	Port will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Linkage value from rIDPB tuple to rITEDB (rIDPB.tedb_own.st_headk).
<i>Good Data =</i>	0. There should not be a linkage between the tuples.
<i>Logical Index =</i>	The rIDPB key.
<i>Dump =</i>	The contents of rIDPB tuple.

Table 62 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The fixed terminal endpoint identifier (FTEI) flag indicator in the rIDPB tuple is incorrect. It does not match the same indicator in the corresponding rIDSLEQUIP tuple. This flag is used by call processing to determine if there is a FTEI resource on this digital subscriber line (DSL).
Possible Error Effect:	FTEI resources on this DSL may not function correctly on this DSL.
Corrective Action Taken:	The flag indicator (pfitei) is recovered to the value specified in the corresponding rIDSLEQUIP flag.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value of pfitei in the rIDPB before correction.
<i>Good Data =</i>	The value of pfitei in the corresponding rIDSLEQUIP tuple.
<i>Logical Index =</i>	The rIDPB key.
<i>Dump 1 =</i>	The contents of rIDPB tuple.
<i>Dump 2 =</i>	The contents of rIDSLEQUIP tuple.

Table 63 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	For non-ISDN line. TEILA should not link to DPB.
Possible Error Effect:	Service cannot be received from this port.
Corrective Action Taken:	The port will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDPB tuple.
<i>Bad Data =</i>	The port which TEILA links to.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The rIDPB key - port name.
<i>Dump =</i>	The rIDPB tuple.

Table 64 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	The tpwait field associated with this DPB is not set correctly according to data in the PORTLA.
Possible Error Effect:	"tpwait" should be on for EOC/TMC ports plus X.75, X.75', FRBC and IP trunks when ISPA terminal process (TP) is waiting for a level 2 setup message, and off for all other ports.
Corrective Action Taken:	The tpwait field will be updated according to the information present in the PORTLA if this is the only error detected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Service class of the dpb tuple (rIFC_PORTTYP.servclass).
<i>Bad Data =</i>	Value of "tpwait" field in the DPB.
<i>Good Data =</i>	"tpwait" value determined from PORTLA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	The DPB tuple.
<i>Dump 2 =</i>	The PORTLA tuple.

Table 65 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	A broken linkage is found when traversing the rTEILA tuple through this rIDPB tuple. A broken linkage to rIRTEILA tuple is found on an IDCU TR303 EOC/TMC port.
Possible Error Effect:	Application codes may assert and some of CPEs may not receive any service.
Corrective Action Taken:	The DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDPB key - port name.
<i>Dump</i> =	The rIDPB tuple.

Table 66 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PH channel type stored in the DPB does not agree which that which was computed from the ODD.
Possible Error Effect:	The PORT could be OOS.
Corrective Action Taken:	The "phctyp" field is updated in the DPB to agree with the ODD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the DPB tuple.
<i>Bad Data</i> =	Value of the "phctyp" field in the DPB.
<i>Good Data</i> =	Value of "phctyp" computer from the ODD.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The rIDPB tuple.

Table 67 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	rIDPB.agi1_bchan and rIDPB.agi2_bchan are equal. This means that both groups are using the same B-channel, which is not allowed.
Possible Error Effect:	CSV, CSD, or ODB calls may be illegally blocked or allowed.
Corrective Action Taken:	None.
Manual Action Required:	Check assignment of associated groups and multi-Line hunt groups (MLHG) on RC/V View 23.2 (DIGITAL SUBSCRIBER LINE) and 23.3 (ASSOCIATED GROUP). Check for overlapping groups, and MLHGs which occur in both groups.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	rIDPB.agi1_bchan
<i>Good Data</i> =	rIDPB.agi2_bchan
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The rIDPB tuple.

Table 68 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	All rIDPB tuples for EOC/TMC ports on an IDCU TR303 should have a linkage to an rIRTEILA tuple.
Possible Error Effect:	Message flow between the SM and IDCU RT303 is interrupted.
Corrective Action Taken:	EOC/TMC port is recovered and communication restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDPB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of DPB (rIDPB.port).
<i>Dump</i> =	The contents of the rIDPB tuple.

DPCDPCI

Software Release: 5E11 and later

1. DESTINATION POINT CODE TO DESTINATION POINT CODE INDEX (DPCDPCI) AUDIT

DPCDPCI audit uses the static relation RLDPC_DPCI to verify the existence of the dynamic RLDPC_STAT relation (destination point code status). The missing rIDPC_STAT tuple will be inserted in the system by this audit.

2. DATA STRUCTURES

For details on the fields contained in the RLDPCDPCI relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_DPC

Error Code:	BAD_DPC
Error Description:	The rIDPC_STAT.dpc does not equal rIDPC_DPCI.dpc for the destination point code (DPC) index (rIDPC_STAT.dpci).
Possible Error Effect:	A copy of the DPC information is kept in the protocol handler (PH). If the PH information is not correct, the routing function in the PH cannot find the correct destination for message signaling units (MSUs).
Corrective Action Taken:	The audit sets rIDPC_STAT.dpc equal to rIDPC_DPCI.dpc and sends a message to the signaling, transfer and route management (STRM) process to order the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.dpc.
<i>Good Data =</i>	rIDPC_DPCI.dpc (key of rIDPC_DPCI).
<i>Logical Index =</i>	rIDPC_STAT.dpci (key of rIDPC_STAT).
<i>Dump =</i>	NA.

Table 2 Error Code: DPC_BAD

Error Code:	DPC_BAD
Error Description:	The CCaccpcidx_dpc[message transfer part (MTP) for accounting. CCaccpcidx_dpc[rIDPC_DPCI.accpcindx] does not equal rIDPC_DPCI.dpc from a tuple whose key is rIDPC_DPCI.dpcindx.
Possible Error Effect:	Incorrect translation occur while attempting to populate the counts message sent to the administration module (AM). All counts are lost for that index value due to the incorrect translations.
Corrective Action Taken:	The audit sets CCaccpcidx_dpc[rIDPC_DPCI.accpcindx] equal to rIDPC_DPCI.dpc.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_DPCI.accpcindx.
<i>Bad Data =</i>	CCaccpcidx_dpc[rIDPC_DPCI.accpcindx].
<i>Good Data =</i>	rIDPC_DPCI.dpc.
<i>Logical Index =</i>	rIDPC_DPCI.dpcindx.
<i>Dump =</i>	NA.

Table 3 Error Code: DUPTUP

Error Code:	DUPTUP
Error Description:	The audit tries to insert an rIDPC_STAT tuple which it believes to be missing. The insert fails because the position of the tuple with the missing key is occupied. This indicates that the key of the tuple which is believed to be missing is incorrect.
Possible Error Effect:	A copy of the DPC information is kept in the PH. If the information is not correct, the routing function in the PH cannot find the correct destination for message signaling units.
Corrective Action Taken:	The DPCSTAT audit deletes the incorrect rIDPC_DPCI tuple which occupies the position of the missing rIDPC_DPCI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rIDPC_STAT.dpci.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 4 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is no tuple in relation RLDPC_STAT with a key equal to rIDPC_DPCI.dpcindx.
Possible Error Effect:	A copy of the DPC information is kept in the PH. The information includes the destination point code index (DPCI) of the DPC. If the information is not correct, the routing function in the PH cannot find the correct destination for message signaling units.
Corrective Action Taken:	The audit constructs, inserts and links the missing rIDPC_STAT tuple to the rILS_STAT and rIRCGTMR idle lists and sends a message to the STRM process to order the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	rIDPC_DPCI.dpcindx.
<i>Logical Index =</i>	rIDPC_DPCI.dpc.
<i>Dump =</i>	NA.

Table 5 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	An entry in CCacc[codx_dpc[] is not in use because its index does not equal rIDPC_DPCI.accpcindx from any rIDPC_DPCI tuple. The entry should be set to its null value ((DMULONG)CCEMPTYTUP) but is not.
Possible Error Effect:	An incorrect value in the table could be translated to a valid DPC.
Corrective Action Taken:	The audit sets the CCaccpcidx_dpc[] entry to (DMULONG)CCEMPTYTUP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CCaccpcidx_dpc[index].
<i>Good Data =</i>	(DMULONG)CCEMPTYTUP.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

Table 6 Error Code: OPCODE

Error Code:	OPCODE
Error Description:	The CCaccpcidx_dpc[] has the point codes related to the MTP for accounting. CCaccpcidx_dpc[1] should equal the office point code (Gloffpc) but does not.
Possible Error Effect:	Incorrect translation occurs while attempting to populate the counts message sent to the AM. All counts are lost for that index value due to the incorrect translations.
Corrective Action Taken:	The audit sets CCaccpcidx_dpc[1] equal to Gloffpc.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CCaccpcidx_dpc[1].
<i>Good Data =</i>	Gloffpc (office point code).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 7 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The audit tries to find a tuple in relation RLDPC_INFO with a key equal to rIDPC_DPCI.dpcindx. The tuple in relation RLDPC_INFO does not exist.
Possible Error Effect:	The DPC information in the PH will not be initialized correctly. If the DPC information is not correct, the routing function in the PH cannot find the correct destination for MSUs.
Corrective Action Taken:	None.
Manual Action Required:	Insert a tuple into RLDPC_INFO such that the rIDPC_INFO.dpcindx equals rIDPC_DPCI.dpcindx for the DPC (rIDPC_DPCI.dpc). Consult the population rules for RLDPC_INFO and RLDPC_DPCI.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rIDPC_DPCI.dpcindx.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIDPC_DPCI.dpc.
<i>Dump =</i>	NA.

DPCSTAT

Software Release: 5E11 and later

1. DESTINATION POINT CODE STATUS (DPCSTAT) AUDIT

DPCSTAT audit step through every tuple in the dynamic relation RLDPC_STAT (destination point code status). For every tuple, it checks the existence of the corresponding static relation RLDPC_DPCI (destination point code to destination point code index). The audit also checks the linkage between RLDPC_STAT and RLLS_STAT (link set status) and RLCLS_STAT (combined link set status). It also checks the semantic data in the RLDPC_STAT against RLS7RDB (CCS7 routing data block), RLLS_STAT and RLLNK_STAT (link status). This audit also makes sure that eight rIS7RDB tuples existing for each destination point code (DPC). Any missing rIS7RDB tuple will be inserted.

2. DATA STRUCTURES

For details on the fields contained in the RLDPC_STAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACC_BAD

Error Code:	ACC_BAD
Error Description:	CCsmdpcls_ht[] is the DPC to link set routing table which is built from RLDPC_STAT. CCsmdpcls_ht[rIDPC_STAT.dpci].access does not equal rIDPC_STAT.acc for a DPC.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit sets CCsmdpcls_ht[rIDPC_STAT.dpci].access equal to rIDPC_STAT.acc for the DPC index (DPCI).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCsmdpcls_ht[rIDPC_STAT.dpci].
<i>Bad Data =</i>	CCsmdpcls_ht[rIDPC_STAT.dpci].access.
<i>Good Data =</i>	rIDPC_STAT.acc.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 2 Error Code: APC_BAD

Error Code:	APC_BAD
Error Description:	The error is due to one of the following: 1 = NMpcc7buf[rIDPC_STAT.dpci].apc is DBYES and rIDPC_STAT.apcindx is 0 for a DPCI. NMpcc7buf[rIDPC_STAT.dpci].apc should be DBNO for the same DPCI, but is not. 2 = NMpcc7buf[rIDPC_STAT.dpci].apc is DBNO and rIDPC_STAT.apcindx is

	not 0 for a DPCI. NMpcc7buff[rIDPC_STAT.dpci].apc should be DBYES for the same DPCI, but is not.
Possible Error Effect:	Corresponds to the Error Description as follows: 1 = Network management (NM) may report that a point code is an adjacent point code (APC) when it is not. Routing through this APC fails. 2 = NM may report that a point code is not an APC when it is. This APC is not used for routing.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = Audit sets NMpcc7buff[rIDPC_STAT.dpci].apc to DBNO. 2 = Audit sets NMpcc7buff[rIDPC_STAT.dpci].apc to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1) or (2) indicates Error Description, Error Effect and Corrective Action.
<i>Bad Data =</i>	NMpcc7buff[rIDPC_STAT.dpci].apc.
<i>Good Data =</i>	rIDPC_STAT.apc.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 3 Error Code: ARRAY_ENTRY_BAD

Error Code:	ARRAY_ENTRY_BAD
Error Description:	CCapci2dpci[] is the adjacent point code index (APCI) to DPCI table. CCsmdpcls_ht[] is the DPC to link set routing table. CCapci2dpci[index] must equal CCsmdpcls_ht[CCapci2dpci[index].dpci].apcindx, but does not.
Possible Error Effect:	Signaling traffic and route management (STRM) messages may be going to incorrect signaling points, resulting in dropped or misrouted calls.
Corrective Action Taken:	The audit sets CCapci2dpci[index] equal to CCsmdpcls_ht[CCapci2dpci[index].dpci].apcindx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCdpclsidx.
<i>Bad Data =</i>	CCapci2dpci[index].
<i>Good Data =</i>	CCsmdpcls_ht[CCapci2dpci[index]].apcindx.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

Table 4 Error Code: BAD_APCI

Error Code:	BAD_APCI
Error Description:	The error is due to one of the following: 1 = rIDPC_STAT.apcindx does not equal rIDPC_INFO.apcindx for the DPCI. 2 = CCsmdpcls_ht[rIDPC_STAT.dpci].apcindx does not equal rIDPC_STAT.apcindx for the DPCI. 3 = CCapci2dpci[CCsmdpcls_ht[rIDPC_STAT.dpci]].apcindx does not equal rIDPC_STAT.apcindx.
Possible Error Effect:	STRM messages may be going to incorrect signaling points, resulting in dropped or misrouted calls in both cases.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets rIDPC_STAT.apcindx equal to rIDPC_INFO.apcindx.

	2 = The audit sets CCsmdpcls_ht[rIDPC_STAT.dpci].apcindx equal to rIDPC_STAT.apcindx for the DPCI. 3 = The audit sets CCapci2dpci[CCsmdpcls_ht[rIDPC_STAT.dpci]].apcindx equal to rIDPC_STAT.apcindx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	(1), (2) or (3) indicates Error Description, Corrective Action and Dump Description.
<i>Bad Data</i> =	Corresponds to the Error Description and Corrective Action, as follows: 1 = rIDPC_STAT.apcindx.
<i>Good Data</i> =	Corresponds to the Error Description and Corrective Action, as follows: 1 = rIDPC_INFO.apcindx. 2 = rIDPC_STAT.apcindx. 3 = rIDPC_STAT.dpci.
<i>Logical Index</i> =	Corresponds to the Error Description and Corrective Action, as follows: 1 = rIDPC_INFO.dpci. 2 = rIDPC_STAT.dpci. 3 = rIDPC_STAT.apcindx.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 5 Error Code: BAD_BIT_SET

Error Code:	BAD_BIT_SET
Error Description:	A DPC is an APC. Its remote route status bit in rIDPC_STAT.rrs is not set.
Possible Error Effect:	Route set testing is done unnecessarily on the direct link to the APC.
Corrective Action Taken:	The audit sets the remote route status bit for the DPC in rIDPC_STAT.rrs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_INFO.apcindx.
<i>Bad Data</i> =	rIDPC_STAT.rrs.
<i>Good Data</i> =	rIDPC_STAT.dpc.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 6 Error Code: BAD_CONGEST

Error Code:	BAD_CONGEST
Error Description:	A DPC is inaccessible. Its remote route congestion (rIDPC_STAT.rmtcg) should be equal to CCNOCONG, but is not.
Possible Error Effect:	STRM tries unsuccessfully to reduce the remote congestion level of an inaccessible DPC.
Corrective Action Taken:	The audit sets rIDPC_STAT.rmtcg equal to CCNOCONG for the DPC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_STAT.dpc.
<i>Bad Data</i> =	rIDPC_STAT.rmtcg.
<i>Good Data</i> =	CCNOCONG.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 7 Error Code: BAD_DPC

Error Code:	BAD_DPC
Error Description:	The audit checks the ordering of the DPCs in CCdpc2dpci[] against the ordering in RLDPC_STAT. The array is missing a DPC.
Possible Error Effect:	Common channel signaling 7 (CCS7) cannot signal/route to the missing DPC.
Corrective Action Taken:	The audit sets CCdpc2dpci[].dpc to rIDPC_STAT.dpc, and CCdpc2dpci[].dpci to rIDPC_STAT.dpci, and triggers table hashsums to be recalculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc of previous rIDPC_STAT tuple.
<i>Bad Data =</i>	CCdpc2dpci[index].dpc which is out of order.
<i>Good Data =</i>	rIDPC_STAT.dpc from the current tuple.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

Table 8 Error Code: BAD_LSOWNER

Error Code:	BAD_LSOWNER
Error Description:	The error is due to one of the following: 1 = A DPC is owned by a link set (rILS_STAT.lsnun), but the link set is neither the normal link set nor one of the alternate link sets used to reach the DPC. 2 = A DPC is owned by a combined link set (rCLS_STAT.lsnun + CC7MXLS), but the combined link set is neither the normal combined link set nor one of the alternate combined link sets used to reach the DPC.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC in both cases.
Corrective Action Taken:	The audit links the rIDPC_STAT tuple to the rILS_STAT idle list and sends a message to do route correction to the STRM system process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 1 = rILS_STAT.lsnun. 2 = rCLS_STAT.lsnun + CC7MXLS.
<i>Good Data =</i>	(1) or (2) indicates Error and Dump Descriptions.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple..

Table 9 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The error is due to one of the following: 1 = rIDPC_STAT.cls_own is set to DBNO. The rIDPC_STAT.dpc_ls.st_ownk should contain the physical key of a tuple in relation RLLS_STAT but it does not. 2 = rIDPC_STAT.cls_own is set to DBYES. The rIDPC_STAT.dpc_ls.st_ownk should contain the physical key of a tuple in relation RLCLS_STAT but it does not.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	In either case, the audit links the rIDPC_STAT tuple to the rILS_STAT idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.

<i>Bad Data</i> =	rIDPC_STAT.dpc.
<i>Good Data</i> =	(1) or (2) indicates Error Description.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 10 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The remote route status bit in rIDPC_STAT.rss for a DPCI is set. The routing data block (RDB) for the DPCI of the DPC is on route set test (RST) timing but it should not be on RST timing.
Possible Error Effect:	Unnecessary RST timing messages are generated for the DPC.
Corrective Action Taken:	The audit removes the DPCI from route set test timing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDPC_STAT tuple.
<i>Bad Data</i> =	rIRSTMR.rnum.
<i>Good Data</i> =	rIDPC_STAT.dpci.
<i>Logical Index</i> =	rIS7RDB.rdb_id.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 11 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The DPC is inaccessible. The RDB for the DPCI of the DPC should be on RST timing but it is not.
Possible Error Effect:	STRM tries unsuccessfully to reduce the remote congestion level of an inaccessible DPC.
Corrective Action Taken:	The RDB is placed on RST timing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDPC_STAT tuple.
<i>Bad Data</i> =	rIRSTMR.rnum invalid owner.
<i>Good Data</i> =	rIDPC_STAT.dpci.
<i>Logical Index</i> =	rIS7RDB.rdb_id.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 12 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	An rIS7RDB tuple should exist for every offset of a DPCI. The audit determines that a tuple is missing and uses a database (DB) primitive to insert the tuple. The primitive fails to insert the tuple.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit tries the insertion during its next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	rIDPC_STAT.dpci.
<i>Good Data</i> =	Offset of the routing data block tuple.
<i>Logical Index</i> =	rIS7RDB.rdb_id.
<i>Dump</i> =	NA.

Table 13 Error Code: BDPCR

Error Code:	BDPCR
Error Description:	CCsmdpcis_ht[rIDPC_STAT.dpci].norm_rte does not equal rIDPC_STAT.rt_prll for a DPCI.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit sets CCsmdpcis_ht[rIDPC_STAT.dpci].norm_rte equal to rIDPC_STAT.rt_prll.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of copy of CCsmdpcls_ht[] entry.
<i>Bad Data</i> =	CCsmdpcls_ht[].norm_rte.
<i>Good Data</i> =	rIDPC_STAT.rt_prll.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 14 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	A DPC is marked accessible (rIDPC_STAT.acc equals DBYES), but the bit for its DPCI in the bitmap in the adjacent point code index array (CCdpc_status[]) is not set to 1.
Possible Error Effect:	CCS7 resources may be wasted trying to make an accessible DPC inaccessible.
Corrective Action Taken:	The audit sets the bit in the bitmap in CCdpc_status[] to 1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_STAT.acc.
<i>Bad Data</i> =	Incorrect bit value.
<i>Good Data</i> =	rIDPC_STAT.dpc.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 15 Error Code: BIT2MAP

Error Code:	BIT2MAP
Error Description:	A DPC is marked inaccessible (rIDPC_STAT.acc equals DBNO) but the bit for its DPCI in the bitmap in adjacent point code index array (CCdpc_status[]) is not set to 0.
Possible Error Effect:	CCS7 resources may be wasted trying to make an inaccessible DPC accessible.
Corrective Action Taken:	The audit sets the bit in the bitmap in CCdpc_status[] to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_STAT.acc.
<i>Bad Data</i> =	Incorrect bit value.
<i>Good Data</i> =	rIDPC_STAT.dpc.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 16 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The DPC is accessible. Its remote congestion is set to not congested. The RDB for the DPCI of the DPC is on RST timing but it should not be on RST timing.
Possible Error Effect:	STRM tries unsuccessfully to reduce the remote congestion level of an inaccessible DPC.
Corrective Action Taken:	The DPC is unlinked from the rIRCGTMR busy list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDPC_STAT tuple.
<i>Bad Data</i> =	Logical key of rIRCGTMR owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 17 Error Code: CCINDEX

Error Code:	CCINDEX
Error Description:	The error is due to one of the following: 1 = The CCS parameter CCapcindx does not equal the number of adjacent point code indexes (APCIs) in the office (rIINDEX.max_idx) where rIINDEX.idx_name = CCAPCINDEX. 2 = The CCS parameter CCdpcsidx does not equal the highest DPCI in the

	office (rIINDEX.max_idx), where rIINDEX.idx_name = CCDPCIMAX.
Possible Error Effect:	In both cases, calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets CCapcindx equal to rIINDEX.max_idx. 2 = The audit sets CCdpclsidx equal to rIINDEX.max_idx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1) or (2) indicates Error and Dump Descriptions.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 1 = CCapcindx. 2 = CCdpclsidx.
<i>Good Data =</i>	rIINDEX.max_idx.
<i>Logical Index =</i>	Corresponds to the Error Description as follows: 1 = CCAPCINDEX. 2 = CCDPCIMAX.
<i>Dump =</i>	NA.

Table 18 Error Code: CLS1_LSNUM_MISS

Error Code:	CLS1_LSNUM_MISS
Error Description:	The owner of a DPC is a combined link set. Link set one (rICLS_STAT.lsn1) of the combined link set (rICLS_STAT.clsnum) is not found in RLLS_STAT.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit inserts the missing link set in RLLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpci.
<i>Bad Data =</i>	rICLS_STAT.lsn1.
<i>Good Data =</i>	rIDPC_STAT.dpc.
<i>Logical Index =</i>	rICLS_STAT.clsnum.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 19 Error Code: CLS2_LSNUM_MISS

Error Code:	CLS2_LSNUM_MISS
Error Description:	The owner of a DPC is a combined link set. Link set two (rICLS_STAT.lsn2) of the combined link set (rICLS_STAT.clsnum) is not found in RLLS_STAT.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit inserts the missing link set in RLLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpci.
<i>Bad Data =</i>	rICLS_STAT.lsn2.
<i>Good Data =</i>	rIDPC_STAT.dpc.
<i>Logical Index =</i>	rICLS_STAT.clsnum.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 20 Error Code: CLSNUM_BAD

Error Code:	CLSNUM_BAD
Error Description:	The error is due to one of the following:

	1 = The link set (rIDPC_DPCI.lnk_set) which owns a DPC is a combined link set. A tuple whose key (rCLS_STAT.clsnum) equals rIDPC_DPCI.lnk_set - CC7MXLS does not exist in RLCLS_STAT. 2 = An alternate link set (rIDPC_DPCI.altls[0]) for a DPC is a combined link set. A tuple whose key (rCLS_STAT.clsnum) equals rIDPC_INFO.altls[] - CC7MXLS does not exist in RLCLS_STAT
Possible Error Effect:	In either case, traffic diversion is not done correctly.
Corrective Action Taken:	Another audit inserts the missing tuple into relation RLCLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into rIDPC_INFO.altls[].
<i>Bad Data =</i>	Corresponds to the Error Description as follows:
	1 = rIDPC_DPCI.lnk_set. 2 = rIDPC_INFO.altls[index].
<i>Good Data =</i>	(1) or (2) indicates Error and Dump Descriptions.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 21 Error Code: CLS_BAD

Error Code:	CLS_BAD
Error Description:	The error is due to one of the following: 1 = The normal route for a DPC is a combined link set (rIDPC_DPCI.lnk_set) but rIDPC_STAT.cls[0] does not equal rIDPC_DPCI.lnk_set. 2 = The normal route for a DPC is a combined link set (rIDPC_DPCI.lnk_set), but rIDPC_STAT.rte[0] does not equal the offset of the routing data block for the normal route used to reach the DPC. 3 = The first alternate route for a DPC is a combined link set (rIDPC_INFO.altls[0] > CC7MXLS), but rIDPC_STAT.cls[] does not equal rIDPC_INFO.altls[] for the DPC. 4 = An alternate route for a DPC is a combined link set (rIDPC_INFO.altls[] > CC7MXLS), but rIDPC_STAT.rte[0] does not equal the offset of the routing data block for the alternate route used to reach the DPC.
Possible Error Effect:	Traffic diversion is not done correctly in all cases.
Corrective Action Taken:	Corresponds to the Error Description as follows:
	1 = The audit sets rIDPC_STAT.cls[0] equal to rIDPC_DPCI.lnk_set for the DPC. 2 = The audit sets rIDPC_STAT.rte[0] equal to 0 for the DPC. 3 = The audit sets rIDPC_STAT.cls[] equal to rIDPC_INFO.altls[]. 4 = The audit set rIDPC_STAT.rte[] equal to the offset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1), (2) or (3) indicates Error Description, Corrective Action and Dump Description.
<i>Bad Data =</i>	Corresponds to the Error Description and Corrective Action, as follows:

	1 = rIDPC_STAT.cls[0]. 2 = rIDPC_STAT.rte[0].
<i>Good Data</i> =	Corresponds to the Error Description and Corrective Action, as follows: 1 = rIDPC_DPCI.lnk_set. 2 = 0 (offset of first routing data block). 4 = Offset of the routing data block for alternate route.
<i>Logical Index</i> =	Corresponds to the Error Description and Corrective Action, as follows: 1 = rIDPC_STAT.dpci. 2 = rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 22 Error Code: CLS_LSNUM_BAD

Error Code:	CLS_LSNUM_BAD
Error Description:	A DPC is owned by a combined link set (rCLS_STAT.clsnum). CCsmdpcls_ht[rIDPC_STAT.dpci].ls does not equal rCLS_STAT.clsnum + CC7MXLS for the DPC.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit sets CCsmdpcls_ht[rIDPC_STAT.dpci].ls equal to rCLS_STAT.clsnum + CC7MXLS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_STAT.dpc.
<i>Bad Data</i> =	CCsmdpcls_ht[rIDPC_STAT.dpci].ls.
<i>Good Data</i> =	rCLS_STAT.clsnum + CC7MXLS.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 23 Error Code: CON_ERR_APC_DPC

Error Code:	CON_ERR_APC_DPC
Error Description:	The audit tries to get the adjacent DPCI for a DPC which is an adjacent DPC for an APCI. It uses rAPCI_APC.apc to find a match in RLDPC_DPCI for rAPCI_APC.apc but cannot find a match.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	None.
Manual Action Required:	Insert an RLDPC_DPCI tuple with the key (rIDPC_DPCI.dpc) equal to rAPCI_APC.apc. Refer to population rules for RLDPC_DPCI.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDPC_DPCI.dpc.
<i>Dump</i> =	NA.

Table 24 Error Code: CONFLICT

Error Code:	CONFLICT
Error Description:	The error is one of the following: 1 = The current DPC (rIDPC_STAT.dpc) is an SCCP DPC because there is a tuple in relation relation RLDPC_SSN whose rIDPC_SSN.dpc equals

	rIDPC_STAT.dpc and whose rIDPC_SSN.net_num equals the current global switching module (GSM) number. The rIDPC_STAT.sccp_con for this DPC should be set to DBYES but is not. 2 = The current DPC (rIDPC_STAT.dpc) is not an SCCP DPC because there is no tuple in relation RLDPC_SSN whose rIDPC_SSN.dpc equals rIDPC_STAT.dpc and whose rIDPC_SSN.net_num equals the current GSM number. The rIDPC_STAT.sccp_con for this DPC should be set to DBNO but is not.
Possible Error Effect:	Corresponds to the Error Description as follows: 1 = The message transfer part (MTP) on this GSM thought that the DPC is not an SCCP DPC. MTP has not been reporting status changes of the DPC to SCCP. SCCP may be routing its traffic over an incorrect route or may be blocking traffic. 2 = The MTP on this GSM thought that the DPC is an SCCP DPC. Some real-time is spent reporting status changes of the DPC to SCCP which will ignore the reports.
Corrective Action Taken:	The audit set rIDPC_STAT.sccp_con to DBYES in the first case and to DBNO in the second case.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	AUnet_num (SCCP's number for the current GSM).
<i>Bad Data =</i>	rIDPC_STAT.sccp_con.
<i>Good Data =</i>	Correct value for rIDPC_STAT.sccp_con.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of RLDPC_STAT tuple.

Table 25 Error Code: COUNT_BAD

Error Code:	COUNT_BAD
Error Description:	While stepping through the list of DPCs in the office (using rIDPC_STAT.nextdpc), the audit has found either: 1 = Too many DPCs (more than rIINDX.max_idx) or 2 = Too few DPCs (less than rIINDX.max_idx).
Possible Error Effect:	CCS7 cannot build the routing table, and the audit cannot verify the routing table because dynamic relation rIDPC_STAT is corrupt. CCS7 cannot signal/route to any DPC.
Corrective Action Taken:	The audit triggers table hashsums to be recalculated and sends a message to STRM to regenerate the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc.
<i>Bad Data =</i>	rIDPC_STAT.nextdpc.
<i>Good Data =</i>	rIINDX.max_idx.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 26 Error Code: DPCZERO_ERROR

Error Code:	DPCZERO_ERROR
Error Description:	The rIDPC_STAT tuple with the key (rIDPC_STAT.dpci) of zero contains the DPCI (rIDPC_STAT.nextdpc) of the first DPC in the office. This tuple with the key of zero must exist, but does not.
Possible Error Effect:	CCS7 cannot signal/route to any DPC.

Corrective Action Taken:	The audit inserts the rIDPC_STAT tuple with the key of zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 27 Error Code: DPCI_CON_ERR

Error Code:	DPCI_CON_ERR
Error Description:	CCdpc2dpci[] is the DPC to DPCI table built from rIDPC_STAT. CCdpc2dpci[].dpci does not equal rIDPC_STAT.dpci for a DPC.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit sets CCdpc2dpci[].dpci equal to rIDPC_STAT.dpci.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpci.
<i>Bad Data =</i>	CCdpc2dpci[index].dpci.
<i>Good Data =</i>	rIDPC_STAT.dpci.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

Table 28 Error Code: DPCPRE_SUC

Error Code:	DPCPRE_SUC
Error Description:	The audit is unable to read the rIDPC_STAT tuple whose key is stored in the current rIDPC_STAT tuple's nextdpc field. rIDPC_STAT.nextdpc from the current rIDPC_STAT tuple is used to read another rIDPC_STAT tuple, but the read is not successful.
Possible Error Effect:	The audit cannot check that the routing table is correctly populated due to incorrect database linkage of the DPCs. CCS7 cannot signal/route to the DPC whose DPCI is stored in rIDPC_STAT.nextdpc of the current rIDPC_STAT tuple.
Corrective Action Taken:	The audit triggers table hashsums to be recalculated and sends a message to STRM to regenerate the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.nextdpc of current tuple.
<i>Good Data =</i>	rIDPC_STAT.dpc of current tuple.
<i>Logical Index =</i>	rIDPC_STAT.dpci of current tuple.
<i>Dump =</i>	NA.

Table 29 Error Code: DPC_ACC

Error Code:	DPC_ACC
Error Description:	The error is due to one of the following: 1 = The DPC is owned by the rILS_STAT idle list because rIDPC_STAT.dpc_ls.st_ownd equals DBDPC_STATIDLE. The rIDPC_STAT.acc in the tuple equals DBYES, but should equal DBNO. 2 = The DPC is owned by the rILS_STAT busy list because rIDPC_STAT.dpc_ls.st_ownd is greater than DBDPC_STATIDLE. The rIDPC_STAT.acc in the tuple equals DBNO, but should equal DBYES. 3 = The DPC is owned by the rICLS_STAT busy list because rIDPC_STAT.dpc_ls.st_ownd has the physical key of a tuple in relation RLCLS_STAT. The rIDPC_STAT.acc equals DBNO, but should equal DBYES.

Possible Error Effect:	Corresponds to the Error Description as follows: 1 = Invalid accessibility status could be sent to the signaling message handler (SMH). The SMH in turn may try to send a message to link set 0. 2 = Incorrect data may be sent to the SMH. If the SMH's tables are built from the corrupted data, the SMH may send message to the DPC since it will assume that DPC is accessible even though it is not. 3 = Incorrect data may be sent to the SMH. If the SMH's tables are built from the corrupted data, the SMH may send message to the DPC since it will assume that DPC is accessible even though it is not.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets rIDPC_STAT.acc equal to DBNO and sets rIDPC_STAT.rmtcg equal to CCNOCONG. 2 = The audit links the rIDPC_STAT tuple to the rILS_STAT idle list and sends a message to the STRM to do route correction. 3 = The audit links the rIDPC_STAT tuple to the rILS_STAT idle list and sends a message to the STRM to do route correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc.
<i>Bad Data =</i>	Corresponds to the Error Description, Error Effect and Corrective Action as follows: 1 = rILS_STAT.lsnnum. 2 = rILS_STAT.lsnnum. 3 = rCLS_STAT.clsnum.
<i>Good Data =</i>	(1), (2), or (3) indicates Error Description, Corrective Action and Dump Description.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 30 Error Code: DPC_BAD

Error Code:	DPC_BAD
Error Description:	The error is due to one of the following: 1 = The rIDPC_STAT.dpc does not equal rIDPC_DPCI.dpc for the DPCI of a DPC. 2 = CCsmdpcls_ht[rIDPC_STAT.dpci].dpc does not equal rIDPC_STAT.dpc for the DPCI of a DPC.
Possible Error Effect:	Operational code accesses incorrect data if it tries to use rIDPC_STAT.dpc as the key to read rIDPC_DPCI.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets rIDPC_STAT.dpc equal to rIDPC_DPCI.dpc. 2 = The audit has CCsmdpcls_ht[rIDPC_STAT.dpci].dpc set equal to rIDPC_STAT.dpc.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1) or (2) indicates Error Description, Corrective Action and Dump Description.
<i>Bad Data =</i>	Corresponds to the Error Description and Corrective Action as follows: 1 = rIDPC_STAT.dpc. 2 = CCsmdpcis_ht[rIDPC_STAT.dpci].dpc.
<i>Good Data =</i>	Corresponds to the Error Description and Corrective Action as follows: 1 = rIDPC_DPCI.dpc. 2 = rIDPC_STAT.dpc.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 31 Error Code: DPC_CON_ERR

Error Code:	DPC_CON_ERR
Error Description:	The number of DPCs in the office (CCno_dpc_sm) does not equal the number the audit reads from rIINDEX.max_idx.
Possible Error Effect:	Any DPC inserted/deleted using recent change (RC) either fails or invalidates the ordering of the DPCs if the insert/delete succeeds. CCS7) cannot signal/route to the DPC which was inserted, or CCS7 may try to signal/route to the DPC which was deleted.
Corrective Action Taken:	The audit sets CCno_dpc_sm equal to rIINDEX.max_idx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCno_dpc_sm.
<i>Bad Data =</i>	Value of CCno_dpc_sm.
<i>Good Data =</i>	rIINDEX.max_idx.
<i>Logical Index =</i>	rIINDEX.idx_name.
<i>Dump =</i>	NA.

Table 32 Error Code: EXTRA_DPC

Error Code:	EXTRA_DPC
Error Description:	RLDPC_STAT indicates that the office has DPCs (rIDPC_STAT.nextdpc != 0 for the tuple with the key of zero), but there are no DPCs in the office (rIINDEX.max_idx for CCS7 == zero).
Possible Error Effect:	There is an invalid DPC in the routing table. CCS7 never accesses it, but the routing table is built incorrectly. This may trigger asserts during SM/PH initializations.
Corrective Action Taken:	The audit sets rIDPC_STAT.nextdpc in the tuple with the key of zero to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.nextdpc
<i>Good Data =</i>	rIINDEX.max_idx.
<i>Logical Index =</i>	CCDPCINDEX.
<i>Dump =</i>	NA.

Table 33 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The APCI (rILS_STAT.apci) does not equal rILS_GRP.apci for the same link set which is a primary/alternate route for a DPC. The link set is the key to both relations (rILS_STAT.lsnnum and rILS_GRP.lsnnum).
Possible Error Effect:	If operational code tries to translate the APCI to an APC, it will get the wrong APC or no APC at all.

Corrective Action Taken:	The audit sets rLS_STAT.apci equal to rLS_GRP.apci for the link set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT.
<i>Bad Data =</i>	rLS_STAT.apci.
<i>Good Data =</i>	rLS_GRP.apci.
<i>Logical Index =</i>	Link set number.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 34 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	A destination point code (DPC) is owned by the rLS_STAT idle list. It is also owned by the rRCGTMR busy list, but should not be on remote congestion timing because it is on the idle list of relation RLLS_STAT.
Possible Error Effect:	STRM tries unsuccessfully to reduce the remote congestion level of an inaccessible DPC.
Corrective Action Taken:	The audit unlinks the DPC's rIDPC_STAT tuple from the rRCGTMR busy list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	Key of rRCGTMR owner tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 35 Error Code: INDEX

Error Code:	INDEX
Error Description:	CCsmdpcls_ht[rIDPC_STAT.dpci].mtxindx does not equal rIDPC_STAT.mtxindx for a DPC.
Possible Error Effect:	Operational code will access incorrect data if it uses CCsmdpcls_ht[rIDPC_STAT.dpci].mtxindx to index into the policing matrix table.
Corrective Action Taken:	The audit sets CCsmdpcls_ht[rIDPC_STAT.dpci].mtxindx equal to rIDPC_STAT.mtxindx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc.
<i>Bad Data =</i>	CCsmdpcls_ht[rIDPC_STAT.dpci].mtxindx.
<i>Good Data =</i>	rIDPC_STAT.mtxindx.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	NA.

Table 36 Error Code: INFO_ERR

Error Code:	INFO_ERR
Error Description:	The accessibility of the DPC in the NM array (NMpcc7buff[rIDPC_STAT.dpci].cur_dpc_inac) does not equal rIDPC_STAT.acc.
Possible Error Effect:	Incorrect point code status could be reported to the remote network management center.
Corrective Action Taken:	The audit sets NMpcc7buff[rIDPC_STAT.dpci].cur_dpc_inac to DBNO if rIDPC_STAT.acc is DBYES and to DBYES if rIDPC_STAT.acc is DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc.
<i>Bad Data =</i>	NMpcc7buff[rIDPC_STAT.dpci].cur_dpc_inac.
<i>Good Data =</i>	rIDPC_STAT.acc.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	NA.

Table 37 Error Code: INV

Error Code:	INV
Error Description:	The remote congestion value (rIDPC_STAT.rmtcg) for a DPC is set to an invalid value.

Possible Error Effect:	The remote congestion of the DPC on the SM might not agree with remote congestion status of the DPC on the responsible PH.
Corrective Action Taken:	The audits corrects the value of the remote route congestion during its next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.rmtcg.
<i>Good Data =</i>	rIDPC_STAT.dpc.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	NA.

Table 38 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	There is an extra DPC in CCdpc2dpci[] that is not linked in rIDPC_STAT.
Possible Error Effect:	The rIDPC_STAT.nextdpc linkage in rIDPC_STAT is corrupt. If the routing table is rebuilt from this incorrect rIDPC_STAT, routing to the DPC is lost (see good data), and CCS7 may do invalid routing diversion. This triggers asserts during fault recovery from signaling link failures or signaling network reconfiguration.
Corrective Action Taken:	The audit triggers table hashsums to be recalculated and sends a message to STRM to regenerate the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc of previous rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.dpc of current rIDPC_STAT tuple.
<i>Good Data =</i>	DPC not linked in rIDPC_STAT.
<i>Logical Index =</i>	rIDPC_STAT.dpci of previous rIDPC_STAT tuple.
<i>Dump =</i>	NA.

Table 39 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The rIDPC_STAT.acc equals DBYES for a DPC. Its rIDPC_STAT.rmtcg equals a valid congestion value, but the rIDPC_STAT tuple is linked to a busy rIRCGTMR tuple whose logical key is greater than the maximum (DBRCGTMX + DBSFRCGTMR) allowed.
Possible Error Effect:	The DPC is continuously congested and certain level 4 MSUs might not be generated.
Corrective Action Taken:	The audit sets rIDPC_STAT.rmtcg equal to CCNOCONG and unlinks the rIDPC_STAT tuple from the rIRCGTMR tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of GIcongupd
<i>Bad Data =</i>	Key of rIRSTMR owner (rIDPC_STAT.dpc ls.st_ownk).
<i>Good Data =</i>	rIDPC_STAT.rmtcg.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 40 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The rIDPC_STAT.acc equals DBYES for a DPC. Its rIDPC_STAT.rmtcg equals a valid congestion value, but the rIDPC_STAT tuple is not linked to the rIRCGTMR busy list. The rIDPC_STAT.dpc ls.st_ownk equals DBNULL.
Possible Error Effect:	The DPC is continuously congested and certain level 4 MSUs might not be generated.
Corrective Action Taken:	The audit sets rIDPC_STAT.rmtcg to CCNOCONG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc.
<i>Bad Data =</i>	Logical key of rIRSTMR owner tuple (rIDPC_STAT.dpc ls.st_ownk).
<i>Good Data =</i>	rIDPC_STAT.rmtcg.
<i>Logical Index =</i>	rIDPC_STAT.dpci.

<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 41 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	The ordering of the DPCs in CCdpc2dpci[] has been verified. The CCdpc2dpci[].dpc of the next entry in the table should be zero to mark the end of the list, but it is not zero.
Possible Error Effect:	CCS7 does invalid routing diversion. This triggers asserts during fault recovery from signaling link failures or signaling network reconfiguration.
Corrective Action Taken:	The audit sets CCdpc2dpci[].dpc in the end of the list entry to zero and triggers table hashsums to be recalculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCdpc2dpci[indx].dpci.
<i>Bad Data</i> =	CCdpc2dpci[index].dpc.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index.
<i>Dump</i> =	NA.

Table 42 Error Code: LSNUM_BAD

Error Code:	LSNUM_BAD
Error Description:	A DPC is owned by a link set because rIDPC_STAT.clsown is set to DBNO but CCsmdpcls_ht[rIDPC_STAT.dpci].ls does not equal rLS_STAT.lsnun of the owning link set.
Possible Error Effect:	Traffic diversion is not done correctly.
Corrective Action Taken:	The audit sets CCsmdpcls_ht[rIDPC_STAT.dpci].ls equal to rLS_STAT.lsnun of the owning link set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_STAT.dpc.
<i>Bad Data</i> =	CCsmdpcls_ht[rIDPC_STAT.dpci].ls.
<i>Good Data</i> =	rLS_STAT.lsnun.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 43 Error Code: LS_CON_ERR

Error Code:	LS_CON_ERR
Error Description:	The error is due to one of the following: 1 = The primary link set for a DPC is a combined link set (rIDPC_DPCI.lnk_set), but there is no tuple in rCLS_LS with a key (rCOS_STAT.clsnum) equal to rIDPC_DPCI.lnk_set. 2 = An alternate link set for a DPC is a combined link set (rIDPC_INFO.altls[]), but there is no tuple in rCLS_LS with a key (rCLS_STAT.clsnum) equal to rIDPC_INFO.altls[]. 3 = A tuple in rLS_GRP with the key (rLS_GRP.lsnun) equal to rIDPC_DPCI.lnk_set or rIDPC_INFO.altls[], if a regular link set, or a component of rIDPC_DPCI.lnk_set or rIDPC_INFO.altls[], if a combined link set, does not exist.
Possible Error Effect:	In all cases, calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	None.
Manual Action Required:	Corresponds to the Error Description as follows: 1 = Insert a tuple into RLCLS_LS with rCLS_LS.cls equal to rIDPC_DPCI.lnk_set. Consult population rules for RLCLS_LS.

	2 = Insert a tuple into RLCLS_LS with rCLS_LS.cls equal to rIDPC_INFO.altls[]. Consult population rules for RLCLS_LS. 3 = Insert a tuple into RLLS_GRP with rLS_GRP.lsnnum equal to the link set number listed in bad data.
Dump Description:	
<i>Error Address =</i>	(1), (2) or (3) indicates Error Description, Manual Action, and Dump Description.
<i>Bad Data =</i>	Corresponds to the Error Description and manual action as follows: 1 = rIDPC_DPCI.lnk_set. 3 = Link set number.
<i>Good Data =</i>	Corresponds to the Error Description and manual action as follows: 1 = rIDPC_DPCI.dpcindx. 2 = rIDPC_STAT.dpc. 3 = NA.
<i>Logical Index =</i>	Corresponds to the Error Description and manual action as follows: 1 = rIDPC_DPCI.dpc. 2 = rIDPC_INFO.dpcindx. 3 = Link set number.
<i>Dump =</i>	NA.

Table 44 Error Code: LS_STAT_NO

Error Code:	LS_STAT_NO
Error Description:	One of the link sets used for routing to a DPC does not exist in rLS_STAT. The link set is either a primary route or an alternate route. The link set may be in relation RLDPC_DPCI (rIDPC_DPCI.lnk_set), relation RLDPC_INFO (rIDPC_INFO.altls[] or relation RLCLS_LS (rCLS_LS.ls1 or rCLS_LS.ls2).
Possible Error Effect:	Calls cannot be routed to the DPC.
Corrective Action Taken:	Another audit inserts the missing link set in relation RLLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Link set number.
<i>Dump =</i>	NA.

Table 45 Error Code: MINDX

Error Code:	MINDX
Error Description:	The policing matrix index (rIDPC_STAT.mtxindx) for a DPC does not equal rIDPC_DPCI.mtxindx where the key of relation RLDPC_DPCI is rIDPC_DPCI.dpc.
Possible Error Effect:	Operational code accesses incorrect data if it uses rIDPC_STAT.mtxindx to index into the policing matrix table.
Corrective Action Taken:	The audit sets rIDPC_STAT.mtxindx equal to rIDPC_DPCI.mtxindx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc.

<i>Bad Data</i> =	rIDPC_STAT.mtxindx.
<i>Good Data</i> =	rIDPC_DPCI.mtxindx.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 46 Error Code: MISSING

Error Code:	MISSING
Error Description:	RLDPC_STAT indicates that the office has no DPCs (rIDPC_STAT.nextdpc == 0 for the tuple with the key of zero), but there are DPC in the office (rINDX.max_idx for CCS7 != zero).
Possible Error Effect:	CCS7 cannot build the routing table, and the audit cannot verify the routing table because dynamic relation rIDPC_STAT is corrupt. CCS7 cannot signal/route to any DPC.
Corrective Action Taken:	The audit sends STRM a message to regenerate the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_STAT.dpci.
<i>Bad Data</i> =	rIDPC_STAT.nextdpc.
<i>Good Data</i> =	rINDX.max_idx.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 47 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Measurements in RLSMDPCCNT are used for adjacent signaling point (ASP) and non-ASP reports. A tuple with the key of rIDPC_STAT.dpci + CCITT7NDX for a DPCI does not exist.
Possible Error Effect:	A DPC exists without any measurements data. Measurements data is lost.
Corrective Action Taken:	The audit inserts the rSMDPCCNT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDPC_STAT tuple.
<i>Bad Data</i> =	rIDPC_STAT.dpci + CCITT7NDX.
<i>Good Data</i> =	rIDPC_STAT.dpci.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 48 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The APCI of a link set for a DPC does not exist in rIAPCI_APC. The key of relation RLAPCI_APC is rIAPCI_APC.apci and it is the APCI of one of the alternate link sets whose remote route status (RRS) bit in rIDPC_STAT.rrs does not match the RRS bit in rIDPC_STAT.rrs for the normal link set. The audit derives the APCIs from relation RLLS_STAT for each link set.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	Another audit corrects the APCIs in relation RLLS_STAT for the link sets of the normal and alternate routes of the DPC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDPC_STAT tuple.
<i>Bad Data</i> =	APCI used to read rIAPCI_APC.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 49 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	CCsmdpcls_htf[] is the DPC to link set routing table. An entry in CCsmdpcls_htf[] should be 0, but is not because the corresponding bit in AUccg_bits[] used to

	indicate that the entry was processed is not set.
Possible Error Effect:	Cross-checks done by recent change and verify (RC/V) may cause data validation errors when the office routing data are changed.
Corrective Action Taken:	The audit sets the CCsmdpcls_ht[] entry to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCsmdpcls_ht[index] entry.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	Contents of non-null CCsmdpcls_ht[index] entry.

Table 50 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The CCdpc2dpci[].dpc of the last entry in the table must be zero to mark the end of the table, but it is not zero.
Possible Error Effect:	Indexes into the routing table may be incorrect. CCS7 code may try to route to an incorrect DPC.
Corrective Action Taken:	The audit sets CCdpc2dpci[\$nbsp;].dpc in the last entry in the table to zero and triggers table hashsums to be recalculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of last CCdpc2dpci[index] entry.
<i>Bad Data =</i>	CCdpc2dpci[index].dpc.
<i>Good Data =</i>	0 (value of what last CCdpc2dpci[].dpc entry should be).
<i>Logical Index =</i>	Index of last CCdpc2dpci[] entry.
<i>Dump =</i>	NA.

Table 51 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIDPC_STAT.acc for a DPC is set to DBNO. The DPC should not be linked to the rRCGTMR busy list, but it is. (The rIDPC_STAT.tcrd.st_ownk should equal DBNULL, but it does not.)
Possible Error Effect:	STRM tries unsuccessfully to reduce remote congestion level of an inaccessible DPC.
Corrective Action Taken:	The audit unlinks the DPC from the rRCGTMR busy list. The DPC is not linked to an idle list since it is assumed that if the DPC is not on the busy list, it is idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	Key of rRCGTMR owner tuple (rIDPC_STAT.tcrd.st_ownk).
<i>Good Data =</i>	DBRCIDLE (value of what rIDPC_STAT.tcrd.st_ownk should be).
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 52 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	There are no DPCs in the office (rLINDX.max_idx for CCS7 is zero). CCdpc2dpci[0].dpc, CCdpc2dpci[0].dpci, and CCdpc2dpci[1].dpc should all be zero, but one or more is not zero.
Possible Error Effect:	CCS7 does invalid routing diversion. This triggers asserts during fault recovery from signaling link failures or signaling network reconfiguration.
Corrective Action Taken:	The audit sets CCdpc2dpci[0].dpc, CCdpc2dpci[0].dpci and CCdpc2dpci[1].dpc to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCdpc2dpci[0] entry.
<i>Bad Data =</i>	CCdpc2dpci[0].dpc.
<i>Good Data =</i>	CCdpc2dpci[0].dpci.
<i>Logical Index =</i>	CCdpc2dpci[1].dpc.
<i>Dump =</i>	NA.

Table 53 Error Code: NOT_SORTED

Error Code:	NOT_SORTED
Error Description:	The previous rIDPC_STAT.dpc is greater than the current rIDPC_STAT.dpc. This indicates that the ordering of the DPCs is incorrect. The DPCs should be in ascending order.
Possible Error Effect:	The audit cannot check that the routing table is correctly populated due to incorrect database linkage of the DPCs. CCS7 may fail to find a particular DPC when searching CCdpc2dpci[], and so would not be able to signal/route to that DPC.
Corrective Action Taken:	The audit triggers table hashsums to be recalculated and sends a message to STRM to regenerate the DPCs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc of current rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.dpci of current rIDPC_STAT tuple.
<i>Good Data =</i>	rIDPC_STAT.dpc of previous rIDPC_STAT tuple.
<i>Logical Index =</i>	rIDPC_STAT.dpci of previous rIDPC_STAT tuple.
<i>Dump =</i>	NA.

Table 54 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	The error is due to one of the following: 1 = An unused element in the combined link set array (rIDPC_STAT.cls[]) or the routing data block array (rIDPC_STAT.rte[]) for a DPC should be 0 or 0xff, respectively, but is not. An unused element in the combined link set array means that there is no combined link set used to reach the DPC. An unused element in the routing data block array corresponds to the unused combined link set array element. 2 = The RRS bits in rIDPC_STAT.rrs for the unused routing data blocks should not be set, but one or more are set.
Possible Error Effect:	Incorrect data may be used for routing.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets the unused rIDPC_STAT.cls[] entry to 0 and sets the unused rIDPC_STAT.rte[] entry to 0xff. 2 = The audit clears the RRS bit for the unused routing data block.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	Corresponds to the Error Description and Corrective Action as follows:
	2 = Offset of the routing data block.
<i>Good Data =</i>	(1) or (2) indicates Error Description, Corrective Action and Dump Description.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 55 Error Code: NO_1RDB_DPCI

Error Code:	NO_1RDB_DPCI
Error Description:	The RRS bits (rIDPC_STAT.rrs) for parallel link sets for a DPC are not equal but there is no tuple in rIS7RDB for the DPC whose APCI for one of the parallel link sets matches the DPCI for the DPC. The key for an rIS7RDB tuple is rIS7RDB.rdb_id and is equal to (rIDPC_STAT.dpci * CCRDB_DPC) + the RDB

	offset).
Possible Error Effect:	Operational code may not use an accessible route if RRS was marked inaccessible, or if the RRS is marked accessible but it really is not, the remote exchange will return transfer prohibited.
Corrective Action Taken:	The audit insets the missing rIS7RDB tuple during the audit's next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_DPCI.dpc.
<i>Bad Data =</i>	rIAPCI_APC.apci_apc.
<i>Good Data =</i>	Offset of routing data block.
<i>Logical Index =</i>	rIDPC_DPCI.dpcindx.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 56 Error Code: NO_MATCH_LS_STAT

Error Code:	NO_MATCH_LS_STAT
Error Description:	Either the primary or one of the alternate link sets for a DPC that is derived from rIDPC_DPCI.lnk_set or rIDPC_INFO.altls[], respectively, does not have a tuple in relation RLLS_STAT whose key (rILS_STAT.lsnun) matches the link set.
Possible Error Effect:	Operational code does not find dynamic information about the link set or any of its links. Change over or change back cannot be performed on any link in the link set.
Corrective Action Taken:	The missing tuple whose key (rILS_STAT.lsnun) is equal to the primary or alternate link set is inserted into relation RLLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_STAT.dpc
<i>Bad Data =</i>	Index.
<i>Good Data =</i>	rIDPC_DPCI.lnk_set.
<i>Logical Index =</i>	rIDPC_INFO.altls[index].
<i>Dump =</i>	NA.

Table 57 Error Code: NO_RST_DPCI

Error Code:	NO_RST_DPCI
Error Description:	The remote route status bit in rIDPC_STAT.rrs for the DPCI of an APCI is clear. The DPC for the DPCI should be on route set test timing, but is not. That is, rIS7RDB.rsrd.st_ownk for the DPCI should have the physical key of a valid rIRSTMR tuple but it does not.
Possible Error Effect:	Calls being routed to the DPC may be blocked when they should no longer be blocked.
Corrective Action Taken:	The audit sends STRM a message to put the DPC on route set test timing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	APCI.
<i>Bad Data =</i>	Offset of the routing data block.
<i>Good Data =</i>	rIDPC_STAT.rrs.
<i>Logical Index =</i>	rIDPC_STAT.dpci for the APCI.
<i>Dump =</i>	NA.

Table 58 Error Code: PCI_BAD

Error Code:	PCI_BAD
Error Description:	The error is one of the following: 1 = The rIDPC_STAT.pci (point code index) does not equal rIDPC_DPCI.pci for the DPCI (destination point code index). 2 = CCsmdpcls_ht[rIDPC_STAT.dpci].pci does not equal rIDPC_STAT.pci for the DPCI (rIDPC_STAT.dpci).
Possible Error Effect:	Corresponds to the Error Description as follows:

	1 = Operational code accesses incorrect data if it tries to use rIDPC_STAT.dpc as the key to read rIDPC_DPCI, and routing will not be done to the correct point code. 2 = SS7 calls do not complete.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets rIDPC_STAT.pci equal to rIDPC_DPCI.pci. 2 = The audit has CCsmdpcls_ht[rIDPC_STAT.dpci].pci set equal to rIDPC_STAT.pci.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	(1) or (2) indicates Error Description, Error Effect, Corrective Action, and Dump Description.
<i>Bad Data</i> =	Corresponds to the Error Description, Error Effect, Corrective Action, and Dump Description as follows: 1 = rIDPC_STAT.pci. 2 = CCsmdpcls_ht[rIDPC_STAT.dpci].pci.
<i>Good Data</i> =	Corresponds to the Error Description, Error Effect, Corrective Action, and Dump Description as follows: 1 = rIDPC_DPCI.pci. 2 = rIDPC_STAT.pci.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Corresponds to the Error Description as follows: 1 = NA. 2 = Contents of rIDPC_STAT tuple.

Table 59 Error Code: PTCODE_BAD

Error Code:	PTCODE_BAD
Error Description:	NMpcc7buf[rIDPC_STAT.dpci].ptcode does not equal rIDPC_STAT.dpc.
Possible Error Effect:	NM reports status of an incorrect DPC for a DPCI. Routing to the DPC may fail.
Corrective Action Taken:	The audit sets NMpcc7buf[rIDPC_STAT.dpci].ptcode equal to rIDPC_STAT.dpc.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of NMpcc7buf[rIDPC_STAT.dpci] entry.
<i>Bad Data</i> =	NMpcc7buf[rIDPC_STAT.dpci].ptcode.
<i>Good Data</i> =	rIDPC_STAT.dpc.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	NA.

Table 60 Error Code: RDB_BAD

Error Code:	RDB_BAD
Error Description:	The error is one of the following: 1 The audit calls a CCS7 function to retrieve the representative RDB for the associated cluster-route one of the APCIs. The audit then tries to find a tuple in relation RLS7RDB whose key (rIS7RDB.rdb_id) equals the retrieved RDB.

	2 = The audit calls a CCS 7 function to retrieve the representative RDB for the associated cluster-route one of the APCIs serving a DPC. The audit then tries to find a tuple in relation RLS7RDB whose key (rlS7RDB.rdb_id) equals the retrieved RDB.
Possible Error Effect:	In both cases, a DPC may have an incorrect accessibility status and signaling traffic may be routed to this DPC over incorrect link sets.
Corrective Action Taken:	In both cases, another audit verifies data in CCCrdata[] and CCcr_rdb[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1) or (2) indicates Error Description.
<i>Bad Data =</i>	APCI for the cluster-route.
<i>Good Data =</i>	RDB for the cluster-route.
<i>Logical Index =</i>	rlDPC_STAT.dpci of current tuple.
<i>Dump =</i>	Contents of RLDPC_STAT tuple.

Table 61 Error Code: RDB_MISSING

Error Code:	RDB_MISSING
Error Description:	An rlS7RDB tuple for one of the offsets for an rlDPC_STAT.dpci does not exist. The offset is the number of the routing data block for a DPCI. It has a value of 0 through CCRDB_DPC.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC for the DPCI.
Corrective Action Taken:	The missing rlS7RDB tuple is inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rlDPC_STAT.dpci.
<i>Good Data =</i>	Offset of the routing data block.
<i>Logical Index =</i>	rlS7RDB.rdb_id.
<i>Dump =</i>	NA.

Table 62 Error Code: READ

Error Code:	READ
Error Description:	The tuple with the key of 0 (rlDPC_STAT.dpci = 0) does not exist in RLDPC_STAT.
Possible Error Effect:	CCS7 cannot signal/route to any DPC.
Corrective Action Taken:	The audit inserts the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 63 Error Code: RID

Error Code:	RID
Error Description:	The error is one of the following: 1 = The RRS bits in rlDPC_STAT.rrs for APCIs are not equal. The audit calls a CCS7 function to retrieve the representative RDB for the associated cluster-route one of the APCIs, but the function fails to retrieve the RDB. 2 = The audit calls a CCS 7 function to retrieve the representative RDB for the associated cluster-route one of the APCIs serving a DPC, but the function fails to retrieve the RDB.
Possible Error Effect:	In both cases, a DPC may have an incorrect accessibility status and signaling traffic may be routed to this DPC over incorrect link sets.

Corrective Action Taken:	In both cases, another audit verifies data in CCCrdata[] and CCcr_rdb[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	(1) or (2) indicates Error Description.
<i>Bad Data</i> =	APCI for the cluster-route.
<i>Good Data</i> =	Cluster.
<i>Logical Index</i> =	rIDPC_STAT.dpci of current tuple.
<i>Dump</i> =	Contents of RLDPC_STAT tuple.

Table 64 Error Code: ROUTE_STATUS

Error Code:	ROUTE STATUS
Error Description:	The audit has detected inconsistent parallel route status. 1 = The owner of a DPC is a regular link set. The normal route for the DPC is regular link set. The APCI of the owning link set equals the APCI of the normal route. This indicates that the routes are parallel, but rIDPC_STAT.rt_prll for the DPC is set to DBNO instead of DBYES. 2 = The owner of a DPC is a regular link set. The normal route for the DPC is a combined link set. The APCI of the owning link set equals the APCI of the normal or the first alternate route. This indicates that the routes are parallel, but rIDPC_STAT.rt_prll for the DPC is set to DBNO instead of DBYES. 3 = The owner of a DPC is a combined link set. The normal route for the DPC is a regular link set. The APCIs of the component link sets equal the APCI of the normal route. This indicates that the routes are parallel, but rIDPC_STAT.rt_prll for the DPC is set to DBNO instead of DBYES. 4 = The normal route for DPC is a combined link set. The APCI of the link set which owns the DPC equals the APCI of the normal or first alternate route and the APCI of the other link set equals the APCI of the normal or first alternate route. This indicates that the routes are parallel, but rIDPC_STAT.rt_prll for the DPC is set to DBNO instead of DBYES.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC in all cases.
Corrective Action Taken:	In all cases, the audit sets rIDPC_STAT.rt_prll to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDPC_DPCI.lnk_set (owning link set).
<i>Bad Data</i> =	rIDPC_STAT.rt_prll.
<i>Good Data</i> =	(1), (2), (3), or (4) indicates Dump Description.
<i>Logical Index</i> =	rIDPC_STAT.dpci.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 65 Error Code: RRSTATUS

Error Code:	RRSTATUS
Error Description:	The normal and alternate route for a DPC are parallel because their APCIs (rIDPC_STAT.apci) are equal. The respective RRS bits in rIDPC_STAT.rrs of these parallel routes must also be equal but are not.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an inaccessible DPC.
Corrective Action Taken:	The audit marks the RRS for both the normal and alternate route inaccessible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	APCI of the normal route.
<i>Bad Data</i> =	APCI of an alternate route.
<i>Good Data</i> =	The RRS bit in rIDPC_STAT.rrs for the normal route.
<i>Logical Index</i> =	The RRS bit in rIDPC_STAT.rrs for the alternate route.
<i>Dump</i> =	Contents of rIDPC_STAT tuple.

Table 66 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The audit reads RLINDEX with the appropriate key (rlINDEX.idx_name) to find: 1 = The number of APCs in the office (rlINDEX.max_idx). 2 = The highest DPCI in the office (rlINDEX.max_idx). 3 = The number of DPCs in the office (rlINDEX.max_idx). The rlINDEX tuple does not exist. The tuple should exist because the SM is a global SM (GI_pcird equals Glglobalsm) and the CCS7 package is present for this office. Which case applies can be determined by the value of bad data.
Possible Error Effect:	No CCS7 signaling/routing is done for the office in all cases.
Corrective Action Taken:	None.
Manual Action Required:	Corresponds to the Error Description as follows: 1 = Insert the missing rlINDEX tuple with rlINDEX.idx_name set to CCAPCINDEX and rlINDEX.max_idx set to the number of APCs in the office. 2 = Insert the missing rlINDEX tuple with rlINDEX.idx_name set to CCDPCIMAX and rlINDEX.max_idx set to the highest DPCI in the office. 3 = Insert the missing rlINDEX tuple with rlINDEX.idx_name set to CCDPCINDEX and rlINDEX.max_idx set to the number of DPCs in the office. Consult the population rules for relation RLINDEX.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	(1), (2), or (3) indicates Error Description, Manual Action and Dump Description.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Appropriate key corresponds to the Error Description and manual action as follows: 1 = CAPCINDEX. 2 = CDPCIMAX. 3 = CDPCINDEX.
<i>Dump =</i>	NA.

Table 67 Error Code: SMLOSS_INC

Error Code:	SMLOSS_INC
Error Description:	NM's count (NMsmc7loss) of the inaccessible DPCs does not equal the count taken by the audit. The audit counts a DPC as inaccessible if its rIDPC_STAT.acc is set to DBNO.
Possible Error Effect:	Incorrect status regarding the accessibility of a DPC may be sent to the remote network centers and calls may not be routed to the DPC.
Corrective Action Taken:	The audit sets NMsmc7loss equal to the count as determined by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NMsmc7loss.
<i>Good Data =</i>	Count as determined by the audit.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 68 Error Code: SMRCG_INC

Error Code:	SMRCG_INC
Error Description:	NM's count (NMsmc7rcg) of the remotely congested DPCs does not equal the count taken by the audit. The audit counts a DPC as remotely congested if its rIDPC_STAT.rmtcg is not set to CCNOCONG.
Possible Error Effect:	Incorrect status regarding the congestion of a DPC may be sent to the remote network centers and calls may not be routed to the DPC.
Corrective Action Taken:	The audit sets NMsmc7rcg equal to the count as determined by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NMsmc7rcg.
<i>Good Data =</i>	Count as determined by the audit.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 69 Error Code: STPRTE_BAD

Error Code:	STPRTE_BAD
Error Description:	The error is due to one of the following: 1 = The signaling transfer point (rIDPC_STAT.stprte) does not equal rIDPC_INFO.stprte for a DPC. 2 = The signaling transfer point in the routing table (CCsmdpcls_ht[rIDPC_STAT.dpci].stprte) does not equal rIDPC_STAT.stprte for a DPC.
Possible Error Effect:	Signaling traffic may be routed or not routed according to the value of the signaling transfer point.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit sets rIDPC_STAT.stprte equal to rIDPC_INFO.stprte. 2 = The audit sets CCsmdpcls_ht[rIDPC_STAT.dpci].stprte equal to rIDPC_STAT.stprte.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	(1) or (2) indicates Error Description, Corrective Action and Dump Description.
<i>Bad Data =</i>	Corresponds to the Error Description and Corrective Action as follows: 1 = rIDPC_STAT.stprte. 2 = CCsmdpcls_ht[rIDPC_STAT.dpci].stprte.
<i>Good Data =</i>	Corresponds to the Error Description and Corrective Action as follows: 1 = rIDPC_INFO.stprte. 2 = rIDPC_STAT.stprte.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of rIDPC_STAT tuple.

Table 70 Error Code: UPDATE_FAILED

Error Code:	UPDATE_FAILED
Error Description:	CCsmdpcls_ht[] is the DPC to link set routing table. An attempt to correct an CCsmdpcls_ht[rIDPC_STAT.dpci] entry failed.
Possible Error Effect:	Calls may be dropped and/or unnecessary network traffic may be routed to an

	inaccessible DPC.
Corrective Action Taken:	The audit makes another update attempt during its next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCsmdpcis ht[rIDPC_STAT.dpci] entry in error.
<i>Bad Data =</i>	Return value from attempt.
<i>Good Data =</i>	rIDPC_STAT.dpc.
<i>Logical Index =</i>	rIDPC_STAT.dpci.
<i>Dump =</i>	Contents of what array entry should be.

Table 71 Error Code: XTRA_DPC

Error Code:	XTRA_DPC
Error Description:	NMpcc7buf[] has an entry for each valid DPC. An entry has its visit and new fields set to DBNO and its pcode field not set to 0. This indicates that there is an extra DPC entry in NMpcc7buf[].
Possible Error Effect:	A nonexistent DPC could be reported to the remote NM center.
Corrective Action Taken:	The audit deletes the extra DPC in NMpcc7buf[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of NMpcc7buf[index] entry.
<i>Bad Data =</i>	NMpcc7buf[index].ptcode.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

Table 72 Error Code: ZERO_TUP_BAD

Error Code:	ZERO_TUP_BAD
Error Description:	The rIDPC_STAT.dpc in the rIDPC_STAT tuple with the key of 0 (rIDPC_STAT.dpci = 0) does not equal 0. This DPC must be the lowest DPC because it is used to check the ordering of the DPCs.
Possible Error Effect:	The ordering of the DPCs may be incorrect. If the ordering is incorrect, signaling/routing to a particular DPC cannot be done.
Corrective Action Taken:	The audit sets rIDPC_STAT.dpc equal to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDPC_STAT tuple.
<i>Bad Data =</i>	rIDPC_STAT.dpc.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

DS0B**Software Release:** 5E10 and later**1. DIGITAL SERVICE 0 BLOCK (DS0B) AUDIT**

The DS0B audit performs the dynamic to static and semantic checks of the integrated digital service unit (IDCU) digital service (DS1) facility time slots. These times slots are also known as DS0Bs (there are 24 DS0Bs per DS1). It uses the indexed access relation driver DRINDEX to loop through the RLDS0B relation.

2. DATA STRUCTURES

The data structure being audited by the DS0B audit is a relational structure known as RLDS0B. One rIDS0B tuple is allocated for each DS0B time slot on an IDCU DS1 facility. The rIDS0B tuple is comprised of many fields, the ones that are audited are:

day	= The day this DS0 was put into 7 second timing.
ds_ch	= Shared one-to-one linkage to the rICHDB or the rIPTSB.
ds0b (KEY)	= The DS1 and DS0 number.
dscm	= Set-member linkage to the rCCBCOM tuple.
dscqo	= Set-owner linkage to the rCQB tuple.
portname	= The port name of the port using this DS0.
quarstat	= The quarantine state.
state	= The state of this DS0.
tod	= Time of day this DS0 was put into 7 second timing.
use	= Indicates how the rIDS0B is being used (such as, PPD, PPB, ODB, nail-up, hairpin, circuit switched analog, or circuit switched digital port).

3. AUDIT CHECKS

The audit checks done are as follows:

ASSOC_PORT	= The rIDS0B tuple is being used for hairpin connection but the rIDS0B.portname field is not a valid port.
BADUSE	= The path information in rNAILUP are not correct for a given hairpin connection. tuples are different.
CCBCOM_LINK	= The rIDS0B tuple linkage to rCCBCOM tuple is broken.
CHDB_LINK	= The rIDS0B tuple linkage to rICHDB tuple is broken.
CKTDATA	= The rNAILUP information associated with the rCKTDATA tuple is not correct.
CQB_LINK	= The rIDS0B tuple linkage to rCQB tuple is broken.
EXTRATUP	= The rCKTDATA tuple for the DS1 facility is missing.

GPORT	= The rIDS0B and rCCBCOM tuples port information do not match.
LOST_CKTDATA	= The rCKTDATA tuple associated with the rNAILUP tuple is missing.
LOST_PORTLA	= The rPORTLA tuple is missing for a given port.
LOST_SMEST	= The rSMEST tuple for the DS1 facility is missing.
LPORT	= The time slot information in an rIDS0B and rPORTLA
NON_NULL	= The rIDS0B tuple linkage to rCCBCOM tuple was broken.
NOT7SEC	= The rIDS0B tuple has been on DS0B7SEC list for more than 100 seconds.
ODDLOSTTUP	= The rNAILUP tuple is missing for a given hairpin connection.
OUTOFRANGE	= The rIDS0B.use field was invalid.
PORTNAME	= The port information in the rIDS0B and rPTSB tuples are inconsistent.
PTSB_LINK	= The rIDS0B tuple linkage to rPTSB tuple is broken.
STATE	= The rIDS0B.state field was not correct.
STUCK	= The rIDS0B tuple is stuck in the set-up state for 2 audit cycles.
WRONGLIST	= The rIDS0B tuple is on the wrong list.
ZERO_LINK	= An idle rIDS0B tuple has a non-null linkage.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	The rIDS0B tuple is being used for hairpin connection but the rIDS0B.portname field is not a valid port. This field should contain the name of the port that the rIDS0B tuple is hairpinned to.
Possible Error Effect:	The rIDS0B tuple is not hairpinned to the correct port and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list will be determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The rIDS0B tuple is used for hairpin connection but the path information in the

	static rINAILUP tuple does not match the information in the rIDS0B tuple.
Possible Error Effect:	The wrong rIDS0B tuple is being used for hairpin connection. The rIDS0B tuple will be recovered to the correct list. The list is determined by IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDS0B.ds0b key field.
<i>Dump 1</i> =	The rIDS0B tuple in error.
<i>Dump 2</i> =	The rINAILUP tuple.

Table 3 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	The rIDS0B tuple was not linked to a valid rCCBCOM tuple.
Possible Error Effect:	The rIDS0B resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDS0B.ds0b field key.
<i>Dump</i> =	The rIDS0B tuple in error.

Table 4 Error Code: CHDB_LINK

Error Code:	CHDB_LINK
Error Description:	The rIDS0B tuple was not linked to a valid rCHDB tuple. to be.
Possible Error Effect:	The rIDS0B tuple resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDS0B.ds0b key field.
<i>Dump</i> =	The rIDS0B tuple in error.

Table 5 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	The rIDS0B tuple is a time slot on a DS1 facility. The audit tried to read the rCKTDATA tuple for the DS1 facility but it was missing from the database.
Possible Error Effect:	The rINAILUP tuple does not contain a valid circuit name and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDS0B.ds0b key field.
<i>Dump</i> =	The rIDS0B tuple in error.

Table 6 Error Code: CQB_LINK

Error Code:	CQB_LINK
Error Description:	The rIDS0B tuple is not validly linked to an rCQB tuple.

Possible Error Effect:	The rIDS0B resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 7 Error Code: GPORT

Error Code:	GPORT
Error Description:	The rIDS0B.portname is not matching the port information in the rCCBCOM.port field
Possible Error Effect:	The rIDS0B tuple is not linked to the correct port and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 8 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The rIDS0B tuple is a time slot on a DS1 facility. The audit tried to read the rCKTADTA tuple for the DS1 facility but it was missing from the database.
Possible Error Effect:	The rNAILUP tuple does not contain a valid circuit name and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 9 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	The rPORTLA tuple whose key is the rIDS0B.portname field in the or the rNAILUP.port3 member is lost.
Possible Error Effect:	The rIDS0B tuple is not hairpinned to the correct port and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	The list that the rIDS0B tuple is linked to.
<i>Good Data =</i>	The list that the rIDS0B tuple should be linked to.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump 1 =</i>	The rIDS0B tuple in error.
<i>Dump 2 =</i>	The dump of the rNAILUP tuple.

Table 10 Error Code: LOST_SMEST

Error Code:	LOST_SMEST

Error Description:	The rIDS0B tuple is a time slot on a DS1 facility. The audit tried to read the rSMEST tuple for the DS1 facility but it was missing from the database.
Possible Error Effect:	Call processing will try to use an unusable rIDS0B tuple.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the out-of-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 11 Error Code: LPORT

Error Code:	LPORT
Error Description:	The rIDS0B.portname field is used as the key to read the rIORTLA tuple. The rIORTLA.ds0b was found not to match rIDS0B.ds0b.
Possible Error Effect:	The rIDS0B tuple is not hairpinned to the correct port and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	The list that the rIDS0B tuple is linked to.
<i>Good Data =</i>	The list that the rIDS0B tuple should be linked to.
<i>Logical Index =</i>	The key of the rIDS0B tuple.
<i>Dump 1 =</i>	The rIDS0B tuple in error.
<i>Dump 2 =</i>	The rINAILUP tuple.

Table 12 Error Code: NON-NULL

Error Code:	NON-NULL
Error Description:	The rIDS0B tuple was not suppose to be linked to an rCCBCOM tuple but was found to be linked to one.
Possible Error Effect:	The rIDS0B tuple is not used correctly and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 13 Error Code: NOT7SEC

Error Code:	NOT7SEC
Error Description:	The rIDS0B tuple is being used for 7 second timing but has been there for more than 100 seconds. The tuple is stuck in that state and is not usable.
Possible Error Effect:	The rIDS0B tuple is not usable and the resource is lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	The list that the rIDS0B tuple is linked to.
<i>Good Data =</i>	The list that the rIDS0B tuple should be linked to.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump 1 =</i>	The rIDS0B tuple in error.
<i>Dump 2 =</i>	The local structure that contains the current time and date.

Table 14 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The rIDS0B tuple is being used for hairpin connection but there is no static rINAILUP tuple in the database with the rIDS0B.portname field used as the key.
Possible Error Effect:	The rIDS0B tuple is not hairpinned to the correct port and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IN/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.portname field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 15 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The rIDS0B.use field was set to an out of range value.
Possible Error Effect:	The rIDS0B tuple is not used correctly and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 16 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	The rIDS0B.use is set for PCDPPD (provisioned packet D-channel) but the rIPTSB.pfdb.ftspstb.portname field does not match the rIDS0B.portname field information.
Possible Error Effect:	The rIDS0B tuple is not used correctly and the resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 17 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The rIDS0B tuple is not linked to a valid rIPTSB tuple.
Possible Error Effect:	The DS0B resource will be lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDS0B tuple in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rIDS0B.ds0b key field.
<i>Dump =</i>	The rIDS0B tuple in error.

Table 18 Error Code: STATE

Error Code:	STATE

Error Description:	The rIDS0B.state field was not set to match the use of the tuple (such as, the rIDS0B tuple may be used in a stable call, but the rIDS0B.state field is set to PCIDLE, or the rIDS0B tuple may be idle but the rIDS0B.state field is set to PCDSTABLE).
Possible Error Effect:	The rIDS0B tuple is not usable.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list that is recovered to is dependent on the IS/OOS state of the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	The value of the rIDS0B.state field.
<i>Good Data</i> =	The value of what the rIDS0B.state field should be.
<i>Logical Index</i> =	The key of the rIDS0B tuple (rIDS0B.ds0b).
<i>Dump</i> =	The rIDS0B tuple in error.

Table 19 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rIDS0B.state field was found to be set to call setup state (PCDSETUP) or call tear down (PCDTEARDOWN) value for 2 audit cycles. It implies that it is stuck in this state since both setup up and tear down take less than 1 audit cycle to be accomplished.
Possible Error Effect:	The rIDS0B tuple is not usable.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list that is recovered to is dependent on the IS/OOS state of the DS1 facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	The value of the rIDS0B.quarstat field
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIDS0B tuple (rIDS0B.ds0b).
<i>Dump</i> =	The rIDS0B tuple in error.

Table 20 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rIDS0B tuple is linked to the wrong list.
Possible Error Effect:	The rIDS0B tuple is not usable.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1's rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	The list that the rIDS0B tuple is currently linked to.
<i>Good Data</i> =	The list the rIDS0B tuple should be linked to.
<i>Logical Index</i> =	The rIDS0B.ds0b key field.
<i>Dump</i> =	The rIDS0B tuple in error.

Table 21 Error Code: ZEROLINK

Error Code:	ZEROLINK
Error Description:	The rIDS0B tuple is idle but one of the linkages has a none null value.
Possible Error Effect:	The rIDS0B tuple is not usable and the resource is lost.
Corrective Action Taken:	The rIDS0B tuple will be recovered to the correct list. The list is determined by the IS/OOS state of the DS1 facility's rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDS0B tuple in error.
<i>Bad Data</i> =	The value of the non zero link found.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIDS0B.ds0b key field.
<i>Dump</i> =	The rIDS0B tuple in error.

DSCC

Software Release: 5E10 and later

1. DIGITAL SERVICE CIRCUIT COMMON BOARDS (DSCC) AUDIT

The global digital service unit (GDSU) provides low usage digital service functions such as conferencing and transmission test facilities. The two service groups in the GDSU contain information about GDSU common boards, such as which peripheral interface data bus (PIDB) time slots are used by the GDSU subtending circuits. Each tuple in the digital service circuit relation (RLDSCC) has a one-to-one correspondence with a GDSU common board.

The digital service circuit common boards audit (DSCC) verifies that every rLDSCC tuple has a corresponding tuple in the static circuit data relation (RLCKTDATA) and in the dynamic circuit data block common area relation (RLCDBCOM). This audit also verifies that the GDSU PIDB time slot channel in rLDSCC is consistent with the channel in the associated rLCDBCOM tuple. If extra tuples collect in RLDSCC, hardware growth is inhibited; therefore, the DSCC audit is needed to identify inconsistencies, delete extra tuples, and correct bad keys.

This audit is part of the peripheral control audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The DSCC audit uses the data in the static circuit data relation (RLCKTDATA) to verify the data in the dynamic digital service circuit common relation (RLDSCC). The RLDSCC relation contains dynamic information such as the circuit name and the TSI channel array, which specifies whether channels are busy. Figure 1 shows the fields in rLDSCC checked by the audit. The value in the TSI channel array is verified against the associated circuit data block common area relation (RLCDBCOM) link to its associated channel data block relation (RLCHDB).

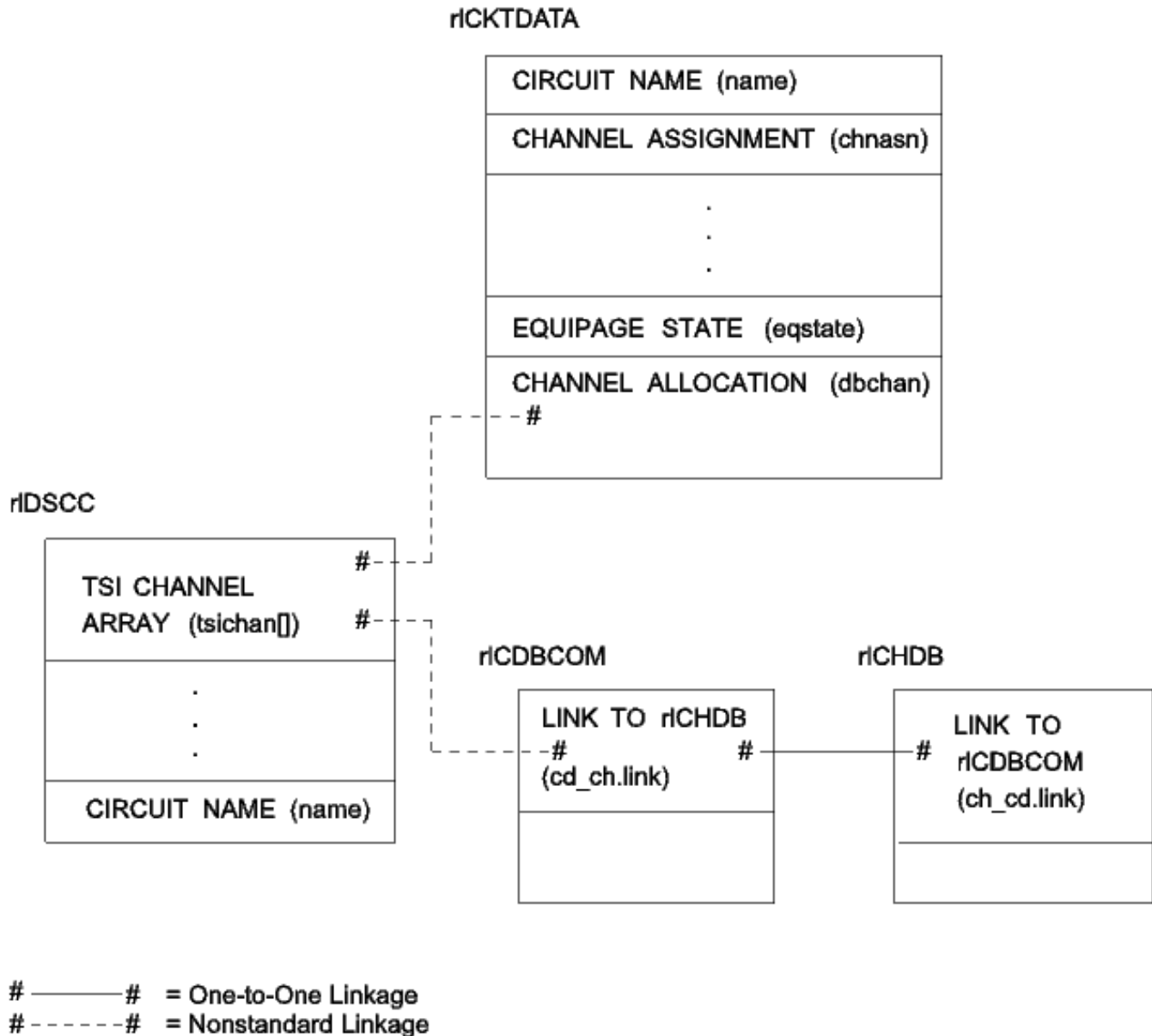


Figure 1 (DSCC-1) Digital Service Circuit Common Relation

3. AUDIT CHECKS

The digital service circuit common audit uses the static driver (DRSTATIC) to step through the RLDSCC relation.

- (a) Initially, the audit calls a dynamic to static existence check function to perform the following checks:
- (1) For each tuple in RLDSCC, the existence check function verifies that a corresponding tuple exists in the static RLCKTDATA relation.
 - (2) It checks for any extra rDSCC tuples.
 - (3) It checks the equipage state of rCKTDATA. If the state is unequipped (UNEQ), no rDSCC tuple should exist for that corresponding circuit.

(b) Next, the audit calls a semantic check function to perform the following checks:

- (1) If rICKTDATA is operational or in a growth state, the semantic check function verifies that an associated rCDBCOM tuple exists.
- (2) The audit calculates all possible GDSU subtending circuit names in the GDSU service group, and for each name, it accesses the associated rCDBCOM tuple. If no rCDBCOM tuple is found, the circuit is unequipped and the audit tries the next circuit.
- (3) The audit accesses the associated rICKTDATA tuple. If no tuple is found, an error is reported. Otherwise, the audit uses the channel allocation field (dbchan) in rICKTDATA as the array entry for obtaining the corresponding PIDB time slot channel number.
- (4) Depending on the channel assignment (the chnasn field in rICKTDATA), the audit compares the value in the rIDSCC tuple array against the rCDBCOM link to rICHDB.
 - If chnasn equals PCNOCHAN (no channel), no checking is done.
 - If chnasn equals PCINITCHAN (dedicated at initialization) or PCALLOCHAN (allocated when used), the audit checks the array value. If an inconsistency exists between the array value and the rCDBCOM link to rICHDB, an error is reported.
- (5) The audit steps through the array in the rIDSCC tuple and accesses an rICHDB tuple for each assigned (non -1) entry. The rICHDB link to rCDBCOM is used to access the associated rICKTDATA tuple. If no rICKTDATA tuple is found, an error is reported. If a tuple is found and the chnasn field does not equal PCNOCHAN, the array entry (not the contents) is compared with the dbchan field in rICKTDATA. If the two fields are not equal, an error is reported. If the chnasn field does equal PCNOCHAN, no checking is done.

4. ERROR HANDLING

Any tuple in RLDSCC that does not correspond to a tuple in RLCKTDATA, or has a corresponding rICKTDATA tuple in the unequipped state (UNEQ), is deleted and an error is reported.

If the rIDSCC tuple has valid rICKTDATA information but no associated rCDBCOM tuple, an error is reported.

If the rIDSCC TSI channel array entry is inconsistent with the rCDBCOM channel (link to rICHDB), the rIDSCC TSI channel array entry is corrected and an error is reported.

The CDBCOM audit is scheduled as a related audit to complete the recovery procedure.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: DBCHAN

Error Code:	DBCHAN
Error Description:	While stepping through the tsichan array in the rIDSCC tuple, an array entry was found that does not match the dbchan value in rICKTDATA.
Possible Error Effect:	The erroneously marked tsichan array entry in rIDSCC may cause serious problems if accessed by other operation or maintenance software (such as, IIA).
Corrective Action Taken:	Correct the tsichan array entries in rIDSCC. Set the erroneous entry to -1 and

	set the entry corresponding to dbchan to the channel number from rCHDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDSCC tuple.
<i>Bad Data</i> =	Value of tsichan array entry.
<i>Good Data</i> =	dbchan value from rCKTDATA.
<i>Logical Index</i> =	Circuit name of the rIDSCC tuple.
<i>Dump</i> =	NA.

Table 2 Error Code: DSCC_CDB

Error Code:	DSCC_CDB
Error Description:	The rIDSCC tsichan array entry does not match the link to rCHDB in rCDBCOM. This error also occurs if the link to rCHDB is 0 and the tsichan array entry is not -1.
Possible Error Effect:	The erroneously marked tsichan array entry in rIDSCC may cause serious problems if accessed by other operation or maintenance software (such as, IIA).
Corrective Action Taken:	Correct the tsichan array entry in rIDSCC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDSCC tuple.
<i>Bad Data</i> =	Value of tsichan array entry.
<i>Good Data</i> =	The link to rCHDB in rCDBCOM.
<i>Logical Index</i> =	Circuit name of the rIDSCC tuple.
<i>Dump 1</i> =	rIDSCC tuple.
<i>Dump 2</i> =	rCDBCOM tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The rIDSCC tuple corresponds to an rCKTDATA tuple which is unequipped or not of the type "PCGDSUCOM".
Possible Error Effect:	Loss of resources and possible problems if operational or maintenance software accesses this data (such as, IIA).
Corrective Action Taken:	Delete the rIDSCC tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDSCC tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name of the rIDSCC tuple.
<i>Dump</i> =	rIDSCC tuple.

Table 4 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	The rCCDBCOM tuple for this valid rIDSCC tuple is vacant.
Possible Error Effect:	The GDSU common board circuit data is missing which may cause problems for the interrupt inhibit administrator (IIA).
Corrective Action Taken:	None. The CDBCOM audit is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name of the rIDSCC tuple.
<i>Dump</i> =	rIDSCC tuple.

Table 5 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The channel assignment for the GDSU subtending circuit in question is assigned at initialization time. However, the tsichan array entry in rIDSCC does not match the link to rCHDB in rCDBCOM.
Possible Error Effect:	The erroneously marked tsichan array entry in rIDSCC may cause serious

	problems if accessed by other operation or maintenance software (such as, IIA).
Corrective Action Taken:	Correct the tsichan array entry in rIDSCC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDSCC tuple.
<i>Bad Data =</i>	Value of tsichan array entry.
<i>Good Data =</i>	The link to rICHDB in rCDBCOM.
<i>Logical Index =</i>	Circuit name of the rIDSCC tuple.
<i>Dump 1 =</i>	rIDSCC tuple.
<i>Dump 2 =</i>	rCDBCOM tuple.

Table 6 Error Code: TSMAP2

Error Code:	TSMAP2
Error Description:	There is garbage in one of the tsichan array entries in rIDSCC.
Possible Error Effect:	This garbage could cause other operational and maintenance software to cause problems (such as, IIA).
Corrective Action Taken:	The array entry is recovered to -1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIDSCC tuple.
<i>Bad Data =</i>	Value in the tsichan array entry.
<i>Good Data =</i>	-1.
<i>Logical Index =</i>	Circuit name of the rIDSCC tuple.
<i>Dump 1 =</i>	rIDSCC tuple.
<i>Dump 2 =</i>	rCDBCOM tuple.

Table 7 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	The rCDBCOM tuple for this GDSU subtending circuit appears to be an extra tuple. The read of rICKTDATA with this circuit name resulted in a no-match.
Possible Error Effect:	The extra tuple ties up resources.
Corrective Action Taken:	None. The CDBCOM audit is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCDBCOM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name of the rIDSCC tuple.
<i>Dump 1 =</i>	rIDSCC tuple.
<i>Dump 2 =</i>	rCDBCOM tuple.

DSLEQUIP

Software Release: 5E10 and later

1. DIGITAL SUBSCRIBER LINE EQUIPMENT STATUS (DSLEQUIP) AUDIT

The DSLEQUIP audit performs a static-to-dynamic existence check against the dynamic terminal equipment data block (TEDB) relation for a subscriber of default service. The DLSEQUIP audit also performs a static-to-dynamic existence check against the dynamic integrated service digital network (ISDN) maintenance data block (ISMDB) relation. This audit is part of ISDNBASE package and resides on the operating system for distributed switching-switching module (OSDS-SM). The DSLEQUIP audit is running at the segmented mode as well as the initialization mode during selective initialization.

2. DATA STRUCTURES

The RLDSLEQUIP is a static relation with the port as the key. If the user subscribes to default service in this digital subscriber line (DSL), the dflt_grp field in DSLEQUIP indicates the default group number. (Refer to Figure 1 .)

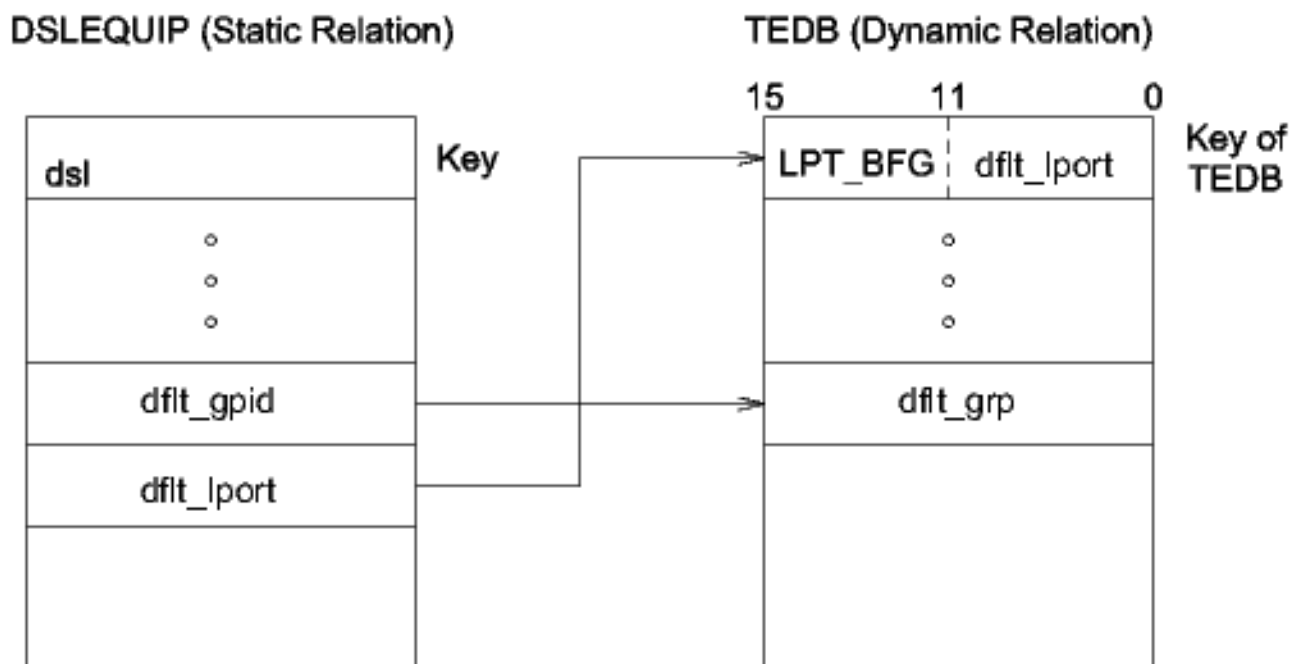


Figure 1 (DSLEQUIP-1) Data Structures of the DSLEQUIP Audit

3. AUDIT CHECKS

The AUdrstatic (static driver) is used to step through the RLDSLEQUIP relation. For every tuple, the driver calls the DSLEQUIP static-to-dynamic existence check function. For DSL subscribed default service, the DSLEQUIP audit ensures that the default service TEDB is initialized. If the TEDB is missing, the DSLEQUIP audit inserts the TEDB and links it to the owning D-channel port block (DPB). For every DSLEQUIP tuple, there should exist a corresponding ISMDB tuple. The key of ISMDB tuple consists of dsl group and member from DSLEQUIP. The ISMDB is used for all application that support ISDN protocols. Any missing ISMDB tuple will be inserted.

4. ERROR HANDLING

Any missing TEDB tuple is initialized and linked to the DPB and any missing ISMDB tuple is just initialized.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The DSL has default service feature but the corresponding TEDB tuple does not exist or for each tuple in DSLEQUIP, there should exist corresponding ISMDB tuple.
Possible Error Effect:	Users cannot originate calls by using default service or no protocol monitor available on this channel.
Corrective Action Taken:	If the TEDB tuple, the tuple is initialized and linked to DPB (D-channel process block). If the ISMDB tuple, the tuple is just initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the missing dynamic (TEDB or ISMDB).
<i>Bad Data =</i>	Relation is of the missing dynamic (TEDB or ISMDB).
<i>Good Data =</i>	Expected dynamic (TEDB or ISMDB) key.
<i>Logical Index =</i>	Port name of DSL.
<i>Dump =</i>	Dump of the rDSLEQUIP tuple.

DSLOWN

Software Release: 5E10 and later

1. DIGITAL SUBSCRIBER LINE (DSL) OWNERSHIP MECHANISM (DSLOWN) AUDIT

The DSLOWN audit verifies that the DSL digital subscriber loop-resources, that are associated with an *ANSI*[®] U-CARD, are allocated and queued promptly. The DSLOWN audit also verifies that no one resource is allocated for more than one user at any time. The audit performs this by checking the entries in every DSLOWN tuple that exist in the system. The DSLOWN tuples exist only when some maintenance activity is taking place on a DSL resource.

2. DATA STRUCTURES

The DSLOWN audit, audits the data structure rDSLOWN. This data structure contains the information regarding the DSL resource that is presently worked on. The other data structures referenced in this audit are the rCKTDATA and rPCBLA.

3. AUDIT CHECKS

The main purpose of the DSLOWN routine audit flow is to sequence through every existing DSLOWN tuple and verify its legitimacy. This verification is performed by the following steps for every existing tuple.

- (1) For every existing DSLOWN tuple, a static CKTDATA tuple should also exist. The "cktname" field of the CKTDATA tuple should be the same as the "cktname" field of the DSLOWN tuple. Also, the "ckttype" field of the CKTDATA tuple should be of the *ANSI*[®] or spare *ANSI*[®] type circuit enumeration.
- (2) In any existing DSLOWN tuple, at least one live Process Identifier (PID) should exist in one of the five structures of that DSLOWN tuple. (Refer to Figure 1.)
- (3) For every live PID that the DSLOWN audit locates in any of the structures, a PCBLA tuple should also exist. The PCBLA's "dslown" field should be set to "DBYES". The DSLOWN audit invokes the DRSTATIC driver to sequence through every existing DSLOWN tuple.

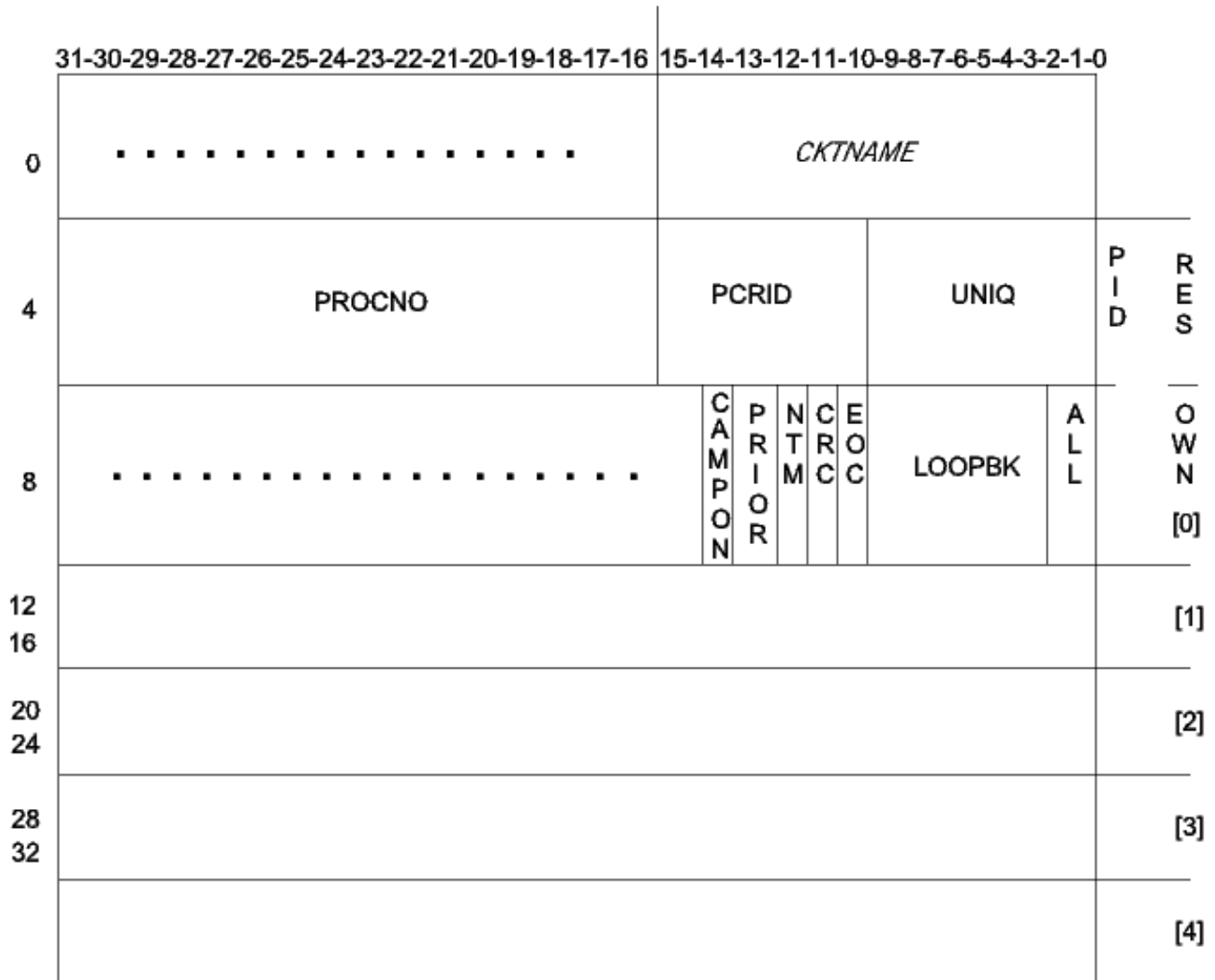


Figure 1 (DSLOWN-1) rIDSLOWN Tuple

4. ERROR HANDLING

Whenever the audit locates a problem with a rIDSLOWN tuple, the audit deletes that tuple and the corresponding related audits that are scheduled to run, to recover the other data structures. (PCBLA audit to recover rIPCBLA tuple). For example, the Process Control Block Linkage Area (PCBLA) audit recovering the rIPCBLA tuple.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The circuit name field in the rIDSLOWN tuple (ckname) contains a circuit name that is not of type <i>ANSI</i> [®] U-card (PCU_ANSI) or <i>ANSI</i> [®] spare U-card (PCSPRU_ANSI) as specified by the circuit type (type) field in the corresponding

	static rICKTDATA tuple.
Possible Error Effect:	Terminal maintenance (TM) may try to seize a DSL resource that does not exist causing asserts.
Corrective Action Taken:	The rIDSLOWN tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIDSLOWN tuple found in error.
<i>Bad Data</i> =	The circuit type field (type) from the rICKTDATA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name field (cktname) from the rIDSLOWN tuple found in error.
<i>Dump 1</i> = 1	The rIDSLOWN tuple found in error.
<i>Dump 2</i> = 1	The process control block (OSbpcb[]) of the owning process of this rIDSLOWN tuple (res_own[].pid.procno).

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	A rIDSLOWN tuple was found to have a PID with no matching rIPCBLA tuple.
Possible Error Effect:	This will cause a resource to be allocated to more than one user, which is not to be allowed.
Corrective Action Taken:	Delete the rIDSLOWN tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDSLOWN tuple.
<i>Bad Data</i> =	The number of the dead PID found.
<i>Good Data</i> =	The number of the RES_OWN structure within the rIDSLOWN tuple.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rIDSLOWN tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIDSLOWN tuple exists with no matching rICKTDATA tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Delete the extra rIDSLOWN tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDSLOWN tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rIDSLOWN tuple.

Table 4 Error Code: NOPBCBLA

Error Code:	NOPBCBLA
Error Description:	A rIDSLOWN tuple was found to have a matching rIPCBLA tuple, but the dslown filed in the rIPCBLA was not set to DBYES.
Possible Error Effect:	Eventual purge of the process owning that tuple if still alive.
Corrective Action Taken:	Delete the rIDSLOWN tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIDSLOWN tuple.
<i>Bad Data</i> =	The value the dslown field of the rIPCBLA.
<i>Good Data</i> =	The number of the RES_OWN structure within the rIDSLOWN.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rIDSLOWN tuple.

DT

Software Release: 5E10 and later

1. DATA TRANSPORT (DT) AUDIT

The DT audit is part of the system integrity audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The audit checks data transport data structures in the switching module processor (SMP).

3. AUDIT CHECKS

The audit checks the master copy of the data against data transports private copy.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CSAERR

Error Code:	CSAERR
Error Description:	The data transport DTSABMRCV bit does not agree with the value found in the ODD office parameter Glsabm.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in Glsabm.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: DSL_RATE

Error Code:	DSL_RATE
Error Description:	The data transport DTRATEPRD bit does not agree with the value found in the SMP.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the SMP rate period.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in the SMP.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: LEV2_CL

Error Code:	LEV2_CL
Error Description:	The data transport DTLEV2ERR bit does not agree with the value found in the ODD office parameter Gll2err.
Possible Error Effect:	The PHs are using the wrong value for their threshold.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in Gll2err.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 4 Error Code: LNK_SM_NUM

Error Code:	LNK_SM_NUM
Error Description:	The data transport DTSMNUM bit does not agree with the value found in the ODD office parameter G1_pcid.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in G1_pcid.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 5 Error Code: LVLNUM

Error Code:	LVLNUM
Error Description:	The data transport DTPSNUM bit does not agree with the value found in the ODD office parameter Glpsn.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in Glpsn.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 6 Error Code: MINDX

Error Code:	MINDX
Error Description:	The data transport DTXMITHRSH bit does not agree with the value found in the ODD office parameter Glretrans.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in Glretrans.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 7 Error Code: OFCNAME

Error Code:	OFCNAME
Error Description:	The data transport DTFCSRCV bit does not agree with the value found in the ODD office parameter G1badfcs.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value read from data transport.
<i>Good Data =</i>	The value found in G1badfcs.
<i>Logical Index =</i>	NA.

<i>Dump</i> =	NA.

Table 8 Error Code: P_TMPINH1

Error Code:	P_TMPINH1
Error Description:	The data transport DTPHAUINH bit does not agree with the PH audit inhibit status which is in the SMP.
Possible Error Effect:	The PH audit inhibit and allow status does not agree with the SMP status.
Corrective Action Taken:	The data transport value is updated to agree with the SMP status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Value read from data transport.
<i>Good Data</i> =	The value found in the SMP.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 9 Error Code: P_TMPINH2

Error Code:	P_TMPINH2
Error Description:	The data transport DTPIAUINH bit does not agree with the PI audit inhibit status which is in the SMP.
Possible Error Effect:	The PI audit inhibit and allow status does not agree with the SMP status.
Corrective Action Taken:	The data transport value is updated to agree with the SMP status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Value read from data transport.
<i>Good Data</i> =	The value found in SMP.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 10 Error Code: SEG_CL

Error Code:	SEG_CL
Error Description:	The data transport DTSEGSIZE bit does not agree with the value found in the ODD office parameter Glsegsz.
Possible Error Effect:	The PHs are using the wrong value.
Corrective Action Taken:	The data transport value is updated to agree with the ODD office parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Value read from data transport.
<i>Good Data</i> =	The value found in Glsegsz.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 11 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit has been entered by audit control to clean up after a reset.
Possible Error Effect:	Causes resets and or asserts.
Corrective Action Taken:	The audit is restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

DYNPICB

Software Release: 5E12 and later

1. DYNAMIC PICB (DYNPICB) AUDIT

Peripheral interface control buses pass control information from the peripheral units to the module controller/time-slot interchanger (MC/TSI). The information passed over the bus includes control and timing configurations. The DYNPICB audit ensures that the dynamic peripheral interface control bus relation (RLDYNPICB) is identical to the static peripheral interface control bus relation (RLPICB).

2. DATA STRUCTURES

For details on the fields contained in the RLDYNPICB relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The static and dynamic copies of rPICB are not identical.
Possible Error Effect:	May not be able to properly set up calls. Depending on the type of peripheral interface control bus (PICB) that is wrong, it might also cause hardware problem.
Corrective Action Taken:	The static version of rPICB is copied into the dynamic version.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Beginning address of the dynamic tuple that is found to be in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	PICB number.
<i>Dump 1 =</i>	Dynamic rPICB tuple under audit.
<i>Dump 2 =</i>	Static rPICB tuple that is considered correct.

15. E - G

E911DB

Software Release: 5E10 and later

1. ENHANCED 911 DATA BLOCK (E911DB) AUDIT

The enhanced 911 data block (E911DB) audit audits the RLE911DB relation, used for the enhanced 911 (E911) feature. 911 service is a Bell Operation Company (BOC) basic exchange service that provides access to emergency bureaus by dialing three digits "911" from any telephone. The major advantages of E911 over the basic 911 (B911) service on the switch are:

- The ability to selectively route the call to the appropriate agency, based on the calling party's automatic number identification (ANI).
- Delivery of caller's ANI to public safety answering point (PSAP) (location or equipment at the location where emergency attendants receive 911 calls) for display at an attendant position.
- Support of the E911 PSAP which enables subsequent retrieval of the caller's name and location from an automatic location identification (ALI)/data management system (DMS) for display at the attendant position.
- The ability to selectively transfer the call to up to six directory numbers (DNs) based upon the caller's ANI.

The E911 capability provides full 1A E911 service transparency on the switch. The E911DB audit runs in the OSDS-M environment in the segmented mode.

2. DATA STRUCTURES

The primary data structure for this audit is the RLE911DB relation. It is a dynamic, indirect access sequential keyed relation that is used to store the alternate PSAP DNs, selective transfer DNs, the ANI, the numbering plan digit (NPD), the emergency serving central office (ESCO) and the flash indicator. It is used by the originating terminal process (OTP) and the terminating terminal process (TTP) of the E911 calls. All idle rLE911DB tuples must be linked with owner-member linkages to the idle list owned by rIDB_HEADC tuple. (Refer to Figure 1.) All busy rLE911DB tuples must be linked one-to-one to rPCBLA tuples. (Refer to Figure 2.) Global parameter Gle911cnt indicates the number of rLE911DB tuples to be populated during initialization.

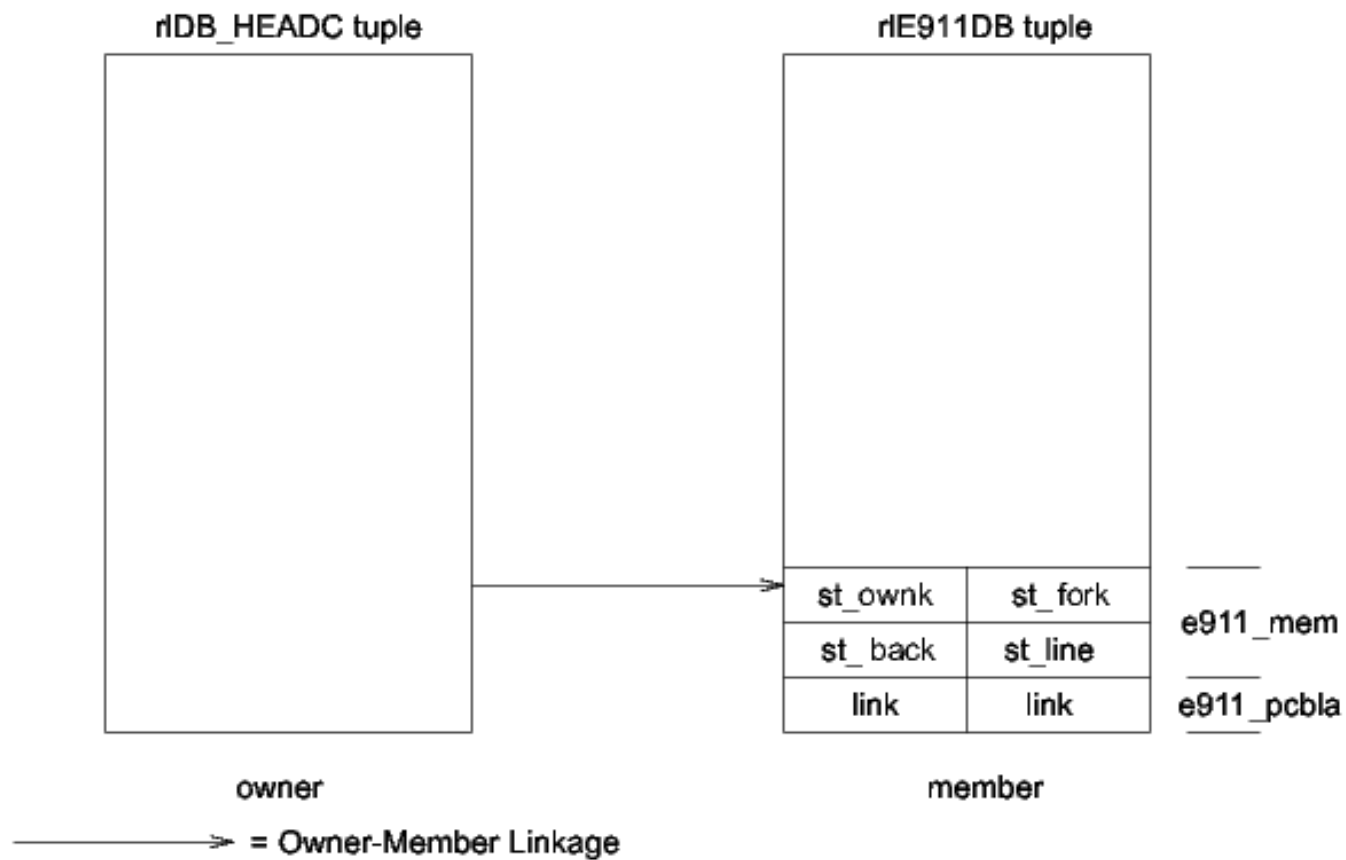


Figure 1 (E911DB-1) Owner-Member Linkages with rIDB_HEADC Tuple

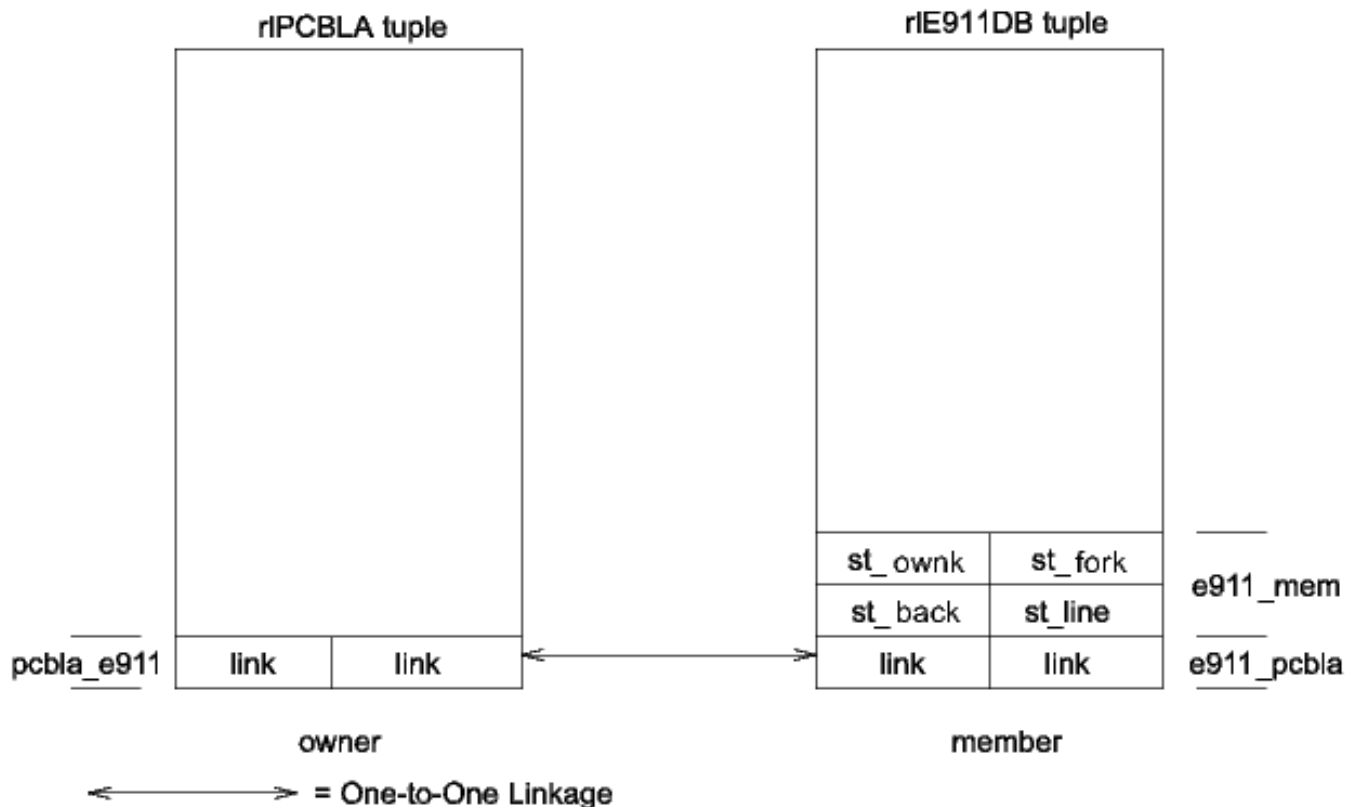


Figure 2 (E911DB-2) One-to-one Linkages with rIPCBLA

3. AUDIT CHECKS

The E911DB audit invokes the DRINASEK driver to make the linkages and semantic checks for each rIE911DB tuple. It also uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given.

IDLE_LINK = Check validity of linkages when rIE911DB tuple is on idle list.
 NOPCBLA = Check validity of linkages when rIE911DB tuple is used in a call and linked to rIPCBLA tuple.
 ZERO_LINK = Check validity of owner field of rIE911DB tuple.

NOTE: Normally when a rIPCBLA tuple (process) is involved with another data block in setting up a call, when we find any kind of linkage errors, we put the data block to free list and purge the process to which it was linked. Since rIE911DB tuple is used only to set up the E911 call, if we find any linkage errors in rIE911DB tuple, only the rIE911DB tuple will be put back to the free list and the process (rIPCBLA tuple) involved with that call will stay. The linkage field linking rIPCBLA tuple to the rIE911DB tuple will be nulled.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The owner field of the rIE911DB tuple, e911db_ptr→e911_mem.st_ownk, indicates that it is neither on the idle list nor linked to a rIPCBLA tuple. The rIE911DB tuple should be linked to rIDB_HEADC tuple when idle and should be linked to rIPCBLA tuple when in use.
Possible Error Effect:	Lost resource which may result in no e911 calls getting originated.
Corrective Action Taken:	The tuple will be put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIE911DB tuple being audited.
<i>Bad Data =</i>	The value of the owner field of the rIE911DB tuple, e911db_ptr→e911_mem.st_ownk.
<i>Good Data =</i>	DBE911IDLE, the number of the rIDB_HEADC tuple to which all idle rIE911DB tuples should be linked.
<i>Logical Index =</i>	Key of the rIE911DB tuple being audited.
<i>Dump =</i>	The rIE911DB tuple being audited.

Table 2 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	A rIE911DB tuple is marked busy (rIE911DB.e911_mem.st_ownk == DALNKNUL) but the one-to-one linkage to an associated rIPCBLA tuple is not marked (rIE911DB.e911_mem.st_link == DALNKNUL). For all busy rIE911DB tuples, there must be a linkage to an associated rIPCBLA tuple.
Possible Error Effect:	Possible E911 resource exhaustion resulting in E911 origination failures. Possible switch resource overload alarms.
Corrective Action Taken:	The tuple is recovered to the RLDB_HEADC idle list head cell (DBE911IDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIE911DB tuple being audited.
<i>Bad Data =</i>	DALNKNUL.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIE911DB tuple being audited (rIE911DB.e911db).
<i>Dump =</i>	The contents of the rIE911DB tuple being audited.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The owner field of the rIE911DB tuple, e911db_ptr→e911_mem.st_ownk, indicates that this should be linked to idle list (to a rIDB_HEADC tuple), but other linkage(s) for this rIE911DB tuple are found to be nonnull.
Possible Error Effect:	An e911 call may be purged by audit or this is a lost resource.
Corrective Action Taken:	The tuple will be put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIE911DB tuple being audited.
<i>Bad Data =</i>	The value of the return variable from idle state linkage checks.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIE911DB tuple being audited.
<i>Dump =</i>	The rIE911DB tuple being audited.

ECD

Software Release: 5E10 and later

1. EQUIPMENT CONFIGURATION DATABASE (ECD) AUDIT

The equipment configuration database (ECD) audit verifies the description of hardware and selected software resources that are stored in the ECD. The audit is divided into three parts (called members), each invoked by a separate input message.

Three copies of the ECD are kept all of the time, one in core memory and two on disk.

ECD 1 produces a structural audit of the incore ECD.

ECD 2 produces a structural audit of either the operating system (ECD) or the application (APPLCD).

ECD 3 produces a raw data audit of the incore ECD.

The ECD audits are routinely scheduled as timed audits in the correcting disk ECD's mode.

Auditing the ECD requires correct information in the low-level access (LLA) system. Therefore, ECD audits may invoke an LLA audit.

SIM requests ECD audits regularly to examine the ECDs. The user can invoke these audits by using the AUD:ECD input messages described.

NOTE: Do not run an ECD audit while a recent change is being applied.

1.1 Requesting ECD Audits

To request an audit of the incore ECD, use the following input message:

```
AUD:ECD=a[:b|:c|:b,c];
```

To request a structural audit of the operating system portion of the disk ECD, use the following input message:

```
AUD:ECD=2,INS="ecd"[:b|:c|:b,c];
```

To request a structural audit of the application portion of the disk ECD, use the following input message:

```
AUD:ECD=2,INS="appecd"[:b|:c|:b,c];
```

In all three input messages:

a = The type of incore audit. Valid values are:

1 = Specifies a structural audit.

3 = Specifies a raw data audit.

b = Correcting or non correcting. Valid values are:

CORR = Specifies that an audit should correct errors (default).

NCG = Specifies that an audit does not correct errors.

c	= The type of report. Valid values are:
DETL	= Generates a raw data report for each error found and an audit completion message (default). (Only a limited number of raw data reports are generated.)
SUM	= Generates only an audit completion message when the audit terminates. Raw data reports are not generated for any errors found.

To interpret system responses to the AUD:ECD input message, refer to the APP:AUD Appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD ECD Output Messages and Error Reports

The ECD audit prints an output message after each audit. It is produced by the system integrity output formatter and includes the information requested by the input message. For more detailed information about the AUD ECD output message, see the Output Messages manual (235-600-750).

Three categories of errors are reported by the AUD ECD output message: 3, 4, and 5.

Category 3 errors:

- Cause the audit to stop.
- Cannot be corrected by the audit.
- Error number starts with a 3.

Category 4 errors:

- Do not stop the audit.
- Cannot be corrected by the audit.
- Error number starts with a 4.

Category 5 errors:

- Do not stop the audit.
- Are corrected when the audit is in the correcting mode.
- Are not corrected when the audit is not in the correcting mode.
- Error number starts with a 5.

Table 1 is a list of error messages.

Table 1 ECD AUDIT ERROR MESSAGES

Category	Error Number	Meaning
3	3000	Cannot open incore ECD.
3	3001	malloc failure.
3	3002	read failure.
3	3003	Audit ECD has more records than incore ECD.

3	3004	Incore ECD has more records than audit ECD.
3	3005	lla_rdget failure for incore ECD to check RCV semaphore.
3	3006	Recent change in progress.
3	3007	lla_ainit failure for structural audit.
3	3008	lla_irel failure for incore ECD.
3	3009	lla_dbrel failure.
3	3010	Cannot open disk ECD.
3	3011	lla_nsgrid failure.
3	3012	getport failure.
3	3013	getseg failure.
3	3014	lla_iatt failure for incore ECD.
3	3015	lla_opensq failure for incore ECD.
3	3016	Audit copy corrupted.
3	3017	fopen failure.
3	3018	fread failure.
3	3019	fseek failure.
3	3020	fwrite failure.
3	3021	Disk ECD time stamp does not match incore ECD. Incorrect or missing audit ECD, request assistance in correcting this error.
3	3022	Driver PID notification table entries out of space.
3	3023	Unable to create segment because of insufficient swap space or there is insufficient user address space.
3	3024	Audits blocked during offline boot - try later.
4	4000	lla_rinfo failure.
4	4001	Incore RID does not match audit.
4	4002	lla_rdget failure.
4	4003	Record size of incore ECD does not match record size audit ECD.
4	4004	Record type of incore does not match audit ECD.
4	4005	lla_uprid failure.
4	4006	Unable to notify driver of local copy record changes.
4	4007	Possible disk ECD inconsistency (structure corrected).
4	4008	Structural audit cannot continue without previous corrections.
4	4009	UCB points down to a UCB which does not point back up.
4	4010	UCB has incorrect UP0 pointer for its side chain.
4	4011	UCB has incorrect UP1 pointer for its side chain.
4	4012	UCB points down to the beginning of a side chain loop.
4	4013	UCB should not have a side pointer.
4	4014	UCB points down to itself.
5	5000	Record corrected. Incore record data will be followed by audit record data.
5	5001	Record needs correction. Incore record data will be followed by audit record data.

The DATA1 through DATA4 fields are always zero in AUD ECD error messages. Additional information is provided in the supplementary data portion of each message. Complete lists of supplementary data messages may be found on the AUD ECD output message page and in Table 2.

If the audits are unable to interface with the audit control subsystem, AUD output messages cannot be issued. In that event, the supplementary data portion of each error message will be printed as a separate output message.

The LLA audit output messages come in three types:

- DIAG = Diagnostic only.
- RCVY = Error detection and associated recovery actions.
- UNRCVY = Detection of an unrecoverable error.

A percent character, "%", followed by an x, lx, d, and ld in an error message means that the appropriate value is substituted into the output message at this point.

- d = Integer decimal number.
- ld = Long (4-byte) decimal number.
- lx = Long (4-byte) hexadecimal number.

x = Integer hexadecimal number.

Table 2 LLA AUDIT ERROR MESSAGES

Error Code	Status	Message	Description
100	RCVY	STRAUD00	Incore database sanity ID overwritten with 0x%lx.
101	RCVY	STRAUD01	Offset to software demand paging (SDP) SPACE structure being reset from 0x%lx to 0x%x.
102	RCVY	STRAUD02	Offset to user data section being reset from 0x%lx to 0x%lx.
103	UNRCVY	STRAUD03	User rootid in DMLHEAD(=0x%lx) was reinitialized.
104	RCVY	STRAUD04	Startup database size being reset from 0x%lx to 0x%lx.
105	RCVY	STRAUD05	Space maxpage being reset from 0x%lx to 0x%lx.
106	RCVY	STRAUD06	Number of SDP space allocation stacks being reset from 0x%lx to 0x%lx.
107	RCVY	STRAUD07	Pointer to SDP stacks being reset from 0x%lx to 0x%lx.
108	RCVY	STRAUD08	SDP space pointer to environ being reset from 0x%lx to 0x%lx.
109	RCVY	STRAUD09	ENVIRON manager pointer being reset from 0x%lx to 0x%lx.
110	RCVY	STRAUD10	SDP manager frame pointer being reset from 0x%lx to 0x%lx.
111	RCVY	STRAUD11	SDP manager page ID being reset from 0x%lx to 0x0.
112	RCVY	STRAUD12	SDP manager use count being reset from 0x%lx to 0x1.
113	RCVY	STRAUD13	SDP manager next index being reset from 0x%lx to 0x1.
114	RCVY	STRAUD14	SDP manager previous index being reset from 0x%lx to 0x1.
115	RCVY	STRAUD15	SDP manager page ID being reset from 0x%lx to IDNULL.
116	RCVY	STRAUD16	SDP manager use count reset from 0x%lx to 0x0.
117	RCVY	STRAUD17	SDP manager next index being reset from 0x%lx to 0x0.
118	RCVY	STRAUD18	SDP manager previous index being reset from 0x%lx to 0x0.
200	DIAG	SDPAUD00	Stack entry 0x%lx next ID overwritten with 0x%lx.
201	DIAG	SDPAUD01	Stack entry 0x%lx on %d list size %ld out of range.
202	RCVY	SDPAUD02	Resetting depth of %d control stack from %ld to %ld.
203	RCVY	SDPAUD03	Resetting next ID of %d control stack from 0x%lx to 0x%lx.
204	RCVY	SDPAUD04	Stack entry 0x%lx next ID being reset from 0x%lx to 0x%lx.
205	RCVY	SDPAUD05	Resetting next ID of %d control stack from 0x%lx to 0x%lx.
206	DIAG	SDPAUD06	Four-byte stack entry 0x%lx top found to have incorrect value 0x%lx.
207	RCVY	SDPAUD07	Resetting top itemid of 4-byte stack from 0x%lx to 0x%lx.
208	RCVY	SDPAUD08	Resetting depth of 4-byte stack from %ld to %ld.
209	RCVY	SDPAUD09	Resetting top of 4-byte stack from 0x%lx to 0x%lx.
210	RCVY	SDPAUD10	Four-byte stack entry 0x%lx next ID being reset from 0x%lx to 0x%lx.
300	UNRCVY	DMLAUD00	Unable to load DMLHEAD structure with SDP.
301	RCVY	DMLAUD01	DML sanity ID at itemid = 0 was overwritten with 0x%lx.
302	RCVY	DMLAUD02	DML page size overwritten with %d, should be %d.
303	RCVY	DMLAUD03	Number of RIDs in ridblk per page overwritten with %d, should be %d.
304	DIAG	DMLAUD04	Start of set header list found to have incorrect value 0x%lx.
305	DIAG	DMLAUD05	Set start points to bad set header at 0x%lx.
306	DIAG	DMLAUD06	End of set header list overwritten with 0x%lx.
307	DIAG	DMLAUD07	Set end points to bad set header at 0x%lx.
308	RCVY	DMLAUD08	Set start itemid being reset to 0x%lx from 0x%lx.
309	RCVY	DMLAUD09	Set end itemid being reset to 0x%lx from 0x%lx.
310	DIAG	DMLAUD10	Start of RID block list found to have incorrect value 0x%lx.
311	DIAG	DMLAUD11	RID block start points to bad RID block header at 0x%lx.
312	DIAG	DMLAUD12	End of RID block list found to have incorrect value 0x%lx.
313	DIAG	DMLAUD13	RID block end points to bad RID block at 0x%lx.
314	RCVY	DMLAUD14	RID block start itemid reset to 0x%lx from 0x%lx.
315	RCVY	DMLAUD15	RID block end itemid being reset to 0x%lx from 0x%lx.
316	RCVY	DMLAUD16	SPACE page size found to have incorrect value %d, should be %d.
317	RCVY	DMLAUD17	SPACE curleft being reset from 0x%lx to 0x%lx.
400	RCVY	SETAUD00	CSet header sanity ID at 0x%lx found to have incorrect value 0x%lx.
401	RCVY	SETAUD01	Previous link field in set header at 0x%lx being reset from 0x%lx to 0x%lx.
402	RCVY	SETAUD02	Next link field in set header at 0x%lx being reset from 0x%lx to 0x%lx.
500	RCVY	RBLAUD00	RID block sanity ID at 0x%lx overwritten with 0x%lx.
501	RCVY	RBLAUD01	Previous link field in RID block at 0x%lx being reset from 0x%lx to 0x%lx.
502	RCVY	RBLAUD02	Next link field in RID block at 0x%lx being reset from 0x%lx to 0x%lx.
600	RCVY	BCKAUD00	Record sanity ID at 0x%lx found to have incorrect value 0x%lx.
601	RCVY	BCKAUD01	Record at 0x%lx back link being reset from 0x%lx to 0x%lx.
602	RCVY	BCKAUD02	RID block slot at 0x%lx being reset from 0x%lx to 0x%lx.
603	RCVY	BCKAUD03	RID block slot 0x%lx bad, being reset from 0x%lx to 0x%lx.
700	RCVY	ORPAUD00	Resetting ridblk slot at 0x%lx to 0x%lx from 0x%lx.
800	RCVY	FREAUD00	RID block 0x%lx being put on free RID block list at head.
801	RCVY	FREAUD01	RID block 0x%lx on free list has no free slots.
900	DIAG	FRLAUD00	RID block 0x%lx next free link found to have incorrect value 0x%lx.

901	DIAG	FRLAUD01	dmlhead link to start of RID block free list found to have incorrect value 0x%lx.
1000	RCVY	FRLAUDR00	RID block 0x%lx next free link being reset from 0x%lx to 0x%lx.
1001	RCVY	FRLAUDR01	Resetting free RID block ID in dmlhead from 0x%lx to 0x%lx.
1200	UNRCVY	SHDAUD00	Set 0x%lx ordinal in both header and accessid bad.
1201	RCVY	SHDAUD01	Set 0x%lx ordinal being reset from 0x%x to 0x%x.
1202	RCVY	SHDAUD02	Set 0x%lx number of access methods being reset from %d to %d.
1203	RCVY	SHDAUD03	Set 0x%lx cluster flag being reset from 0x%x to 0x%x.
1204	RCVY	SHDAUD04	Set 0x%lx, access method %d, accessid being reset from 0x%lx to 0x%lx.
1205	RCVY	SHDAUD05	Set 0x%lx, access method %d, page size being reset from 0x%x to 0x%x.
1206	RCVY	SHDAUD06	Set 0x%lx, access method %d, key length being reset from %d to %d.
1300	RCVY	DMLAUDR0	Start and end RID block links being reset from 0x%lx, 0x%lx to 0x%lx, 0x%lx.
1301	RCVY	DMLAUDR1	Start and end set header links being reset from 0x%lx, 0x%lx to 0x%lx, 0x%lx.

1.3 Responding to AUD ECD Error Reports

If the ECD audit was run in the noncorrecting mode, run it again in the correcting mode. If the error persists, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

ECDOWN

Software Release: 5E10 and later

1. EQUIPMENT CONFIGURATION DATABASE OWNER (ECDOWN) AUDIT

The equipment configuration database owner (ECDOWN) audit verifies the entries in the ECD manager requester table. It ensures that no software resources are owned by an inactive or dead process. This information includes records of software resources assigned to the different active processes and devices.

The ECDOWN audit is scheduled to run routinely as a timed audit in the correcting mode.

ECDOWN has only one member and can be requested by either SIM or the user.

NOTE: Do not run an ECDOWN audit while a recent change or a manual diagnostic is being applied.

1.1 Requesting the ECDOWN Audit

To request an ECDOWN audit, use the following input message:

```
AUD:ECDOWN=1[:a|:b|:a,b];
```

where:

a = Correcting or non correcting. Valid values are:

CORR = Specifies that an audit should correct errors (default).

NCG = Specifies that an audit does not correct errors.

b = Type of report. Valid values are:

DETL = Generates a raw data report for each error found and an audit completion message (default). (Only a limited number of raw data reports are generated.)

SUM = Generates only an audit completion message when the audit terminates. Raw data reports are not created for any errors found.

To interpret system responses to the AUD:ECDOWN input message, refer to the APP:AUD Appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD ECDOWN Output Messages and Error Reports

The ECDOWN audit prints an output message after each audit. It is produced by the system integrity output formatter and includes the information requested by the input message. For more detailed information about the AUD:ECDOWN output message, see the Output Messages manual (235-600-750).

ECDOWN can find two categories of errors: 1 and 2. Category 1 errors are not correctable; category 2 errors are correctable. Table 1 gives the error codes and their meanings.

Table 1 ECDOWN AUDIT ERROR MESSAGES

Category	Error Number	Meaning
1	100	Unable to release special device file; RID = 0x%1x.
1	101	Unable to release UCB; RID = 0x%1x.

1	102	Improper reserved entity - not SDF or UCB.
2	200	Special device file released; RID = 0x%1x.
2	201	UCB released; RID = 0x%1x.
2	202	Special device file should be released; RID = 0x%1x.
2	203	UCB should be released; RID = 0x%1x.

1.3 Responding to AUD ECDOWN Error Reports

If the ECDOWN audit was run in the noncorrecting mode, rerun it in the correcting mode. If the error persists, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

EDSLTRKG

Software Release: 5E11 and later

1. ELECTRONIC DIRECTORY SERVICE TRUNK GROUP (EDSLTRKG) AUDIT

Calling name information delivery-primary rate interface (CNAM-PRI) is one of three components of the calling name for ISDN PRI feature on custom and national ISDN PRI. The CNAM-PRI component provides name information; either name characters, a private indication, or an unavailable indication, to a subscribed PRI interface at the time of call setup. The electronic directory service trunk group audit verifies the static data relation RLEDSLTRKG to the dynamic data relation RLPCNAM. For each tuple in RLEDSLTRKG where calling name is "allowed" and automatic message accounting is "usage sensitive", the relation RLPCNAM is searched for a tuple with the same group number. If an entry is not found in RLPCNAM then it is added and initialized.

2. DATA STRUCTURES

For details on the fields contained in the RLEDSLTRKG relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

EFD-A

Software Release: 5E10 - 5E12

1. ETHERNET FRAME DESCRIPTOR (EFD) AUDIT

Frames are used to transmit data through the ethernet network. ethernet frame descriptors (EFDs), elements of CMeTfduse[] or CMeRfduse[], are used to represent per-frame information such as type and ownership. EFDs are passed between data structures by linking them in different lists.

2. DATA STRUCTURES

The following data is used by the EFD audit:

CMeTfduse[]	= Transmit Ethernet Frame Descriptors.
CMeTfdidle	= Transmit EFD idle list.
CMeRfduse[]	= Receive Ethernet Frame Descriptors.
CMeRfdidle	= Receive EFD idle list.
CMqccscswcfg[]	= QUICC SCC software configuration profiles.
CMqccdb[]	= QUICC Channel data block.
CMscddb	= SONIC Channel data block.
CMeinq	= Ethernet interrupt/base level input queue head.
CMsipcb	= System integrity protocol control block.

3. AUDIT CHECKS

The EFD audit makes the following checks:

BADLINK	= Check an EFD's level 2 linkages.
BADUSE	= Check an EFD's usetype.
CORRUPT_IDLE	= Verify an idle EFD.
HEAD	= Verify idle list head.
IDLE_LINK	= Check an in-use list entry.
INVUSAGE	= Check for lost EFDs.
IS_NOTUSED	= Check an unused list entry.
RCVACCBK	= Check a receiving EFD's level 1 linkages and buffer.
QUE	= Verify CMeinq EFD ownership.
ZAP	= Check for system interrupt handling.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMeRfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMeRfdidle). One or more of the following is true of a non-idle (usetype != CMENFREE) EFD: - EFD.I2_next = CMNIL_NDX, but the EFD is not the last one on the list. [On the message handler (MH), the end of the list is stored in CMsipcb.l_ifd. On the switching module (SM, not valid for SM-2000), it is the same, unless the EFD.usetype = CMENINQ, in which case it is CMeinq.unload]. - EFD.I2_prev = CMNIL_NDX, but the EFD is not the first one on the list. (On the MH, the head of the list is stored in CMsipcb.f_ifd. On the ASM, it is the same, unless the EFD.usetype = CMENINQ, in which case it is CMeinq.load). - EFD.I2_next != CMNIL_NDX, but is not valid (is ≥ max). - EFD.I2_prev != CMNIL_NDX, but is not valid (is ≥ max). - EFD.I2_next is < max, but that EFD's I2_prev field != the current EFD's index. - EFD.I2_prev is < max, but that EFD's I2_next field != the current EFD's index.
Possible Error Effect:	The EFD is not available for use. If enough EFDs became unavailable, the SM (switching module) would lose any ethernet connections.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	EFD type (2 means TEFD, 3 means REFD).
<i>Bad Data</i> =	EFD's user bits (EFD.bf_owner).
<i>Good Data</i> =	EFD's usage type (EFD.usetype).
<i>Logical Index</i> =	EFD's index.
<i>Dump 1</i> =	The EFD.
<i>Dump 2</i> =	EFD indexed by EFD.I2_next if I2_next is < max; otherwise NA.
<i>Dump 3</i> =	EFD indexed by EFD.I2_prev if I2_prev is < max; otherwise NA.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An EFD's usage type is out-of-range (EFD.usetype ≥ CMENUMAX).
Possible Error Effect:	The out-of-range value may cause asserts. In any case, the EFD is not

	processed correctly. Any frame associated with the EFD may be dropped.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The EFD type: 2 means transmit EFD, an element of CMeTfduse[]. 3 means receive EFD, an element of CMeRfduse[].
<i>Bad Data</i> =	EFD's usage type (EFD.usetype).
<i>Good Data</i> =	CMENUMAX.
<i>Logical Index</i> =	EFD's index.
<i>Dump</i> =	The EFD.

Table 3 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMeTfduse[CMETNUMBUF]) and one for receive EFDs (CMeRfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMeTfdidle and CMeRfdidle). An idle EFD (EFD.usetype = CMENFREE) does not contain the expected default value in one or more of its attributes: • EFD.qindx is out-of-range (> the idle stack's toppt field). • The idle stack's efdindx[EFD.qindx] != the EFD's index. • One or more of the following are != CMNIL_NDX: • I1_prev • I1_next • I2_prev • I2_next • EFD.bf_owner != CME_NO_OWNER.
Possible Error Effect:	The EFD is not available for use or the EFD idle stack is corrupt. If enough EFDs became unavailable, the SM would lose ethernet connectivity.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	EFD type (2 means TEFD, 3 means REFD).
<i>Bad Data</i> =	EFD's user bits (EFD.bf_owner).
<i>Good Data</i> =	EFD's usage type (EFD.usetype).
<i>Logical Index</i> =	EFD's index.
<i>Dump</i> =	The EFD.

Table 4 Error Code: HEAD

Error Code:	HEAD
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMeTfduse[CMETNUMBUF]), and one for receive EFDs (CMeRfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMeTfdidle and CMeRfdidle).

	An EFD idle stack is corrupt. One or more of the following are true: - The idle stack toppt value is out-of-range ($> \text{max}$ or $< \text{min}$). - The idle stack toppt is $< \text{max}$ and the idle stack entry fdindx[toppt+1] $\neq \text{CMNIL\_NDX}$. - The idle stack toppt is $< \text{max}$ and the idle stack entry fdindx[toppt+1] $\neq \text{CMNIL\_NDX}$.
Possible Error Effect:	The EFDs can not be allocated. The SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the EFD idle stack (toppt = 0) and rebuilds the idle list (links all EFDs whose usetype is CMENFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The EFD type. Valid values are: 2 = Transmit EFDs. 3 = Receive EFDs.
<i>Bad Data</i> =	The idle stack toppt .
<i>Good Data</i> =	The value of "max".
<i>Logical Index</i> =	The minimum number of EFDs expected in the list.
<i>Dump</i> =	NA.

Table 5 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). An EFD in-use idle stack entry ($\text{index} \leq \text{idle stack toppt}$) is corrupt. One or more of the following is true ("idle_efd" refers to the contents of the idle stack entry fdindx[index]): $\text{idle\_efd} \geq \text{max}$ - The EFD indexed the contents of idle stack entry efdindx[idle_efd] and does not point back to the idle list ($\text{EFD.qindx} \neq \text{idle\_efd}$).
Possible Error Effect:	EFDs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry ($\text{index} = \text{idle stack toppt}$), the audit clears the idle stack entry (efdindx[index] = CMNIL_NDX) and decrements the count of idle EFDs (toppt is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets efdindx[index] = efdindx[toppt], relinks ($\text{EFD.qindx} = \text{index}$), clears the old entry (efdindx[CMbufidle.toppt] = CMNIL_NDX), and decrements the number of idle EFDs (decrements toppt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	EFD type (2 means TEFD, 3 means REFD).
<i>Bad Data</i> =	Value from idle stack entry (EFD index).
<i>Good Data</i> =	value of "max".
<i>Logical Index</i> =	Index into idle stack.

<i>Dump</i> =	NA.

Table 6 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMeTfduse[CMETNUMBUF]), and one for receive EFDs (CMeRfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMeTfdidle and CMeRfdidle). An EFD is owned only by level 1 (EFD.bf_owner & CM_L1_OWNER != 0, but EFD.bf_owner and CM_L2_OWNER = 0) and has not been touched for several minutes (EFD.au_detect = 1 and EFD.au_recover = 1).
Possible Error Effect:	The EFD is not available for use. If enough EFDs became unavailable, the SM could lose ethernet connectivity.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	EFD type (2 means TEFd, 3 means REFD).
<i>Bad Data</i> =	EFD's user bits (EFD.bf_owner).
<i>Good Data</i> =	EFD's usage type (EFD.usetype).
<i>Logical Index</i> =	EFD's index.
<i>Dump</i> =	The EFD.

Table 7 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMeTfduse[CMETNUMBUF]), and one for receive EFDs (CMeRfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMeTfdidle and CMeRfdidle). An unused EFD idle stack entry is not null (index < toppt and efdindx[index] != CMNIL_NDX).
Possible Error Effect:	EFDs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the EFD idle stack entry (efdindx[index] = CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	EFD type (2 means TEFd, 3 means REFD).
<i>Bad Data</i> =	Contents of the idle stack entry (efdindx[index]).
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into the idle stack.
<i>Dump</i> =	NA.

Table 8 Error Code: QUE

Error Code:	QUE
Error Description:	CMeinq contains invalid EFD linkages. One or more of the following is true: • CMeinq.load ≥ CMERNUMBUF and is != CMNIL_NDX. • CMeinq.unload ≥ CMERNUMBUF and is != CMNIL_NDX. • CMeinq.load < CMERNUMBUF but CMeRfduse[CMeinq.load].usetype != CMENINQ. • CMeinq.unload < CMERNUMBUF but CMeRfduse[CMeinq.unload].usetype != CMENINQ.

Possible Error Effect:	The queue of receive EFDs is corrupt. The SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears CMeinq (CMeinq.load and CMeinq.unload = CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMeinq.load.
<i>Good Data =</i>	CMeinq.unload.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 9 Error Code: RCVACCBK

Error Code:	RCVACCBK
Error Description:	An receive ethernet frame (REFD) descriptor, entry in CMerfduse[], is in use (EFD.usetype = CMENRCV), but is not linked correctly at level 1. That is, one or more of the following are true: - EFD.I1_channel != CMQCENET (this check is only for the SM). - EFD.I1_next = CMNIL_NDX, but CMCDBENTRY.rcv_unload != the EFD's index. - EFD.I1_prev = CMNIL_NDX, but CMCDBENTRY.rcv_load != the EFD's index. - EFD.I1_next != CMNIL_NDX, but is not valid (is ≥ max). - EFD.I1_prev != CMNIL_NDX, but is not valid (is ≥ max). - EFD.I1_next is < max, but CMerfduse[EFD.I1_next].I1_prev != the EFD's index. - EFD.I1_prev is < max, but CMerfduse[EFD.I1_prev].I1_next != the EFD's index. - EFD.bd does not contain a valid linkage to the buffer.
Possible Error Effect:	The SM may lose ethernet connectivity.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack (CMerfdidle).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	EFD type (2 means TEFD, 3 means REFD).
<i>Bad Data =</i>	EFD's user bits (EFD.bf_owner).
<i>Good Data =</i>	EFD's usage type (EFD.usetype).
<i>Logical Index =</i>	EFD's index.
<i>Dump 1 =</i>	The EFD.
<i>Dump 2 =</i>	EFD indexed by EFD.I1_next if I1_next is < max; otherwise NA.
<i>Dump 3 =</i>	EFD indexed by EFD.I1_prev if I1_prev is < max; otherwise NA.

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). The audit takes this error as a result of a write protect violation, out-of-range

	read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle stack entry (efdindx[] = CMNIL_NDX), if any, that was being audited when the stimulus occurred. • Recovers the EFD, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Message handler (MH): SMmhrunning (the current base-level loop job). SM: NA.
<i>Bad Data</i> =	EFD type being audited when the stimulus occurred (2 means transmit EFD, 3 means receive EFD).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Current EFD index (which may be out-of-range).
<i>Dump</i> =	The EFD.

EFD-B

Software Release: 5E13 and later

1. ETHERNET FRAME DESCRIPTOR (EFD) AUDIT

Frames are used to transmit data through the ethernet network. ethernet frame descriptors (EFDs), elements of CMeTfduse[] or CMerfduse[], are used to represent per-frame information such as type and ownership. EFDs are passed between data structures by linking them in different lists.

2. DATA STRUCTURES

The following data is used by the EFD audit:

CMeTfduse[]	= Transmit ethernet frame descriptors.
CMeTfdidle	= Transmit EFD idle list.
CMerfduse[]	= Receive ethernet frame descriptors.
CMerfdidle	= Receive EFD idle list.
CMqccscswcfg[]	= QUICC SCC software configuration profiles.
CMqccdb[]	= QUICC channel data block.
CMscddb	= SONIC channel data block.
CMeinq	= Ethernet interrupt/base level input queue head.
CMsipcb	= System integrity protocol control block.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanations in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMeTfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMeTfdidle and CMerfdidle). One or more of the following is true of a non-idle (<i>usetype</i> != CMENFREE) EFD: EFD.I2_next = CMNIL_NDX, but the EFD is not the last one on the list.

	[On the MH (message handler), the end of the list is stored in Cmsipcb.l_ifd. On the SM (switching module, not valid for SM-2000), it is the same, unless the EFD.usetype = CMENINQ, in which case it is CMeinq.unload]. - EFD.I2_prev = CMNIL_NDX, but the EFD is not the first one on the list. (On the MH, the head of the list is stored in Cmsipcb.f_ifd. On the ASM, it is the same, unless the EFD.usetype = CMENINQ, in which case it is CMeinq.load). - EFD.I2_next != CMNIL_NDX, but is not valid (is $\geq$ max). - EFD.I2_prev != CMNIL_NDX, but is not valid (is $\geq$ max). - EFD.I2_next is $<$ max, but that EFD's I2_prev field != the current EFD's index. - EFD.I2_prev is $<$ max, but that EFD's I2_next field != the current EFD's index.
Possible Error Effect:	The EFD is not available for use. If enough EFDs became unavailable, the SM (switching module) would lose any ethernet connections.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The EFD type. Valid values are: 2 = TEFD. 3 = REFD.
<i>Bad Data =</i>	EFD's user bits (EFD.bf_owner).
<i>Good Data =</i>	EFD's usage type (EFD.usetype).
<i>Logical Index =</i>	EFD's index.
<i>Dump 1 =</i>	The EFD.
<i>Dump 2 =</i>	EFD indexed by EFD.I2_next if I2_next is $<$ max otherwise NA.
<i>Dump 3 =</i>	EFD indexed by EFD.I2_prev if I2_prev is $<$ max otherwise NA.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An EFD's usage type is out-of-range (EFD.usetype $\geq$ CMENUMAX).
Possible Error Effect:	The out-of-range value may cause asserts. In any case, the EFD is not processed correctly. Any frame associated with the EFD may be dropped.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The EFD type. Valid values are: 2 = TEFD. 3 = REFD.
<i>Bad Data =</i>	EFD's usage type (EFD.usetype).
<i>Good Data =</i>	CMENUMAX.
<i>Logical Index =</i>	EFD's index.
<i>Dump =</i>	The EFD.

Table 3 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]) and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). An idle EFD (EFD.usetype = CMENFREE) does not contain the expected default value in one or more of its attributes: • EFD.qindx is out-of-range (> the idle stack's <i>toppt</i> field). • The idle stack's efdindx[EFD.qindx] != the EFD's index. • One or more of the following are != CMNIL_NDX: • <i>l1_prev</i> • <i>l1_next</i> • <i>l2_prev</i> • <i>l2_next</i> • EFD.bf_owner != CME_NO_OWNER.
Possible Error Effect:	The EFD is not available for use or the EFD idle stack is corrupt. If enough EFDs became unavailable, the SM would lose ethernet connectivity.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The EFD type. Valid values are: 2 = Transmit EFD, an element of CMetfduse[]. 3 = Receive EFD, an element of CMerfduse[].
<i>Bad Data</i> =	EFD's user bits (EFD.bf_owner).
<i>Good Data</i> =	EFD's usage type (EFD.usetype).
<i>Logical Index</i> =	EFD's index.
<i>Dump</i> =	The EFD.

Table 4 Error Code: HEAD

Error Code:	HEAD
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). An EFD idle stack is corrupt. One or more of the following are true: • The idle stack <i>toppt</i> value is out-of-range (> max or < min). • The idle stack entry <i>fdindx[toppt]</i> = CMNIL_NDX.

	The idle stack <i>toppt</i> is < max and the idle stack entry <i>fdindx[toppt+1] != CMNIL_NDX</i>.
Possible Error Effect:	The EFDs can not be allocated. The SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the EFD idle stack (<i>toppt</i> = 0) and rebuilds the idle list (links all EFDs whose <i>usetype</i> is CMENFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The EFD type. Valid values are: 2 = TEFD. 3 = REFD.
<i>Bad Data</i> =	The idle stack <i>toppt</i> .
<i>Good Data</i> =	The value of "max".
<i>Logical Index</i> =	The minimum number of EFDs expected in the list.
<i>Dump</i> =	NA.

Table 5 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). An EFD in-use idle stack entry (<i>index</i> ≤ idle stack <i>toppt</i>) is corrupt. One or more of the following is true ("idle_efd" refers to the contents of the idle stack entry <i>fdindx[index]</i>): <i>idle_efd</i> ≥ max - The EFD indexed the contents of idle stack entry <i>efdindx[idle_efd]</i> and does not point back to the idle list (<i>EFD.qindx != idle_efd</i>).
Possible Error Effect:	EFDs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry (<i>index</i> = idle stack <i>toppt</i>), the audit clears the idle stack entry (<i>efdindx[index] = CMNIL_NDX</i>) and decrements the count of idle EFDs (<i>toppt</i> is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets <i>efdindx[index] = efdindx[toppt]</i>, relinks (<i>EFD.qindx = index</i>), clears the old entry (<i>efdindx[CMbufidle.toppt] = CMNIL_NDX</i>), and decrements the number of idle EFDs (decrements <i>toppt</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The EFD type. Valid values are: 2 = Transmit EFDs. 3 = Receive EFDs.
<i>Bad Data</i> =	Value from idle stack entry (EFD index).
<i>Good Data</i> =	value of "max".
<i>Logical Index</i> =	Index into idle stack.
<i>Dump</i> =	NA.

Table 6 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). An EFD is owned only by level 1 (EFD.bf_owner & CM_L1_OWNER != 0, but EFD.bf_owner and CM_L2_OWNER = 0) and has not been touched for several minutes (EFD.au_detect = 1 and EFD.au_recover = 1).
Possible Error Effect:	The EFD is not available for use. If enough EFDs became unavailable, the SM could lose ethernet connectivity.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The EFD type. Valid values are: 2 = TEFD. 3 = REFD.
<i>Bad Data =</i>	EFD's user bits (EFD.bf_owner).
<i>Good Data =</i>	EFD's usage type (EFD.usetype).
<i>Logical Index =</i>	EFD's index.
<i>Dump =</i>	The EFD.

Table 7 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). An unused EFD idle stack entry is not null ($index < toppt$ and $efdindx[index] \neq CMNIL\_NDX$).
Possible Error Effect:	EFDs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the EFD idle stack entry ($efdindx[index] = CMNIL\_NDX$).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The EFD type. Valid values are: 2 = TEFD. 3 = REFD.
<i>Bad Data =</i>	Contents of the idle stack entry ($efdindx[index]$).
<i>Good Data =</i>	CMNIL_NDX.
<i>Logical Index =</i>	Index into the idle stack.
<i>Dump =</i>	NA.

Table 8 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs (CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle).

	One or more of the following is true of a non-idle (<i>usetype</i> != CMENFREE) EFD. - EFD.melnk is out of range ($0 \leq \text{melnk} < \text{SMELNUMELNK}$ is in range). - EFD.I3_next = CMNIL_NDX, but the EFD is not the last one on the list. - EFD.I3_prev = CMNIL_NDX, but the EFD is not the first one on the list. - EFD.I3_next != CMNIL_NDX, but is not valid ($\text{is} \geq \text{max}$). - EFD.I3_prev != CMNIL_NDX, but is not valid ($\text{is} \geq \text{max}$). - EFD.I3_next is $< \text{max}$, but that EFD's <i>I2_prev</i> field != the current EFD's index. - EFD.I3_prev is $< \text{max}$, but that EFD's <i>I2_next</i> field != the current EFD's index. The list owner can be determined by the value of CMextsysbl[melnk][0].melnk_serv: CMESRTCD: CMMhatpl3q <i>first</i> and <i>last</i>. CMSEEC: CMmhesl3q[melnk] <i>first</i> and <i>last</i>.
Possible Error Effect:	The EFD list is broken. This may cause asserts, lost resources, lost frames or other problems.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The EFD type. Valid values are:
	2 = TEFD.
	3 = REFD.
<i>Bad Data</i> =	EFD's user bits (EFD.bf_owner).
<i>Good Data</i> =	EFD's usage type (EFD.usetype).
<i>Logical Index</i> =	EFD's index.
<i>Dump 1</i> =	The EFD.
<i>Dump 2</i> =	EFD indexed by EFD.I3_next if <i>I3_next</i> is $< \text{max}$ otherwise NA.
<i>Dump 3</i> =	EFD indexed by EFD.I3_prev if <i>I3_prev</i> is $< \text{max}$ otherwise NA.

Table 9 Error Code: QUE

Error Code:	QUE
Error Description:	CMeinq contains invalid EFD linkages. One or more of the following is true: - CMeinq.load $\geq$ CMERNUMBUF and is != CMNIL_NDX. - CMeinq.unload $\geq$ CMERNUMBUF and is != CMNIL_NDX. - CMeinq.load $<$ CMERNUMBUF but CMerfduse[CMeinq.load].usetype != CMENINQ. - CMeinq.unload $<$ CMERNUMBUF but

	CMerfduse[CMeinq.unload].usetype != CMENINQ.
Possible Error Effect:	The queue of receive EFDs is corrupt. The SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears CMeinq (CMeinq.load and CMeinq.unload = CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMeinq.load.
<i>Good Data =</i>	CMeinq.unload.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 10 Error Code: RCVACCBK

Error Code:	RCVACCBK
Error Description:	An receive ethernet frame (REFD)descriptor , entry in CMerfduse[], is in use (EFD.usetype = CMENRCV), but is not linked correctly at level 1. That is, one or more of the following are true: - EFD.I1_channel != CMQCENET (this check is only for the SM). - EFD.I1_next = CMNIL_NDX, but CMCDBENTRY.rcv_unload != the EFD's index. - EFD.I1_prev = CMNIL_NDX, but CMCDBENTRY.rcv_load != the EFD's index. - EFD.I1_next != CMNIL_NDX, but is not valid (is $\geq$ max). - EFD.I1_prev != CMNIL_NDX, but is not valid (is $\geq$ max). - EFD.I1_next is < max, but CMerfduse[EFD.I1_next].I1_prev != the EFD's index. - EFD.I1_prev is < max, but CMerfduse[EFD.I1_prev].I1_next != the EFD's index. - EFD.bd does not contain a valid linkage to the buffer.
Possible Error Effect:	The SM may lose ethernet connectivity.
Corrective Action Taken:	The audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the EFD idle stack (CMerfdidle).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The EFD type. Valid values are: 2 = TEFD. 3 = REFD.
<i>Bad Data =</i>	EFD's user bits (EFD.bf_owner).
<i>Good Data =</i>	EFD's usage type (EFD.usetype).
<i>Logical Index =</i>	EFD's index.
<i>Dump 1 =</i>	The EFD.
<i>Dump 2 =</i>	EFD indexed by EFD.I1_next if I1_next is < max, otherwise NA.
<i>Dump 3 =</i>	EFD indexed by EFD.I1_prev if I1_prev is < max otherwise NA.

Table 11 Error Code: ZAP

Error Code:	ZAP
Error Description:	There are two arrays of EFDs. One is for transmit EFDs (CMetfduse[CMETNUMBUF]), and one for receive EFDs

	(CMerfduse[CMERNUMBUF]). The error address value indicates which array was being checked when the error was detected. Each array is also associated with it's own stack of idle elements (CMetfdidle and CMerfdidle). The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle stack entry (<i>efdindx[]</i> = CMNIL_NDX), if any, that was being audited when the stimulus occurred. • Recovers the EFD, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the EFD from any other data structures, sets all fields to default values, and puts the EFD on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Message handler (MH): SMmhrunning (the current base-level loop job). SM: NA.
<i>Bad Data =</i>	The EFD type being audited when the stimulus occurred. Valid values are: 2 = Transmit EFD. 3 = Receive EFD.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Current EFD index (which may be out-of-range).
<i>Dump =</i>	The EFD.

EISCALL

Software Release: 5E10 and later

1. EXTERNAL INFORMATION SYSTEM CALL (EISCALL) AUDIT

An rEISCALL tuple contains data associated with the external information system (EIS) interface for a listings services call assigned to a call processing data links (CPDLs).

2. DATA STRUCTURES

The data structure of EISCALL audit is the EIS call relation (RLEISCALL). When the rEISCALL tuples are not handling calls they are connected to rEIS_LNK idle head cell (DBEISIDLE) and if they are in use, they are linked to rEIS_LNK CPDLs.

3. AUDIT CHECKS

The EISCALL audit invokes the DRINASEK driver to execute the syntactic checks, and invoke the semantic function. For each tuple, the EISCALL audit makes the following checks after the driver has verified the linkages. For details regarding each error code, see the comment block given.

EISTP = Check rEISCALL terminal process against rEIS_LNK.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: EISTP

Error Code:	EISTP
Error Description:	The rEISCALL terminal process (eistp) differs from rEIS_LNK terminal process (eistp). Therefore the call is not connected to the correct CPDLs.
Possible Error Effect:	In the worst case no EISCALL tuples are available.
Corrective Action Taken:	rEISCALL tuple is initialized and placed on the idle list (DBEISIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEISCALL tuple.
<i>Bad Data</i> =	rEISCALL terminal process (eistp).
<i>Good Data</i> =	rEIS_LNK terminal process (eistp).
<i>Logical Index</i> =	Logical key of rEISCALL (smcid) tuple.
<i>Dump</i> =	rEISCALL tuple.

EISLKSEL

Software Release: 5E10 and later

1. EXTERNAL INFORMATION SYSTEM LINK SELECT (EISLKSEL) AUDIT

An rEISLKSEL tuple contains data associated with the external information system (EIS) interface which contains the number of available call processing data links (CPDLs). Its purposes is to indicate which of the 16 EIS CPDLs on a specific EIS switching module (SM) are available for assignment of new calls for a particular EIS.

2. DATA STRUCTURES

The data structure for EISLKSEL audit is the EIS link select relation (RLEISLKSEL). This audit runs on SMs which contains OPSMDAAC package.

3. AUDIT CHECKS

The EISLKSEL audit invokes the DRINDEX driver to execute the syntactic checks and then invoke the semantic function.

For each tuple, the EISLKSEL audit makes the following check after the driver has verified the keys. For details regarding each error code, see the comment blocks given.

AVAIL_CNT = Verify the number of CPDLs that is available.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT
Error Description:	The number of call processing data link (rEISLKSEL.avail_cnt) available for EIS is invalid.
Possible Error Effect:	An available CPDLs might not be taken.
Corrective Action Taken:	Update the available count (rEISLKSEL.avail_cnt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEISLKSEL tuple.
<i>Bad Data</i> =	rEISLKSEL old available count (avail_cnt).
<i>Good Data</i> =	rEISLKSEL newly calculated available count (avail_cnt).
<i>Logical Index</i> =	Logical key of rEISLKSEL (eis_id) tuple.
<i>Dump</i> =	rEISLKSEL tuple.

EISLNK

Software Release: 5E10 and later

1. EXTERNAL INFORMATION SYSTEM LINK MESSAGE DATA (EISLNK) AUDIT

An rEIS_LNK tuple contains data associated with the external information system (EIS) interface. RLEIS_LNK contains a tuple for each assigned call processing data links (CPDLs) on a switching module (SM). It is keyed by the SM relative CPDLs number (0 through 15).

2. DATA STRUCTURES

The data structure for EISLNK audit is the EIS link message data relation (RLEIS_LNK). RLEIS_LNK provides a set-owner relationship with rEISCALL tuples. A logical key of negative 1 (DBEISIDLE), the idle list, contains the idle list for rEISCALL tuples. Each of the rEISCALL tuples represent a call associated with the respective CPDLs.

3. AUDIT CHECKS

The EISLNK audit invokes the DRLINK driver to execute the syntactic checks and then invokes the semantic function. For each tuple, the EIS_LNK audit makes the following checks after the driver has verified the linkages. For details regarding each error code, see the comment blocks given.

CAL_CNT	= Check if the call count is valid.
EISID	= Check if the EISID is correct.
ESSWID	= Check if the EIS Switch ID is valid.
EISTP	= Check if the EIS Terminal process is valid.
EISVENDOR	= Check if the EIS vendor is correct.
MSG_SET_ID	= Check if the message set id is correct.
MX_CALL_LNK	= Check if the maximum number of calls are correct.
MX_LNK_CAP	= Check if the maximum link capacity is correct.
NOTZERO	= Check if the tuple attributes are initialized to 0.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: CAL_CNT

Error Code:	CAL_CNT
Error Description:	The RLEIS_LNK audit call count is invalid for the number of linked calls.
Possible Error Effect:	The number of available call processing data link is incorrect.
Corrective Action Taken:	Update the call count.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEIS_LNK tuple.
<i>Bad Data</i> =	Old EIS_LNK (call_cnt) call count.
<i>Good Data</i> =	Newly calculated EIS_LNK call count.

<i>Logical Index</i> =	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump</i> =	rEIS_LNK tuple.

Table 2 Error Code: EISID

Error Code:	EISID
Error Description:	Verify that the EISID is equal to X_NAME in static relation rEISDSL.
Possible Error Effect:	The EIS ID is out of range.
Corrective Action Taken:	Update EISID with static relation rEISDSL x_name.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEIS_LNK tuple.
<i>Bad Data</i> =	rEIS_LNK eis_id.
<i>Good Data</i> =	rEISDSL x_name.
<i>Logical Index</i> =	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump</i> =	rEIS_LNK tuple.

Table 3 Error Code: EISSWID

Error Code:	EISSWID
Error Description:	rEIS_LNK EIS Switch ID differs from static rEIS_DATA EIS Switch ID.
Possible Error Effect:	The EIS switch id contains the wrong value.
Corrective Action Taken:	Update rEIS_LNK eis_swid with rEIS_DATA eis_swid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEIS_LNK tuple.
<i>Bad Data</i> =	rEIS_LNK EIS switch id (eis_swid).
<i>Good Data</i> =	rEIS_DATA EIS switch id (eis_swid).
<i>Logical Index</i> =	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump</i> =	rEIS_LNK tuple.

Table 4 Error Code: EISTP

Error Code:	EISTP
Error Description:	The EIS terminal process is not the same as the feature data block terminal process.
Possible Error Effect:	Invalid terminal process.
Corrective Action Taken:	The corresponding rEIS_LNK tuple is reinitialized and all the associated rEISCALL tuples are initialized and place on the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEIS_LNK tuple.
<i>Bad Data</i> =	EIS_LNK terminal process(eistp).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump</i> =	rEIS_LNK tuple.

Table 5 Error Code: EISVENDOR

Error Code:	EISVENDOR
Error Description:	The rEIS_LNK vendor is different from rEIS_DATA vendor.
Possible Error Effect:	Trying to communicate with the wrong EIS vendor.
Corrective Action Taken:	Update rEIS_LNK eis_vendor with rEIS_DATA eis_vendor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEIS_LNK tuple.
<i>Bad Data</i> =	rEIS_LNK EIS vendor (eis_vendor) number.
<i>Good Data</i> =	rEIS_DATA EIS vendor (eis_vendor) number.
<i>Logical Index</i> =	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump</i> =	rEIS_LNK tuple.

Table 6 Error Code: MSG_SET_ID

Error Code:	MSG_SET_ID
Error Description:	The rEIS_LNK message set identification is different from rEIS_DATA message set identification.

Possible Error Effect:	Receiving invalid messages.
Corrective Action Taken:	Update rEIS_LNK msg_set_id with rEIS_DATA msg_set_id
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rEIS_LNK tuple.
<i>Bad Data =</i>	rEIS_LNK message set id (msg_set_id).
<i>Good Data =</i>	rEIS_DATA message set id (msg_set_id).
<i>Logical Index =</i>	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump =</i>	rEIS_LNK tuple.

Table 7 Error Code: MX_CALL_LNK

Error Code:	MX_CALL_LNK
Error Description:	The maximum number of calls supported by an rEIS_LNK is different from rEIS_DATA maximum number of calls.
Possible Error Effect:	Allow the incorrect number of calls on the link.
Corrective Action Taken:	Update max_call_lnk with rEIS_DATA max_call_lnk
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rEIS_LNK tuple.
<i>Bad Data =</i>	EIS link maximum call link (mx_call_lnk) value.
<i>Good Data =</i>	rEIS_DATA maximum call link (mx_call_lnk) value.
<i>Logical Index =</i>	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump =</i>	rEIS_LNK tuple.

Table 8 Error Code: MX_LNK_CAP

Error Code:	MX_LNK_CAP
Error Description:	The maximum link capacity in messages per second is different from rEIS_DATA maximum link capacity.
Possible Error Effect:	Allow too many or not enough calls on the link at one time
Corrective Action Taken:	Update rEIS_LNK mx_lnk_cap with rEIS_DATA mx_lnk_cap
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rEIS_LNK tuple.
<i>Bad Data =</i>	rEIS_LNK maximum link capacity (mx_lnk_cap).
<i>Good Data =</i>	rEIS_DATA maximum link capacity (mx_lnk_cap).
<i>Logical Index =</i>	Logical key of rEIS_LNK (sm_rlnk) tuple.
<i>Dump =</i>	rEIS_LNK tuple.

Table 9 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The attributes in the tuples are not initialized to zero.
Possible Error Effect:	The tuple is initialized incorrectly.
Corrective Action Taken:	Zap the tuple (AUzaptup) and recover the key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rEIS_LNK tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Initialize the tuple to 0.
<i>Logical Index =</i>	Logical key of rEIS_LNK number.
<i>Dump =</i>	rEIS_LNK tuple.

EISSM

Software Release: 5E10 and later

1. EXTERNAL INFORMATION SYSTEM SWITCHING MODULE (EISSM) AUDIT

An rEISSM tuple contains data associated with the external information system (EIS) interface for listings services. The data contains the minimum number of call processing data links (CPDLs) needed to support the switching module (SM) during the busy hours and the total number of CPDLs assigned to the SM.

2. DATA STRUCTURES

The data structure for EISSM audit is the static relation containing the minimum number of data CPDL needed by an EIS on a SM relation (RLEIS_SM) during busy hours.

3. AUDIT CHECKS

The EISSM audit invokes the DRSTATIC driver to execute the syntactic check and invoke the semantic checks. For each tuple, the EISSM audit makes the following check after the driver has verified the keys. For details regarding each error code, see the comment block given.

LOSTTUP = Checking for LOST tuple in RLEISLKSEL relation.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A tuple is in the static relation RLEIS_SM and not in RLEISLKSEL.
Possible Error Effect:	A tuple is missing from relation RLEISLKSEL.
Corrective Action Taken:	The lost tuple will be inserted in RLEISLKSEL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rEIS_SM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Static rEIS_SM EIS ID (eis_id) tuple key.
<i>Logical Index</i> =	Logical key of rEIS_SM (eis_id) tuple.
<i>Dump</i> =	rEIS_SM tuple.

ENTRY7

Software Release: 5E10 and later

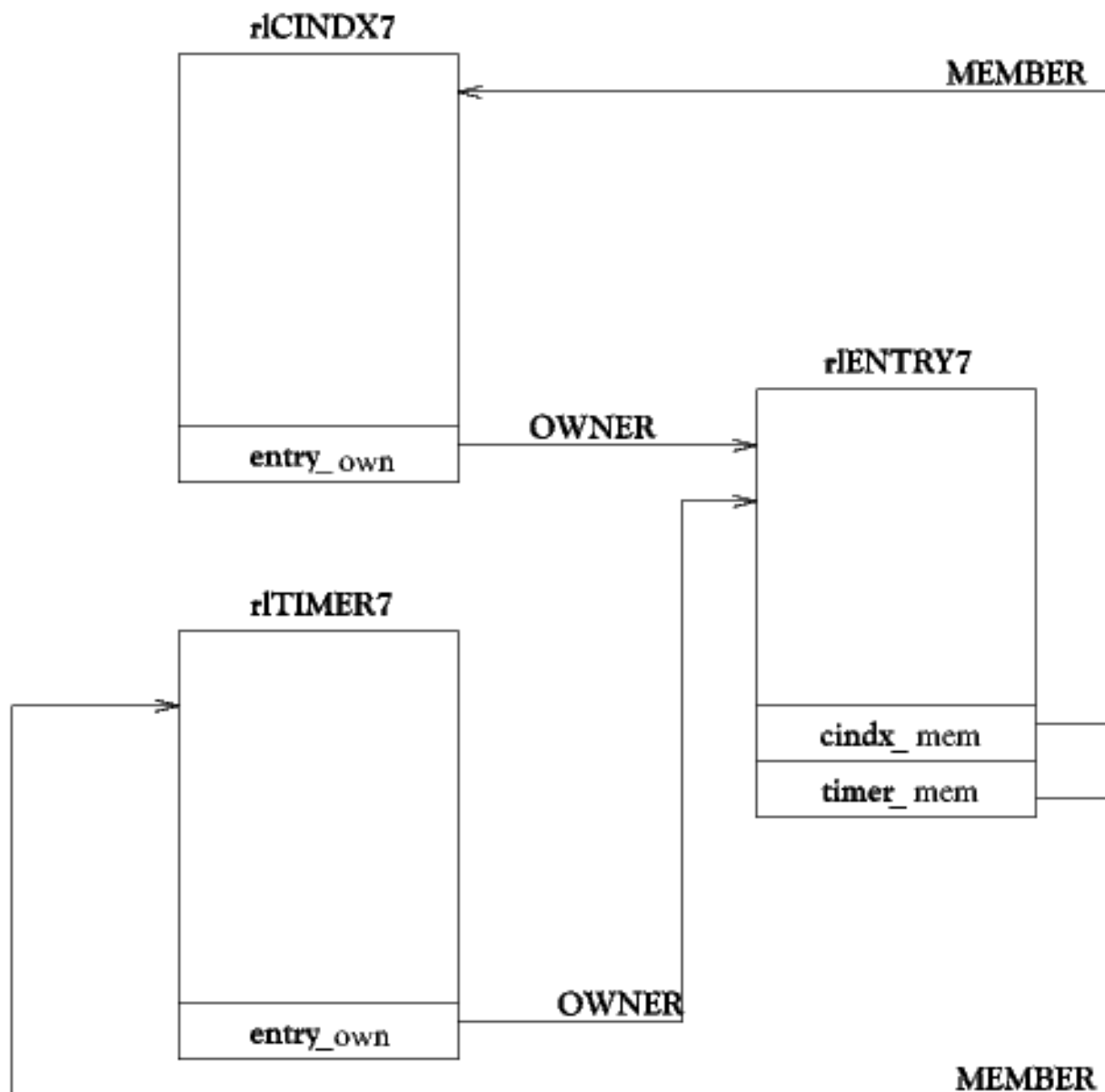
1. MCAH TABLE ENTRY (ENTRY7) AUDIT

The ENTRY7 audit is responsible for verifying the linkage data and other semantic data which is stored in the RENTRY7 relation. This audit runs in both the switching module (SM) and the administrative module (AM). Refer to Figure 1 for a pictorial view of the linkages related to the rENTRY7 tuple.

2. DATA STRUCTURES

The ENTRY7 relation is a dynamic relation which contains CCS7 message consolidation and acknowledgement handler (MCAH) table entries. Each entry in this relation represents one of the following MCAH entities:

- A consolidation entry for a CCS7 maintenance message about to be sent to the far-office.
- An acknowledgement pending entry representing a CCS7 maintenance message already sent to the far-office by MCAH.
- A "reset acknowledgement received" entry representing pending PSA work for reset message acknowledgements already received.



⇔ =One to One linkage.
 → =Owner-member linkage.
 — =Nonstandard linkage.

Figure 1 (ENTRY7-1) RLENTY7 Linkages

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the

ERROR HANDLING section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	The value of cic for this rENTRY7 tuple was found to be greater than the cic for the next linked ENTRY7.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Quarantine the related CINDX7 to be recovered by the CINDX7 audit and recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	The value of the cic field in rENTRY7 tuple.
<i>Good Data =</i>	The value of the cic field in the next rENTRY7 tuple.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump 1 =</i>	Contents of the rENTRY7 tuple.
<i>Dump 2 =</i>	Contents of the next linked rENTRY7 tuple.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	An rENTRY7 tuple was found to not be linked to either the RAR or IDLE list but the linkage to the CINDX7 was null. The CINDX linkage should not be null.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	The linkage field to the CINDX7.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump =</i>	Contents of the rENTRY7 tuple.

Table 3 Error Code: BADPROTOCOL

Error Code:	BADPROTOCOL
Error Description:	The value of protocol for this rENTRY7 tuple was not set to DBISUP7 and it should be.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	The value of the protocol field in rENTRY7 tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump =</i>	Contents of the rENTRY7 tuple.

Table 4 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	If this ENTRY7 is owned by the RAR list, then either the mtype field in this rENTRY7 tuple was not set to TMMC_RESET or the type field was not set to TMMC_I_RAR or TMMC_B_RAR. Otherwise, the error is reported because the mtype and/or type fields are not set to in-range values.

Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	The value of the type field in rENTRY7 tuple.
<i>Good Data =</i>	The value of the mtype field in rENTRY7 tuple.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump =</i>	Contents of the rENTRY7 tuple.

Table 5 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The value of cic_map for this rENTRY7 tuple was set to TMNULCICMASK when it should not be.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	TMNULCICMASK.
<i>Good Data =</i>	The value of the cic_map field in rENTRY7 tuple.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump =</i>	Contents of the rENTRY7 tuple.

Table 6 Error Code: HEAD

Error Code:	HEAD
Error Description:	The value of cic for this rENTRY7 tuple was found to be less than the cic for the previous linked ENTRY7.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Quarantine the related CINDX7 to be recovered by the CINDX7 audit and recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	The value of the cic field in rENTRY7 tuple.
<i>Good Data =</i>	The value of the cic field in the previous rENTRY7 tuple.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump 1 =</i>	Contents of the rENTRY7 tuple.
<i>Dump 2 =</i>	Contents of the previous linked rENTRY7 tuple.

Table 7 Error Code: INDEX

Error Code:	INDEX
Error Description:	The value of pci for this rENTRY7 tuple does not match the PCI of the owning CINDX7's related PCILST7.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the associated rENTRY7 tuple.
<i>Bad Data =</i>	The value of the pci field in rENTRY7 tuple.
<i>Good Data =</i>	The value of the pci field in the related rPCILST7 tuple.
<i>Logical Index =</i>	The key of the rENTRY7 tuple (entry).
<i>Dump 1 =</i>	Contents of the rENTRY7 tuple.
<i>Dump 2 =</i>	Contents of the owning rCINDX7 tuple.
<i>Dump 3 =</i>	Contents of the rPCILST7 tuple related to the owning rCINDX7 tuple.

Table 8 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	This rENTRY7 tuple is not linked to the correct owner tuple. For CNI platform rENTRY7 tuples (rPCI7_LAB.psu_mod == 0) the rENTRY7 tuple should be linked to an owner in the range of (TMT7CNI_MIN + DBSFTIMER7) <

	(rENTRY7.timer_mem.st_ownk) < (TMT7CNI_MAX + DBSFTIMER7). For PSU platform rENTRY7 tuples (rPCI7_LAB.psu_mod != 0), the owner should be in the range of (TMT7PSU_MIN + DBSFTIMER7) < (rENTRY7.timer_mem.st_ownk) < (TMT7PSU_MAX + DBSFTIMER7).
Possible Error Effect:	Possible asserts during CCS timer operations. Possible timer resource exhaustion.
Corrective Action Taken:	The rENTRY7 tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rENTRY7 tuple.
<i>Bad Data</i> =	The key of the rENTRY7 tuple.
<i>Good Data</i> =	The contents of rPCI7_LAB.psu_mod.
<i>Logical Index</i> =	The key of the rENTRY7 tuple.

Table 9 Error Code: MISSING

Error Code:	MISSING
Error Description:	For a given rENTRY7 tuple under audit, a corresponding rLAB7PRT or rPCI7_LAB static tuple is missing. For every rENTRY7 dynamic tuple there should exist a corresponding rLAB7PRT and rPCI7_LAB static tuple, where (rENTRY7.pci == rPCI7_lab.pci, for tuples where rENTRY7.type == TMMC_ACKPEND) and (rLAB7PRT.cic == rENTRY7.cic, for tuples where rENTRY7.singlet == DBNO). This error indicates that the rENTRY7 tuple is marked active, but there are no associated timer resources in use.
Possible Error Effect:	Possible asserts from CCS timer operations. Possible exhaustion of CCS timer resources.
Corrective Action Taken:	The rENTRY7 tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not an address. The relation ID of the missing static relation's tuple (RLLAB7PRT, or RLPCI7_LAB) is dumped.
<i>Bad Data</i> =	The cic field in the rENTRY7 tuple.
<i>Good Data</i> =	The pci field in the rENTRY7 tuple.
<i>Logical Index</i> =	The logical key, entry of the rENTRY7 tuple.
<i>Dump</i> =	The contents of the rENTRY7 tuple.

Table 10 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The two pass check for owner linkage has detected that the ENTRY7 is not on the RAR or IDLE list and the owner link is still set to DALNKNUL.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rENTRY7 tuple.
<i>Bad Data</i> =	The value of the quarstate field in rENTRY7 tuple.
<i>Good Data</i> =	The value of the owner link field in rENTRY7 tuple.
<i>Logical Index</i> =	The key of the rENTRY7 tuple (entry).
<i>Dump</i> =	Contents of the rENTRY7 tuple.

Table 11 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The value of range for this rENTRY7 tuple was found to not be in range (1-23) when the type is TMMC_ACKPEND and the singlet is set to DBNO.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rENTRY7 tuple.
<i>Bad Data</i> =	The value of the range field in rENTRY7 tuple.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	The key of the rENTRY7 tuple (entry).

<i>Dump</i> =	Contents of the rENTRY7 tuple.

Table 12 Error Code: TRACK_SM

Error Code:	TRACK_SM
Error Description:	In the SM, the value of sm for this rENTRY7 tuple does not agree with the value of Gl_pcrd.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Reset the value of sm to agree with Gl_pcrd.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rENTRY7 tuple.
<i>Bad Data</i> =	The value of the sm field in rENTRY7 tuple.
<i>Good Data</i> =	Gl_pcrd.
<i>Logical Index</i> =	The key of the rENTRY7 tuple (entry).
<i>Dump</i> =	Contents of the rENTRY7 tuple.

Table 13 Error Code: TYPE

Error Code:	TYPE
Error Description:	For an rENTRY7 tuple on the RAR list, either the mtype or type field are not set to a valid value. The valid value for mtype is TMMC_RESET and the valid values for type are TMMC_I_RAR or TMMC_B_RAR.
Possible Error Effect:	Incorrect MCAH table entry usage.
Corrective Action Taken:	Recover this ENTRY7 to the IDLE list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rENTRY7 tuple.
<i>Bad Data</i> =	The value of the type field in rENTRY7 tuple.
<i>Good Data</i> =	The value of the mtype field in rENTRY7 tuple.
<i>Logical Index</i> =	The key of the rENTRY7 tuple (entry).
<i>Dump</i> =	Contents of the rENTRY7 tuple.

Table 14 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle ENTRY7 has a non-zero value in one of its linkage fields.
Possible Error Effect:	Resources are not made available to users.
Corrective Action Taken:	Recover this rENTRY7 tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rENTRY7 tuple.
<i>Bad Data</i> =	Return value from zero linkage check.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rENTRY7 tuple (entry).
<i>Dump</i> =	Contents of the rENTRY7 tuple being audited.

EPKB

Software Release: 5E10 and later

1. EXTENDED PACKET BUFFER (EPKB) AUDIT

The EPKB audit performs some checks on the dynamic relation RLEPKB. The tuples of RLEPKB are temporary storage buffers for messages from the applications processor system process (APSP) destined for an applications processor (AP). The EPKB audit uses the indirect access sequential key driver (DRINASEK) to traverse through the RLEPKB relation tuple by tuple. The key and quarantine bit are checked for each tuple. The audit does not look at the other attributes in the tuple. Users of this other information do self auditing on the data.

2. DATA STRUCTURES

The rLEPKB relation is a dynamic relation of temporary storage buffers for messages from the APSP destined for an AP. The EPKB audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The EPKB audit insures that the key in each rLEPKB tuple is correct and a check of the quarantine bit is made. The indirect access sequential key driver (DRINASEK) is used to step through the rLEPKB relation tuple by tuple. The audit is entered from audit control.

4. ERROR HANDLING

If an error is detected, the DRINASEK driver zeroes out the tuple and puts the key back in.

EQUIP

Software Release: 5E10 and later

1. *SLC*[®] 96 EQUIPMENT (EQUIP) AUDIT

The *SLC*[®] 96 sends signals from subscriber lines to the digital carrier line unit (DCLU) within the switching modules. The subscriber lines terminate on shelves in the *SLC*[®] 96 remote terminal; each shelf connects to a *SLC*[®] 96 digital facility interface circuit (S-DFI) in the switching module over a T1 line. The equipment configuration relation (RLS96EQUIP) contains static (write-protected) data about the shelves, S-DFIs, and T1 lines. The equipment audit (EQUIP) verifies that the static *SLC*[®] 96 data in rls96EQUIP is internally consistent and also identifies missing tuples in the *SLC*[®] 96 shelf relation (RLS96SHELF), the *SLC*[®] 96 digital facilities interface relation (RLS96DFI), and the *SLC*[®] 96 T1 usage relation (RLS96T1USG). If tuples are missing in any of these dynamic relations, dynamic resources are lost.

This audit is part of the switch maintenance audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The EQUIP audit uses information in the static equipment configuration relation (RLS96EQUIP) to verify information in the *SLC*[®] 96 shelf relation (RLS96SHELF), the *SLC*[®] 96 digital facilities interface relation (RLS96DFI), and the *SLC*[®] 96 T1 usage relation (RLS96T1USG). Figure 1 shows the fields in RLS96EQUIP checked by the audit. The audit verifies that the digital facility interface (dfi) field and the bank mode (bank0mode or bank1mode) field are consistent.

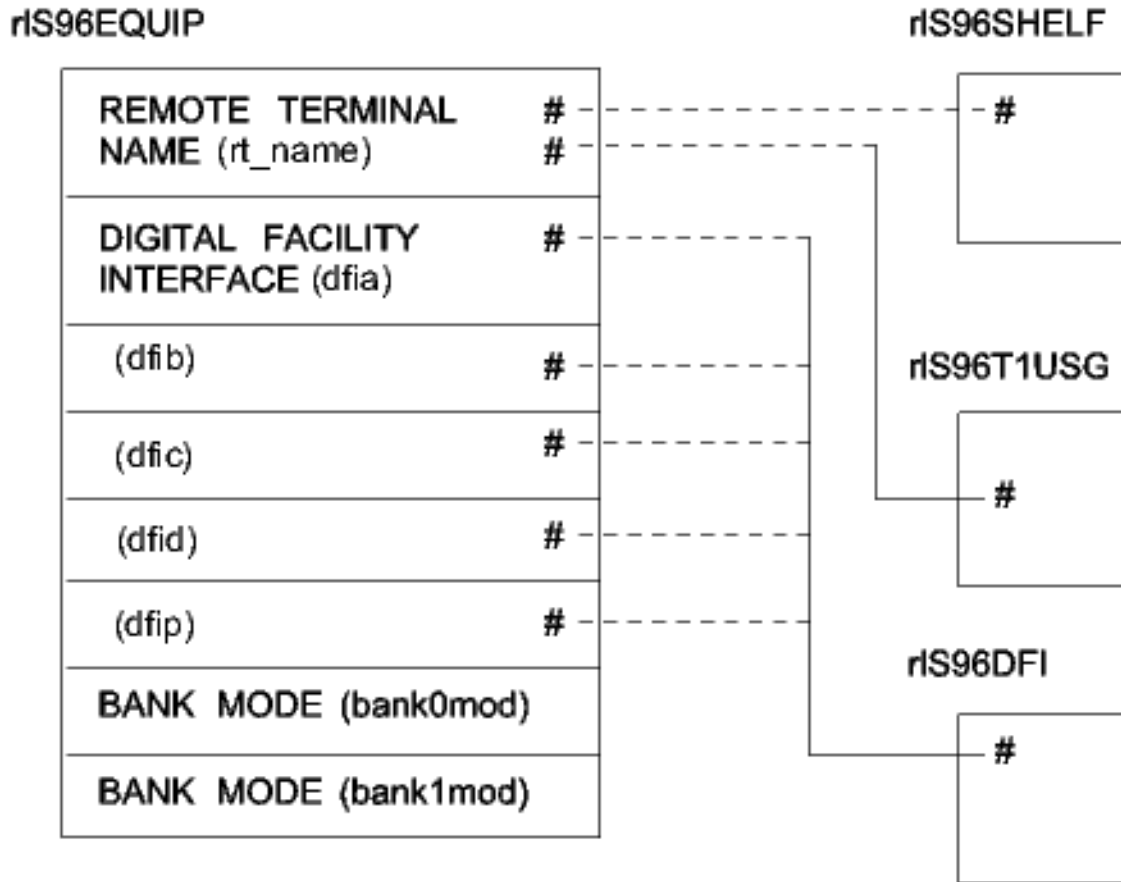


Figure 1 (EQUIP-1) Equipment Configuration Relation

3. AUDIT CHECKS

The EQUIP audit checks the modes of the RLS96EQUIP relation. If RLS96EQUIP is running in the update mode, the audit ends and runs during its next audit cycle. If RLS96EQUIP is not in the update mode, the audit invokes the static driver (DRSTATIC) to step through the tuples in RLS96EQUIP. The DRSTATIC driver calls a static to dynamic existence check function to perform the following checks:

- (a) For each tuple in rLS96EQUIP, the audit compares the information in the digital facility interface (dfia) and bank mode (bank0mode) fields.
 - (1) If no inconsistency exists, the audit verifies the existence of an associated tuple in either the RLS96SHELF, RLS96DFI, or RLS96T1USG relation. The audit uses the information in the dfi field to verify the existence of a tuple in the rLS96DFI relation. The audit uses the information in the remote terminal name (rt_name) field plus a shelf name determined by the dfi field to verify the existence of tuples in the RLS96SHELF and RLS96T1USG relations.
 - (2) If an inconsistency exists, an error is reported requiring manual action.
- (b) The audit performs the same comparison as stated in (a) between the dfib and bank0mode fields.

- (c) The audit performs the same comparison as stated in (a) between the dfic and bank1mode fields.
- (d) The audit performs the same comparison as stated in (a) between the dfid and bank1mode fields.

4. ERROR HANDLING

Any inconsistency in the RLS96EQUIP relation is reported as an error. This type of error requires manual action to correct the problem.

If a tuple that should exist is missing in either the RLS96SHELF, RLS96DFI, or RLS96T1USG relation, the audit inserts that tuple into the appropriate relation.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: DFIA

Error Code:	DFIA
Error Description:	When a rIS96EQUIP tuple exists, shelf A must be marked equipped and a valid DFI name must be in dfia field. This error was taken because either the bank0mode field is marked PCUNEQ or the difa field is PCNULLDFI.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Correct either the bank0mode field or the dfia field.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96EQUIP tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIS96EQUIP tuple in error.
<i>Dump =</i>	Dump of the rIS96EQUIP tuple.

Table 2 Error Code: DFIB

Error Code:	DFIB
Error Description:	The bank0mode of the rIS96EQUIP tuple indicates PCMODE2, concentrated mode, but the dfib field is not null.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Correct either the bank0mode field or the dfib field.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96EQUIP tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIS96EQUIP tuple in error.
<i>Dump =</i>	Dump of the rIS96EQUIP tuple.

Table 3 Error Code: DFIC

Error Code:	DFIC
Error Description:	The bank1mode of the rIS96EQUIP tuple indicates PCUNEQ, this bank unequipped, but the DFIC field is not null.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Correct either the bank1mode field or the dfic field.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96EQUIP tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIS96EQUIP tuple in error.

<i>Dump</i> =	Dump of the rIS96EQUIP tuple.

Table 4 Error Code: DFID

Error Code:	DFID
Error Description:	The bank1mode of the rIS96EQUIP tuple indicates PCMODE2, concentrated mode, but the dfid field is not null.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Correct either the bank1mode field or the dfid field.
Dump Description:	
<i>Error Address</i> =	Address of the rIS96EQUIP tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIS96EQUIP tuple in error.
<i>Dump</i> =	Dump of the rIS96EQUIP tuple.

Table 5 Error Code: EQUIP_DFID

Error Code:	EQUIP_DFID
Error Description:	The bank1mode of the rIS96EQUIP tuple indicates PCMODE2, concentrated mode, but the dfid field is not null.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Correct either the bank1mode field or the dfid field.
Dump Description:	
<i>Error Address</i> =	Address of the rIS96EQUIP tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIS96EQUIP tuple in error.
<i>Dump</i> =	Dump of the rIS96EQUIP tuple.

Table 6 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The rIS96EQUIP indicates that a rIS96DFI, rIS96SHELF, or rIS96T1USG tuple should exist but the tuple does not exist.
Possible Error Effect:	This could cause asserts because the static data exists but the dynamic data does not exist, causing dynamic data to be erroneously used.
Corrective Action Taken:	The rIS96DFI, rIS96SHELF, or the rIS96T1USG tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIS96EQUIP tuple.
<i>Bad Data</i> =	The relation identification of the tuple found missing.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the tuple found missing.
<i>Dump</i> =	Dump of the rIS96EQUIP tuple.

EXENET-A

Software Release: 5E12 only

1. EXTERNAL ETHERNET LINK (EXENET) AUDIT

Sockets are the interface to send and receive data using the ethernet network. They are used to represent per-connection information such as state and data buffers. The SOCKET can be linked to a PIPE or a MELNK based upon its socket system type or its protocol type. The audit verifies linkages among CMsocket[], CMesmepipe, CMesmelnk and CMmhatpl3q.

2. DATA STRUCTURES

The following data are used by the EXENET audit:

CMsocket[] = Socket protocol control block.

CMesmepipe = MEPIPE dynamic data.

CMesmelnk = MELNK dynamic data.

CMmhatpl3q = MH level 3 queue dynamic data.

3. AUDIT CHECKS

The EXENET audit makes the following checks:

BADLINK = Check of MHATPL3Q frame descriptor ownership.

BADRANGE_INDEX = Range check index into CMsocket[] for MELNK.

BADTYPE = Check value of CMsocket[].so_systype for MEPIPE.

BDCOUNT = Check of MHATPL3Q frame descriptor count.

NULL3LINK = Consistency check of MHATPL3Q frame descriptor ownership fields.

OUTOFRANGE = Range check index into CMsocket[] for MEPIPE.

TYPE = Check values in CMsocket[] for MELNK.

ZAP = Check for system interrupt handling.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between a MHATPL3Q and the ethernet frame descriptors (EFDs) it appears to own is invalid. Specifically, one or more of the following is true: - CMetfduse[CMmhatpl3q[0].first].l3_prev does not equal CMNIL_NDX - CMetfduse[CMmhatpl3q[0].first].bf_owner & CM_L3_OWNER does not equal 0

	- CMetfduse[CMmhatpl3q[0].first].usetype does not equal CMENXMT - CMetfduse[CMmhatpl3q[0].last].l3_next does not equal CMNIL_NDX - CMetfduse[CMmhatpl3q[0].last].bf_owner & CM_L3_OWNER does not equal 0 - CMetfduse[CMmhatpl3q[0].last].usetype does not equal CMENXMT
Possible Error Effect:	Frames associated with the MHATPL3Q are not transmitted properly.
Corrective Action Taken:	The audit sets CMmhatpl3q[index].first, .last, .ready and .blocked to CMNIL_NDX and the transmit count to 0.
Manual Action Required:	None.
Dump Description:	
Error Address =	Owner of EFDs (CMMHATPL3Q).
Bad Data =	NA.
Good Data =	NA.
Logical Index =	Index into CMmhatpl3q[].
Dump =	CMmhatpl3q[index].

Table 2 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	The audit finds that the index into the CMsocket[] in CMesmelnk[] (CMesmelnk[AUEXE_IDX][side].so_indx[0]) is not within range.
Possible Error Effect:	The MELNK is removed.
Corrective Action Taken:	The audit sets the index into CMsocket[] in CMesmelnk[] (CMesmelnk[AUEXE_IDX][i].so_indx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
Error Address =	Side of CMesmelnk[][].
Bad Data =	CMesmelnk[AUEXE_IDX][side].so_indx[0].
Good Data =	CM_NUMSOPCB.
Logical Index =	AUEXE_IDX.
Dump =	Contents of CMesmelnk[AUEXE_IDX][side].

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit finds that the systype in the CMsocket[] entry to which the PIPE points is not equal to CMSOST_HUB.
Possible Error Effect:	The MELNK is removed and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit sets the index into CMsocket[] in CMesmepipe[][] (CMesmepipe.ping_so_indx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
Error Address =	NA.
Bad Data =	CMsocket[CMesmepipe.ping_so_indx[side]].so_systype.
Good Data =	CMSOST_HUB.
Logical Index =	Side.
Dump 1 =	Contents of CMsocket[CMesmepipe.ping_so_indx[side]].
Dump 2 =	Contents of CMesmepipe.

Table 4 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	An MHATPL3Q does not own any EFDs (elements of CMetfduse[]), but has a non-zero transmit queue count (MHATPL3Q.xmit_q != 0).
Possible Error Effect:	MHATPL3Q frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the transmit queue count (MHATPL3Q.xmit_q = 0).
Manual Action Required:	None.
Dump Description:	
Error Address =	AUCMATPL3Q.
Bad Data =	The transmit queue count (MHATPL3Q.xmit_q).
Good Data =	NA.

<i>Logical Index</i> =	Index into CMmhatpl3q[].
<i>Dump</i> =	NA.

Table 5 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One or more of an MHATPL3Q's EFD (element of CMetfduse[]) ownership fields (<i>f_ifd</i> , <i>l_ifd</i> , <i>ready</i> , or <i>blocked</i>) contained an out-of-range value (!= CMNIL_NDX and > CM_NUMBUF).
Possible Error Effect:	Frames associated with the MHATPL3Q are not transmitted properly.
Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to CMNIL_NDX. Any lost frame descriptors are picked up by the EFD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUCMATPL3Q..
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Transmit queue count (MHATPL3Q.xmit_q).
<i>Logical Index</i> =	Index into CMmhatpl3q[].
<i>Dump</i> =	Contents of CMmhatpl3q[index].

Table 6 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit finds the index into CMsocket[] in CMesmepipe[] (CMesmepipe.ping_so_idx[0]) is not within range.
Possible Error Effect:	The MELNK is removed and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit sets the index into CMsocket[] in CMesmepipe[] (CMesmepipe.ping_so_idx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMesmepipe.ping_so_idx[side].
<i>Good Data</i> =	CM_NUMSOPCB.
<i>Logical Index</i> =	Side.
<i>Dump</i> =	Contents of CMesmepipe.

Table 7 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit finds one of the following: - CMsocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_protocol does not equal CMIPTCP or - CMsocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_systype does not equal CMSOST_RTCD.
Possible Error Effect:	The MELNK is removed.
Corrective Action Taken:	The audit sets the index into CMsocket[] in CMesmelnk[] (CMesmelnk[AUEXE_IDX][side].so_idx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUEXE_IDX.
<i>Bad Data</i> =	CMsocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_systype.
<i>Good Data</i> =	CMsocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_protocol.
<i>Logical Index</i> =	Side.
<i>Dump 1</i> =	Contents of CMsocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].
<i>Dump 2</i> =	Contents of CMesmelnk[AUEXE_IDX][side].

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit requests that the system integrity protocol (SIP) be reset to a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the PIPE or the MELNK.
<i>Logical Index =</i>	Either 0 for PIPE data structure or 1 for MELNK data structure.
<i>Dump =</i>	NA.

EXENET-B

Software Release: 5E13 and later

1. EXTERNAL ETHERNET LINK (EXENET) AUDIT

Sockets are the interface to send and receive data using the ethernet network. They are used to represent per-connection information such as state and data buffers. The SOCKET can be linked to a MEPIPE or a MELNK based upon its socket system type or its protocol type. The audit verifies linkages among CMsocket[], CMesmepipe, CMesmelnk and CMmhatpl3q.

2. DATA STRUCTURES

The following data are used by the EXENET audit:

CMsocket[] = Socket protocol control block.

CMesmepipe = MEPIPE dynamic data.

CMesmelnk = MELNK dynamic data.

CMmhatpl3q = MH level 3 queue dynamic data.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between either CMmhatpl3q or an entry in CMmhesl3q[melnk] and the ethernet frame descriptors (EFDs) it appears to own is invalid. Specifically, one or more of the following is true: • CMetfduse[CMmhatpl3q.first].l3_prev does not equal CMNIL_NDX • CMetfduse[CMmhatpl3q.first].bf_owner & CM_L3_OWNER does not equal 0 • CMetfduse[CMmhatpl3q.first].usetype does not equal CMENXMT • CMetfduse[CMmhatpl3q.last].l3_next does not equal CMNIL_NDX • CMetfduse[CMmhatpl3q.last].bf_owner & CM_L3_OWNER does not equal 0 • CMetfduse[CMmhatpl3q.last].usetype does not equal CMENXMT
Possible Error Effect:	Layer 3 may assert, lose frames or EFDs.
Corrective Action Taken:	The audit sets the EFD owner's <i>first</i> , <i>last</i> , <i>ready</i> and <i>blocked</i> fields to CMNIL_NDX and the <i>xmit_cnt</i> to 0.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	EFD owner type.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The EFD count value (<i>xmit_cnt</i>).
<i>Logical Index</i> =	NA or melnk number.
<i>Dump</i> =	The EFD owner.

Table 2 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	The audit finds that the index into the CMsocket[] in CMesmelnk[] (CMesmelnk[AUEXE_IDX][side].so_idx[0]) is not within range.
Possible Error Effect:	The MELNK is restarted.
Corrective Action Taken:	The audit sets the index into CMsocket[] in CMesmelnk[] (CMesmelnk[AUEXE_IDX][side].so_idx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Side of CMesmelnk[][].
<i>Bad Data</i> =	CMesmelnk[AUEXE_IDX][side].so_idx[0].
<i>Good Data</i> =	CM_NUMSOPCB.
<i>Logical Index</i> =	MELNK number.
<i>Dump</i> =	Contents of CMesmelnk[AUEXE_IDX][side].

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit finds that the systype in the CMsocket[] entry to which the PIPE points is not equal to CMSOST_HUB.
Possible Error Effect:	The MEPIPE is restarted and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit sets the index into CMsocket[] in CMesmepipe[][] (CMesmepipe.ping_so_idx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMsocket[CMesmepipe.ping_so_idx[side]].so_systype.
<i>Good Data</i> =	CMSOST_HUB.
<i>Logical Index</i> =	Side.
<i>Dump 1</i> =	Contents of CMsocket[CMesmepipe.ping_so_idx[side]].
<i>Dump 2</i> =	Contents of CMesmepipe.

Table 4 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	Either CMmhatpl3q or CMmhesl3q[melnk] do not own any EFDs (elements of CMetfduse[]), but have a non-zero EFD count (<i>xmit_cnt</i> != 0). The owner can be determined based on the value in the report's <i>Error Address</i> field.
Possible Error Effect:	Layer 3 EFD frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the EFD count (<i>xmit_cnt</i> = 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	EFD owner type.
<i>Bad Data</i> =	The EFD count (<i>xmit_cnt</i>).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA or the melnk value.
<i>Dump</i> =	NA.

Table 5 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	An out of range value (> CMETNUMBUF and != CMNIL_NDX) was found in at least one EFD linkage field (<i>first</i> , <i>last</i> , <i>ready</i> , <i>blocked</i>) in either CMmhatpl3q or an entry in CMmhesl3q[melnk].
Possible Error Effect:	The out of range values may cause asserts, lost frames or lost EFDs.

Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to CMNIL_NDX. Any lost frame descriptors are picked up by the EFD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The EFD owner type.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The EFD count field (<i>xmit_cnt</i>).
<i>Logical Index =</i>	NA or melnk number.
<i>Dump =</i>	The EFD owner structure.

Table 6 Error Code: OUTFRANGE

Error Code:	OUTFRANGE
Error Description:	The audit finds the index into CMssocket[] in CMesmepipe[] (CMesmepipe.ping_so_idx[side]) is not within range.
Possible Error Effect:	The MEPIPE is restarted and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit sets the index into CMssocket[] in CMesmepipe[][] (CMesmepipe.ping_so_idx[side]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMesmepipe.ping_so_idx[side].
<i>Good Data =</i>	CM_NUMSOPCB.
<i>Logical Index =</i>	Side.
<i>Dump =</i>	Contents of CMesmepipe.

Table 7 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit finds one of the following: - CMssocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_protocol does not equal CMIPTCP or - CMssocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_systype does not equal CMSOST_RTCD.
Possible Error Effect:	The MELNK is restarted.
Corrective Action Taken:	The audit sets the index into CMssocket[] in CMesmelnk[] (CMesmelnk[AUEXE_IDX][side].so_idx[0]) to its null value (CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	AUEXE_IDX.
<i>Bad Data =</i>	CMssocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_systype.
<i>Good Data =</i>	CMssocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].so_protocol.
<i>Logical Index =</i>	Side.
<i>Dump 1 =</i>	Contents of CMssocket[CMesmelnk[AUEXE_IDX][side].so_idx[0]].
<i>Dump 2 =</i>	Contents of CMesmelnk[AUEXE_IDX][side].

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Index used in auditing data.
<i>Logical Index =</i>	An identifier indicating the data being audited at the time of the error.
<i>Dump =</i>	NA.

EXRR

Software Release: 5E10 and later

1. EXTENDED REROUTE (EXRR) AUDIT

Trunk group controls are a set of regulators used to protect an office from excessive routing requests and to expand routing patterns during busy periods. These regulators control the number of calls that can access a trunk group. A hierarchy exists within the trunk group controls. The group trunk group controls relation (RLGRP_TGC) maps active trunk groups in an office to trunk groups that have controls associated with them. The trunk group controls relation (RLTGC) maps the trunk groups and their associated controls to a specific control. The extended reroute relation (RLEXRR) maps a specific reroute control to a series of trunk groups designated to be the trunks on which the call attempts termination. The extended reroute audit (EXRR) verifies the linkages between the extended reroute relation (RLEXRR) and the trunk group controls relation (RLTGC). The audit also checks for lost resources by verifying linkages to the data base head cells relation (RLDB_HEADC).

This audit is part of the Routing and Terminal Allocation audits subgroup. It operates in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The data structures checked by the EXRR audit are the extended reroute relation (RLEXRR), the trunk group controls relation (RLTGC), and the database headcells relation (RLDB_HEADC). Figure 1 illustrates the layout of RLEXRR and shows the items checked by the audit. Included in the information within the tuples is the control identification, which is the point-back linkage to the RLTGC relation.

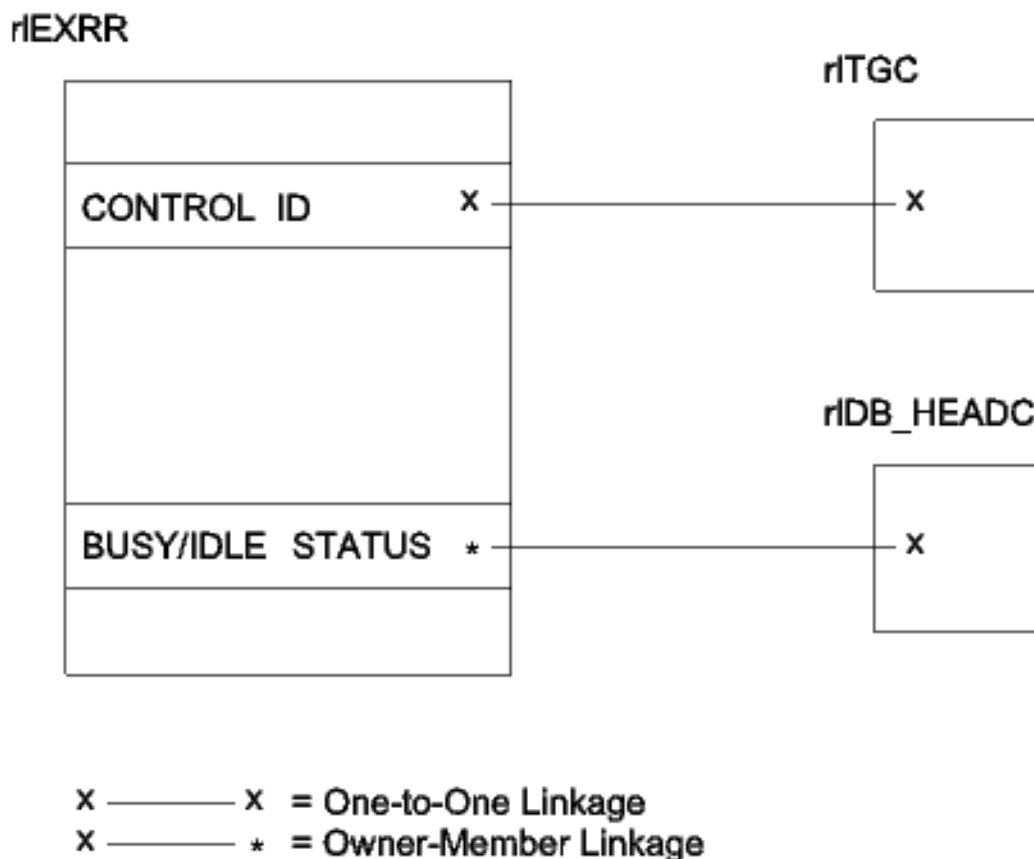


Figure 1 (EXRR-1) Extended Reroute Relation

3. AUDIT CHECKS

To verify the linkages in the extended reroute relation, the EXRR audit calls the linkage driver (DRLINK). The DRLINK driver ensures that each tuple in RLEXRR has a valid one-to-one linkage to the associated rITGC relation. The DRLINK driver also verifies the owner-member linkages to the RLDB_HEADC relation. The DRLINK driver invokes the semantic check function after verifying the linkages of each rLEXRR tuple. The semantic check function performs the following checks:

- (a) The audit reads the control ID field and verifies that the rITGC associated with the current rLEXRR tuple has attributes with the proper hunt and control types.
- (b) The audit verifies that all tuples in rLEXRR are either busy or idle.

4. ERROR HANDLING

The audit reports errors when any linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples and links them to the appropriate idle list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A tuple with valid linkages and data to qualify as busy is not on the busy list.
Possible Error Effect:	This tuple is taking up space in the relation that could be used for active tuples.
Corrective Action Taken:	The tuple is reinitialized and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of current tuple.
<i>Bad Data</i> =	Contents of owner field.
<i>Good Data</i> =	Idle list headcell.
<i>Logical Index</i> =	rrlink field of rLEXRR tuple.
<i>Dump</i> =	Contents of rLEXRR tuple.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A tuple has null pointers and data to indicate that it should be on the idle list. However, it is not on the idle list.
Possible Error Effect:	There is an idle tuple which can never be made active since it is not on the idle list.
Corrective Action Taken:	The tuple is reinitialized and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of current tuple.
<i>Bad Data</i> =	Contents of owner field.
<i>Good Data</i> =	Idle list headcell.
<i>Logical Index</i> =	rrlink field of rLEXRR tuple.
<i>Dump</i> =	Contents of rLEXRR tuple.

Table 3 Error Code: RRHTYPE

Error Code:	RRHTYPE
Error Description:	The rLEXRR tuple does not have a spray or ordered hunt type indicated in rlvTGC.
Possible Error Effect:	This rLEXRR tuple should never be accessed by network management but it is

	taking up space in the relation.
Corrective Action Taken:	The tuple is reinitialized and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of current rEXRR tuple.
<i>Bad Data =</i>	Hunt type indicated in the rTGC tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rlink field of rEXRR tuple.
<i>Dump 1 =</i>	Contents of rEXRR tuple.
<i>Dump 2 =</i>	Contents of associated rTGC tuple.

FACPM

Software Release: 5E10 and later

1. FACILITY PERFORMANCE MONITORING STATE (FACPM) AUDIT

Performance monitoring can be inhibited or allowed on a DS1 facility. The current status of both the 15 minute and daily reports is contained in the RLFACTHRS relation. This relation contains both the dynamic and static inhibit/allow status for both the 15 minute and daily reports. The dynamic performance monitoring status can be changed by using the inhibit/allow commands and this is not audited. The performance monitoring static status can be set by using recent change and is maintained in a protected relation (either RLFACTHRS or RLDF2THRS). The FACPM audit verifies that the static inhibit/allow states are consistent in the rIFACPM tuple with the protected data information in the threshold relation.

2. DATA STRUCTURES

The protected RLPMFACGRP relation associates a DS1 or DFI-2 facility with a performance monitoring group. Each facility has one rIPMFACGRP tuple and one rIFACPM tuple. The protected RLFACTHRS relation contains the performance monitoring thresholds and the static inhibit/allow status for each DS1 performance monitoring group. The RLDF2THRS relation contains the performance monitoring thresholds and the static inhibit/allow status for each DFI-2 performance monitoring group. The IDCU performance monitoring data structures are shown in Figure 1. The DFI-2 performance monitoring data structures are shown in Figure 2.

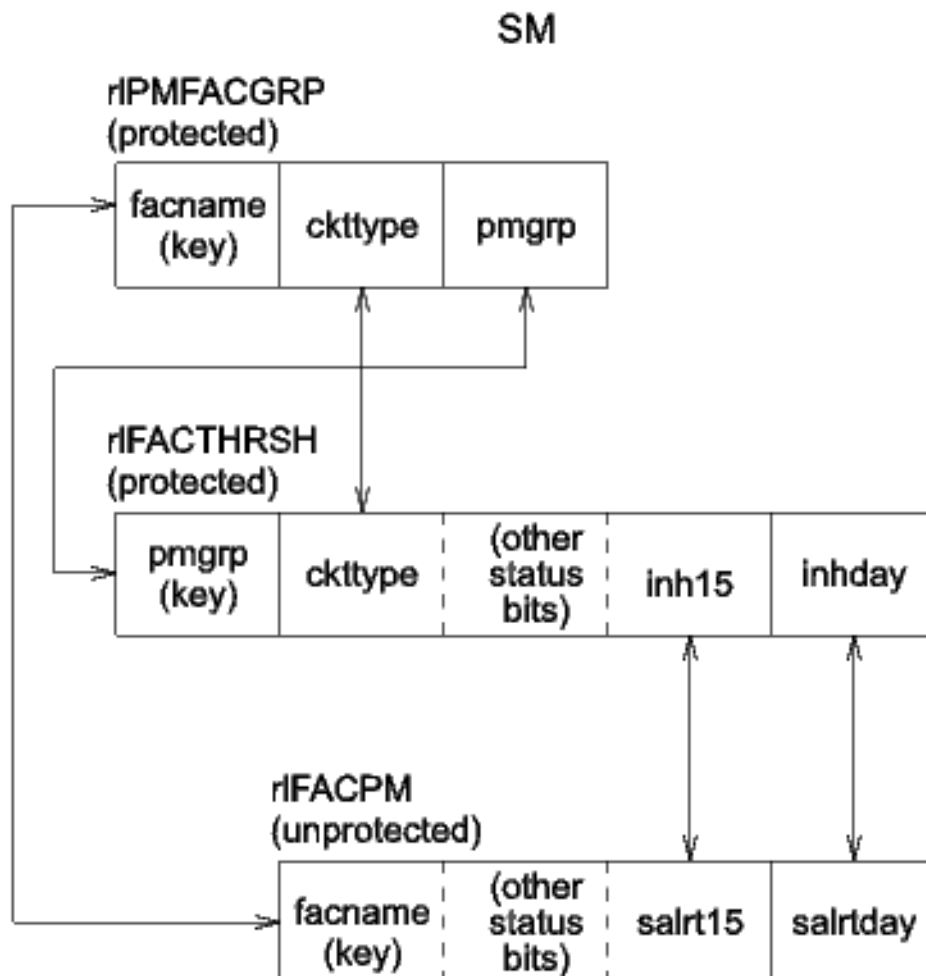


Figure 1 (FACPM-1) IDCU Performance Monitoring Data Structures

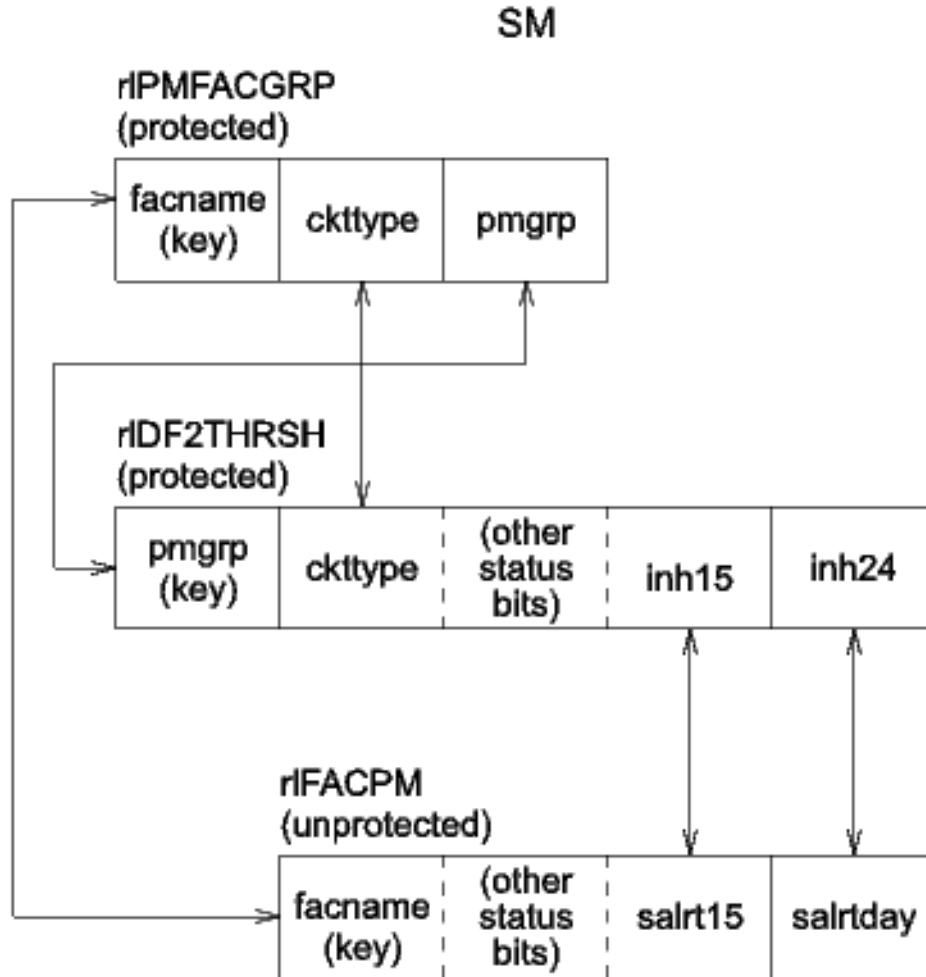


Figure 2 (FACPM-2) DFI-2 Performance Monitoring Data Structures

3. AUDIT CHECKS

The FACPM audit uses the DRSTATIC audit driver to step through the RLFACPM relation. The FACPM audit uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given. For each tuple, the FACPM audit makes the following checks:

- BADTYPE = Check the relation ID type.
- EXTRATUP = Check for extra tuples.
- KEY = Check the tuple's key.
- ODDLOSTUP = Check threshold tuples.
- STATUS = Check the static inhibit/allow bits.
- TYPE = Check the circuit type.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The relation ID based on the circuit type and determined in the audit for the threshold data does not equal either RLFACTHRSH or RLDF2THRSH.
Possible Error Effect:	User process stack overwritten.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIFACPM tuple.
<i>Bad Data =</i>	Relid of threshold relation.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	DS1 or DFI-2 Facility name (rIFACPM.facname).
<i>Dump 1 =</i>	The contents of rIFACPM tuple.
<i>Dump 2 =</i>	The contents of rIPMFACGRP tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A dynamic tuple in RLFACTPM relation with the facility name as key does not have a corresponding tuple in RLPMFACGRP with the same key value.
Possible Error Effect:	Extra rIFACPM tuple contains performance monitoring status for unknown facility.
Corrective Action Taken:	The extra rIFACPM tuple is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIFACPM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Facility name (rIFACPM.facname).
<i>Dump =</i>	The contents of rIFACPM tuple.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	The rIFACPM tuple has a corrupt key value. The FACPM audit has reconstructed the key value which does not match the current key value in rIFACPM.
Possible Error Effect:	The database has been corrupted. Users cannot update the performance monitoring status for this facility.
Corrective Action Taken:	The key field is updated with the reconstructed facility name.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIFACPM tuple.
<i>Bad Data =</i>	Invalid key for facility name (rIFACPM.facname).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Facility name (rIPMFACGRP.facpmkey).
<i>Dump =</i>	The contents of rIFACPM tuple.

Table 4 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The protected relation that holds the threshold information for performance monitoring facilities is missing.
Possible Error Effect:	Threshold information is missing and could cause performance monitoring errors.
Corrective Action Taken:	None. The threshold information has to be inserted manually.

Manual Action Required:	Using the circuit type as reported in Bad Data below and relation ID as reported in Good Data below for the facility, insert the desired threshold information into either the RLFACTHRSH(1734) or RLDF2THRSH(1851).
Dump Description:	
<i>Error Address</i> =	Address of rIPMFACGRP tuple.
<i>Bad Data</i> =	Circuit type (rIPMFACGRP.ckttype).
<i>Good Data</i> =	Relation ID of missing threshold tuple.
<i>Logical Index</i> =	Performance monitoring group (rIPMFACGRP.pmgrp).
<i>Dump</i> =	The contents of rIPMFACGRP tuple with the key (rIPMFACGRP.facname).

Table 5 Error Code: STATUS

Error Code:	STATUS
Error Description:	The status for the static inhibit/allow state for the 15 minute or daily report in the unprotected RLFACTHRSH relation does not match the data in the protected threshold relation (RLFACTHRSH or RLDF2THRSH).
Possible Error Effect:	Reports that should be inhibited could be printed or reports that should be printed could be suppressed.
Corrective Action Taken:	Status in the unprotected rIFACPM tuple is reset to match the status in the threshold tuple for this facility.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The 15 minute report status in rIFACPM tuple (rIFACPM.salrt15).
<i>Bad Data</i> =	The daily report status in rIFACPM tuple (rIFACPM.salrtday).
<i>Good Data</i> =	The 15 minute report status in either rIFACTHRSH or rIDF2THRSH tuple (rIFACTHRSH.inh15 or rIDF2THRSH.inh15).
<i>Logical Index</i> =	The daily report status in either rIFACTHRSH or rIDF2THRSH tuple (rIFACTHRSH.inhday or rIDF2THRSH.inh24).
<i>Dump 1</i> =	The contents of rIFACPM tuple.
<i>Dump 2</i> =	The contents of rIPMFACGRP tuple.

Table 6 Error Code: TYPE

Error Code:	TYPE
Error Description:	Invalid circuit type was found in the RLPMFACGRP relation.
Possible Error Effect:	Performance monitoring is attempted on circuit which is not appropriate for this type of monitoring.
Corrective Action Taken:	The audit error is reported to inform the user of this invalid circuit so that the invalid rIPMFACGRP tuple can be removed.
Manual Action Required:	Using the circuit type reported in Bad Data, the performance monitoring group reported in Good Data, and facility name reported in Logical Index delete the rIPMFACGRP tuple.
Dump Description:	
<i>Error Address</i> =	Address of rIPMFACGRP tuple.
<i>Bad Data</i> =	The circuit type (rIPMFACGRP.ckttype).
<i>Good Data</i> =	Performance monitoring group (rIPMFACGRP.pmgrp).
<i>Logical Index</i> =	Facility name (rIPMFACGRP.facname).
<i>Dump</i> =	The contents of rIPMFACGRP tuple.

FACSTAT-A

Software Release: 5E10 - 5E11

1. FACILITY STATUS FOR FAST PORT TERMINATION (FACSTAT) AUDIT

The facility status for fast port termination is used to keep the inhibit/allow traffic status for all facilities connected to a given time slot interchanger (TSI) link. This information is used by call processing to determine quickly if a port is available rather than just assigning a port and having it fail later in the process.

The FACSTAT audit is responsible for ensuring the integrity of tuples in the RLFACSTAT relation. For the digital networking unit-synchronous optical network (SONET) (DNU_S), this relation contains bitmaps that indicates whether the virtual peripheral interface data bus (VPIDB) associated with a digital signal 1 (DS1) facility has a carrier group alarm (CGA) or is inhibited from receiving termination attempts. Bits in the bitmaps are audited against the rISMEST tuples associated with facilities represented by this bit position. For DNU_S, each bit position indicates the VPIDB number of the facilities used in port termination.

2. DATA STRUCTURES

In the case of DNU_S, the RLFACSTAT relation is a dynamic relation that contains information used to describe the inhibit/allow traffic status of VPIDB facilities used in basic call processing. The RLFACSTAT relation is an indexed compaction access relation that is keyed by the *fakey* field. The *fakey* field is composed of the switching module (SM) number and the TSI slice and port number.

For DNU_S facilities, the *inhibit* attribute is a bitmap indicating whether the VPIDB associated with a digital signal level 1 (DS1) facility is to be inhibited from receiving termination attempts. The least significant bit (rightmost bit) is associated with VPIDB 0 and the most significant bit (leftmost bit) is associated with VPIDB 31.

The hierarchy DNU_S facilities is as follows: the signaling terminal equipment (STE) has an associated subscriber test station 1 (STS1). Each STS1 has 28 virtual tributaries 1.5 (VT1.5) and 28 DS1s associated with it. Each DS1 has a total of 24 digital signal level 0 (DS0) circuits with it. CGAs can occur on any of the circuits except the DS0 level. If a parent circuit is out of service, then its children circuits should be inhibited from receiving termination attempts. If any parent circuits have GCAs, then its children circuits should also have a CGA.

For DNU_S facilities, the *cga* attribute is a bitmap indicating whether the VPIDB associated with a DS1 facility was inhibited from receiving termination attempts because of a CGA on that facility. The least significant bit (rightmost bit) is associated with VPIDB 0 and the most significant bit (leftmost bit) is associated with VPIDB 31.

The *quarstat* field is the audits quarantine state that is currently not being used.

The rFACSTAT tuples have no linkage associated with them. These tuples reside on both the communication module processor (CMP) and each switching module (SM). In the CMP, there are 40 tuples per SM in continuous storage up to a maximum of 920 tuples. Each SM has 40 RLFACSTAT tuples in continuous storage. The SM versions are considered the masters. Every time the audit runs in the SM, it will request that a copy of the 40 tuples be sent to the CMP so that its tuples are refreshed every audit cycle. This refresh mechanism prevents the need for an interprocessor audit.

3. AUDIT CHECKS

The FACSTAT audit will perform different checks in both the CMP and the SM environments. In the CMP, the FACSTAT audit uses the DRSTATIC driver to step through the RLMODATT static relation and make sure the corresponding rFACSTAT tuples exist. The SM version of the FACSTAT audit uses the DRINDEXED driver to step through the RLFACSTAT relation and perform the semantic checks on the rFACSTAT tuples. The CMP and SM checks are listed separately below.

If the CMP FACSTAT audit is entered, the following checks are performed for each tuple in the RLMODATT

relation:

LOSTTUP = Check to make sure that each rIMODATT tuple with a *unittype* of SMSM2000 has 40 rIFACSTAT tuples associated with it in the CMP.

If the SM FACSTAT audit is entered, the following checks are performed for each tuple in the RLFACSTAT relation.

ALARM = Check to make sure that the CGA bits corresponding to unequipped DNU_S or DLTU2 facilities are 0.

ALWOOS = Check to make sure that a DS1 circuit is not marked inhibited in RLFACSTAT when none of its parent circuits (STE, STS, or VT1.5) are out of service.

INHHDW = Check to make sure that a DS1 circuit that has a parent with a CGA is also marked as having a CGA in rIFACSTAT.

INHSTATE = Check to make sure that a DS1 circuit is marked inhibited in RLFACSTAT when one of its parent circuits is out of service.

LOSTTUP = Check to make sure there are 40 rIFACSTAT tuples in an SM-2000.

ODDLOSTUP = Check to make sure the rUNIT_CKT tuples are available for the STE, STS1, VT1.5, and DS1 circuits. Also make sure the rIPLI_LNK tuple is available for these circuits. The error type is determined by the key value as to which tuple was missing.

SALW = Check to make sure that a DS1 circuit is not marked with a CGA in the RLFACSTAT relation when the circuit itself or none of its parents have a CGA associated with them.

SMEST = Check to make sure there are rSMEST tuples available for the STE, STS1, VT1.5, and DS1 circuits. The key field in the dump will determine which tuple is missing.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The FACSTAT audit running in the CMP has detected that an rIFACSTAT tuple corresponding to an rIMODATT tuple is missing. Every rIMODATT tuple with <i>unittype</i> of SMSM2000 should have 40 rIFACSTAT tuples corresponding to it.
Possible Error Effect:	Facility status for fast port termination may be impaired affecting call processing.
Corrective Action Taken:	The missing rIFACSTAT tuple is inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rIMODATT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The <i>module</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The logical key of the rIFACSTAT tuple inserted (<i>fakey</i>).
<i>Dump</i> =	The rIMODATT tuple.

Table 2 Error Code: ALARM

Error Code:	ALARM

Error Description:	The audit detected that a <i>cga</i> bit associated with an unequipped DNU_S or DLTU2 facility is not 0.
Possible Error Effect:	The alarm levels in DNU_S or DLTU2 facilities may not reflect the correct levels.
Corrective Action Taken:	The corresponding <i>cga</i> bit in the rFACSTAT tuple is turned off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rFACSTAT tuple.
<i>Bad Data</i> =	This is 0 if a DNU_S facility or 1 if a DLTU2 facility.
<i>Good Data</i> =	The corresponding <i>pct_link</i> field from the corresponding rPLI_LNK tuple if this is a DNU_S facility. Otherwise it is the <i>pidb_num</i> field from the corresponding rPIDB for DLTU2 facilities.
<i>Logical Index</i> =	The bit mask showing the position of the corrupted <i>cga</i> bit.
<i>Dump 1</i> =	The rFACSTAT tuple.
<i>Dump 2</i> =	The associated rVPIDB_F tuple for DNU_S facilities or the associated rPIDB tuple for DLTU2 facilities.

Table 3 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The audit detected that a DS1 facility is marked inhibited in an rFACSTAT tuple (that is, inhibit bit is set to 1) when the DS1 and all its parents are not out of service.
Possible Error Effect:	Calls may be routed away from this DS1 facility when it is operational and in service.
Corrective Action Taken:	The inhibit bit corresponding to this DS1 facility is turned off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rFACSTAT tuple.
<i>Bad Data</i> =	The value of the inhibit bit associated with this DS1 before correction.
<i>Good Data</i> =	The bit mask showing the relative position of the bit in rFACSTAT.
<i>Logical Index</i> =	The logical key of the rFACSTAT tuple.
<i>Dump 1</i> =	The rFACSTAT tuple.
<i>Dump 2</i> =	The associated rVPIDB_F tuple.
<i>Dump 3</i> =	The associated DS1 rSMEST tuple.
<i>Dump 4</i> =	The associated VT1.5 rSMEST tuple.
<i>Dump 5</i> =	The associated STS1 rSMEST tuple.
<i>Dump 6</i> =	The associated STE rSMEST tuple.

Table 4 Error Code: INHHDW

Error Code:	INHHDW
Error Description:	The audit has detected a DS1 facility that either has a CGA associated with it or one of its parents has a CGA and the corresponding CGA bit in the rFACSTAT tuple is not set.
Possible Error Effect:	Calls may be routed to this DS1 facility when it has a CGA and unable to terminate calls.
Corrective Action Taken:	The CGA bit corresponding to this DS1 facility is turned on.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rFACSTAT tuple.
<i>Bad Data</i> =	The value of the CGA bit associated with this DS1 before correction.
<i>Good Data</i> =	The bit mask showing the relative position of the bit in rFACSTAT.
<i>Logical Index</i> =	The logical key of the rFACSTAT tuple.
<i>Dump 1</i> =	The rFACSTAT tuple.
<i>Dump 2</i> =	The associated rVPIDB_F tuple.
<i>Dump 3</i> =	The associated DS1 rSMEST tuple.
<i>Dump 4</i> =	The associated VT1.5 rSMEST tuple.
<i>Dump 5</i> =	The associated STS1 rSMEST tuple.
<i>Dump 6</i> =	The associated STE rSMEST tuple.

Table 5 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	The audit has detected a DS1 facility that is not marked inhibited in the rFACSTAT tuple when either it or one of its parents are out of service.

	Therefore, it should be marked inhibited.
Possible Error Effect:	Calls may be routed to this DS1 facility when it is out of service and unable to terminate calls.
Corrective Action Taken:	The inhibit bit corresponding to this DS1 facility is turned on.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rFACSTAT tuple.
<i>Bad Data</i> =	The value of the inhibit bit associated with this DS1 before correction.
<i>Good Data</i> =	The bit mask showing the relative position of the bit in rFACSTAT.
<i>Logical Index</i> =	The logical key of the rFACSTAT tuple.
<i>Dump 1</i> =	The rFACSTAT tuple.
<i>Dump 2</i> =	The associated rVPIDB_F tuple.
<i>Dump 3</i> =	The associated DS1 rSMEST tuple.
<i>Dump 4</i> =	The associated VT1.5 rSMEST tuple.
<i>Dump 5</i> =	The associated STS1 rSMEST tuple.
<i>Dump 6</i> =	The associated STE rSMEST tuple.

Table 6 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The SM portion of the FACSTAT audit detected a missing rFACSTAT tuple. Every SM must have at least 40 rFACSTAT tuples.
Possible Error Effect:	Facility status for fast port termination for this SM may be degraded or not operational. This may impact call terminations.
Corrective Action Taken:	The missing rFACSTAT tuple is inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the buffer containing the rFACSTAT tuple inserted.
<i>Bad Data</i> =	The value of the <i>tsi link</i> associated with this rFACSTAT tuple.
<i>Good Data</i> =	The SM number of the SM missing the rFACSTAT tuple.
<i>Logical Index</i> =	The logical key of the rFACSTAT tuple.
<i>Dump</i> =	NA.

Table 7 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The SM portion of the FACSTAT audit was unable to obtain a static relation tuple needed to audit the dynamic rFACSTAT tuples. The KEY field in the audit dump will indicate which static tuple the audit was unable to obtain. The KEY field values and the associated static tuples missing are listed below: 0 = The audit was unable to read rUNIT_CKT for a DS1 facility. 1 = The audit was unable to read rUNIT_CKT for a VT1.5 facility. 2 = The audit was unable to read rUNIT_CKT for an STS-1 facility. 3 = The audit was unable to read rUNIT_CKT for an STE facility. 4 = The audit was unable to read rPLI_LNK for this TSI link. The good data in the dump will vary according to the KEY field as follows: 0 Good data is the <i>tsi link</i> for these DNU_S facilities. 1 = Good data is the circuit index of the DS1 facility. 2 = Good data is the circuit index of the VT1.5 facility. 3 = Good data is the circuit index of the STS-1 facility.

	4 = Good data is the circuit index of the STE facility. The dump in the audit error will also vary according to the KEY value as follows: 0 = The rIFACSTAT and rVPIDB_F tuples are dumped. 1 = The rIFACSTAT and rVPIDB_F tuples are dumped. 2 = The rIFACSTAT and rVPIDB_F tuples are dumped. 3 = The rIFACSTAT and rVPIDB_F tuples are dumped. 4 = Only the rIFACSTAT tuple is dumped.
Possible Error Effect:	The contents of the rIFACSTAT tuple can not be audited and is therefore subject to corruption.
Corrective Action Taken:	None.
Manual Action Required:	The missing tuple(s) must be inserted by recent change.
Dump Description:	
<i>Error Address</i> =	The address of the rIFACSTAT tuple under audit.
<i>Bad Data</i> =	The return code from DBfrdtup() used to read the static relation.
<i>Good Data</i> =	This will vary based on the Logical Index field as indicated above.
<i>Logical Index</i> =	This is the value of KEY as indicated above.
<i>Dump</i> =	This will vary based on the value of KEY.

Table 8 Error Code: SALW

Error Code:	SALW
Error Description:	The audit detected that a DS1 facility is marked as having a CGA when the DS1 and all its parents have no CGAs
Possible Error Effect:	Calls may be routed away from this DS1 facility when it is operational and in service.
Corrective Action Taken:	The CGA bit corresponding to this DS1 facility is turned off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIFACSTAT tuple.
<i>Bad Data</i> =	The value of the CGA bit associated with this DS1 before correction.
<i>Good Data</i> =	The bit mask showing the relative position of the bit in rIFACSTAT.
<i>Logical Index</i> =	The logical key of the rIFACSTAT tuple.
<i>Dump 1</i> =	The rIFACSTAT tuple.
<i>Dump 2</i> =	The associated rVPIDB_F tuple.
<i>Dump 3</i> =	The associated DS1 rSMEST tuple.
<i>Dump 4</i> =	The associated VT1.5 rSMEST tuple.
<i>Dump 5</i> =	The associated STS1 rSMEST tuple.
<i>Dump 6</i> =	The associated STE rSMEST tuple.

Table 9 Error Code: SMEST

Error Code:	SMEST
Error Description:	The SM portion of the FACSTAT audit was unable to obtain an rSMEST tuple for either a DS1, VT1.5, STS-1, or STE facility. 0 = The audit was unable to read rSMEST for a DS1 facility. 1 = The audit was unable to read rSMEST for a VT1.5 facility. 2 = The audit was unable to read rSMEST for a STS-1 facility. 3 = The audit was unable to read rSMEST for a STE facility.
Possible Error Effect:	The contents of the rIFACSTAT tuple can not be audited and is therefore subject

	to corruption.
Corrective Action Taken:	The SMEST audit is scheduled to correct the RLSMEST relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIFACSTAT tuple under audit.
<i>Bad Data =</i>	This is the circuit index of the DNU_S facility.
<i>Good Data =</i>	This is the circuit name of the DNU_S facility.
<i>Logical Index =</i>	This is the value of KEY as indicated above.
<i>Dump 1 =</i>	The rIFACSTAT tuple.
<i>Dump 2 =</i>	The associated rVPIDB_F tuple.

FACSTAT-B

Software Release: 5E12 and later

1. FACILITY STATUS FOR FAST PORT TERMINATION (FACSTAT) AUDIT

The facility status for fast port termination is used to keep the inhibit/allow traffic status for all facilities connected to a given time slot interchanger (TSI) link. This information is used by call processing to determine quickly if a port is available rather than just assigning a port and having it fail later in the process.

The FACSTAT audit is responsible for ensuring the integrity of tuples in the RLFACSTAT relation. For the digital networking unit-synchronous optical network (SONET) (DNU-S), this relation contains bitmaps that indicates whether the virtual peripheral interface data bus (VPIDB) associated with a digital signal 1 (DS1) facility has a carrier group alarm (CGA) or is inhibited from receiving termination attempts. Bits in the bitmaps are audited against the rISMEST tuples associated with facilities represented by this bit position. For DNU-S, each bit position indicates the VPIDB number of the facilities used in port termination.

2. DATA STRUCTURES

For details on the fields contained in the RLFACSTAT relation, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALARM

Error Code:	ALARM
Error Description:	The audit detected that a <i>cga</i> bit associated with a facility is not correct according to the carrier group alarm (CGA) status of that facility.
Possible Error Effect:	The CGA processing for this facility may be incorrect.
Corrective Action Taken:	The <i>cga</i> bit in the rFACSTAT tuple is corrected to the proper value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rFACSTAT tuple that is corrupted.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The position of the corrupted bit (that is, bit mask position).
<i>Logical Index</i> =	The logical key of the corrupted rFACSTAT tuple.
<i>Dump</i> =	The corresponding AUFACSDUMP audit structure.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The FACSTAT audit has encountered an unknown facility type connected to a PLI link.
Possible Error Effect:	Inhibit and allow processing in rFACSTAT for this unknown facility will not be processed correctly.

Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the buffer to hold the corresponding rIPLI_LNK tuple.
<i>Bad Data</i> =	The <i>unitttype</i> field from the corresponding rIVPIDB_F tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The corresponding <i>tsi_link</i> value.
<i>Dump 1</i> =	The corresponding rIFACSTAT tuple.
<i>Dump 2</i> =	The corresponding rIVPIDB_F tuple.

Table 3 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The FACSTAT audit could not obtain the rICGACKT tuple for a facility connected to a DLTU.
Possible Error Effect:	The fast port inhibit feature for this facility may not be functioning properly.
Corrective Action Taken:	The CGACKT audit will be scheduled as a related audit to correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rISMEST tuple.
<i>Bad Data</i> =	The <i>unitttype</i> field from the corresponding rIPICB tuple.
<i>Good Data</i> =	The corresponding facility number.
<i>Logical Index</i> =	The circuit name of the facility.
<i>Dump 1</i> =	The corresponding rISMEST tuple.
<i>Dump 2</i> =	The corresponding rIPICB tuple.

Table 4 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	The FACSTAT audit found that the corresponding inhibit bit for a facility in the rIFACSTAT was not correct.
Possible Error Effect:	The fast port inhibit feature for this facility may not be working correctly.
Corrective Action Taken:	The inhibit status for this facility is corrected in the corresponding rIFACSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rIFACSTAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The bit mask showing the position of the bad bit value in rIFACSTAT.
<i>Logical Index</i> =	The logical key of the rIFACSTAT tuple.
<i>Dump</i> =	The AUFACSDUMP audit structure.

Table 5 Error Code: NCKTS

Error Code:	NCKTS
Error Description:	The FACSTAT audit was unable to obtain the rICGACKT tuple for a facility that is connected to a PLI link.
Possible Error Effect:	The fast port inhibit processing for this facility may not be processed correctly.
Corrective Action Taken:	The CGACKT audit is scheduled as a related audit to correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rICGACKT tuple returned from DA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The vpidb number associated with the facility.
<i>Logical Index</i> =	The circuit name of the facility.
<i>Dump 1</i> =	The corresponding rIFACSTAT tuple.
<i>Dump 2</i> =	The corresponding rIVPIDB_F tuple.

Table 6 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The FACSTAT audit was not able to obtain needed static data for a facility capable of fast port inhibit. The dump will vary depending on the value of <i>Good Data</i> and <i>Error Address</i> .
Possible Error Effect:	Fast port inhibit for this facility may not function correctly.
Corrective Action Taken:	None.

Manual Action Required:	Office personnel must insert the missing static data.
Dump Description:	
<i>Error Address</i> =	If <i>Good Data</i> is zero, this is the address of the buffer to hold the missing rPLI_LNK tuple. Otherwise, it is the relation ID of the static tuple missing, either RLCKTDATA or RLUNIT_CKT.
<i>Bad Data</i> =	If <i>Good Data</i> is zero, this is the return from DBfrdtup. Otherwise, it is the <i>unitnumber</i> associated with the facility.
<i>Good Data</i> =	This is zero or the value PCDS1SFAC.
<i>Logical Index</i> =	If <i>Good Data</i> is zero, this is the <i>tsi_link</i> value associated with the facility. Otherwise, it is the circuit name of the facility.
<i>Dump 1</i> =	This is the corresponding rFACSTAT tuple.
<i>Dump 2</i> =	If <i>Good Data</i> is not zero, this is the dump of the corresponding rVPIDB_F tuple.

Table 7 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The FACSTAT audit could not read the associated rSMEST tuple for a DLTU connected facility.
Possible Error Effect:	The fast port inhibit processing for this feature may not be working properly.
Corrective Action Taken:	The CKTDATA audit is scheduled as a related audit to correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the buffer to receive the rPICB tuple.
<i>Bad Data</i> =	The associated <i>unittype</i> of the facility.
<i>Good Data</i> =	The associated facility number.
<i>Logical Index</i> =	The circuit name of the facility.
<i>Dump 1</i> =	The associated rPIDB for this facility.
<i>Dump 2</i> =	The associated rPICB for this facility.

Table 8 Error Code: SMEST

Error Code:	SMEST
Error Description:	The FACSTAT audit was not able to obtain the associated rSMEST tuple for a DNU DS1 facility.
Possible Error Effect:	Fast port inhibit will probably not work correctly for this DS1 facility.
Corrective Action Taken:	The CKTDATA audit is scheduled as a related audit to correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The circuit index for the DS1 facility.
<i>Bad Data</i> =	The <i>unitnumber</i> associated with this facility.
<i>Good Data</i> =	The value of PCDS1SFAC.
<i>Logical Index</i> =	The DS1 facility circuit name.
<i>Dump 1</i> =	The associated rFACSTAT tuple.
<i>Dump 2</i> =	The associated rVPIDB_F tuple.

FARPID

Software Release: 5E10 and later

1. FAR PROCESS ID AUDIT (FARPID)

A plain old telephone service (POTS) telephone call involves linkages between two terminal processes. This linkage is not implemented by means of one-to-one or owner-member linkage schemes. Instead, two terminal processes are linked when the system places an audit identification mnemonic (ID) within the process control block linkage area of each process. This ID identifies the other process, called the "far process"; the ID is called the far process ID. Since feature calls may involve more than two parties, each process control block linkage area contains two far process IDs. For example, each process in a three-way call contains two far process IDs. Far process IDs are also stored in associated data structures such as the associated process block relation (RLAPB), the general purpose annexed data block relation (RLGPAXDB), and the call leg data block relation (RLCLDB) in the switching module (SM). For POTS calls, the second far process ID is null, which indicates that it is not linked to another process.

Terminal processes involved in a call communicate by means of the message mechanism provided by the operating system for distributed switching (OSDS). For example, when the originating party hangs up, the corresponding process sends a message to the far process that tells it to disconnect the call and terminate. If the ID for the far process is incorrect, the far process will not receive the message. The far process is stuck and its resources (port, time slot, and so forth) are lost.

The FARPID audit verifies the far process linkage for all stable terminal processes by examining the far process ID items of the process control block linkage areas and associated data. Processes are either transient or stable, depending on the content of the customer state (custat) item in the process control block linkage area of a process (for example, dialing is transient and talking is stable). This audit checks linkages only within a given switching module (SM). E-bit discontinuity checks handle inter-SM linkages.

This audit is part of the feature control audits subgroup.

2. DATA STRUCTURES

The data structures checked by the FARPID audit are the process control block linkage area relation (RLPCBLA), the RLGPAXDB, the RLCLDB, and the RLAPB. Figure 1 illustrates the layout of a process control block linkage area. The two items checked by the audit are the far process ID fields (farpid and fpid2). The data is used in this way during POTS calls. Figure 2 illustrates how far processes are monitored for business and residential customer services (BRCS), advanced mobile phone service (AMPS), and operator services position system (OSPS) type calls. For BRCS calls, the far process data is stored in rICLDB tuples linked from the rIPCBLA. For AMPS calls, the far process data is stored in rIGPAXDB tuples linked from the rIPCBLA. For OSPS calls, the far process data is stored in rIAPB tuples linked from the rIPCBLA.

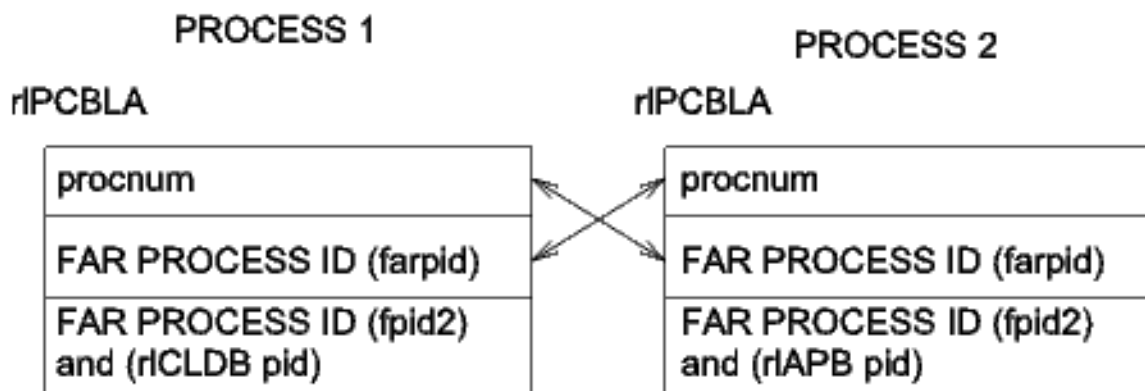


Figure 1 (FARPID-1) Process Control Block Linkage Area Far Process ID Fields

PROCESS 1

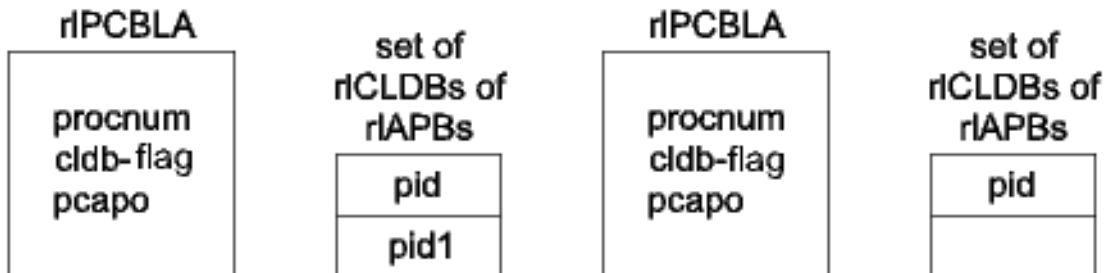


Figure 2 (FARPID-2) Process Control Block Linkage Area Far Process ID Fields

3. AUDIT CHECKS

The FARPID audit calls the static driver (DRSTATIC) once for each audit segment. The DRSTATIC driver accesses each process control block linkage area and performs the following semantic checks for each process field:

- If the process control block linkage area is transient, a headcell, a system process, or associated with a host switching module (HSM) terminal process, nothing is checked and the next process control block linkage area is accessed.
- If the process control block linkage area is stable, the process control block linkage area far process ID(s) is examined. If a far process ID is null or exists in a different processor, no further checks are made for that far process ID. If a far process ID contains a value other than null, the audit verifies that the specified far process actually exists. If it does not exist, control is returned to the DRSTATIC driver with an error indication.
- If the specified far process exists in the current processor, the audit verifies that the far process ID of the far process process control block linkage area contains the ID of the original process (point-back check). If it does not, control is returned to the static driver (DRSTATIC) driver with an error indication.
- If the process control block linkage area is stable and more than ten minutes have passed since the last time that Keep-Alive checking was done for an AMPS terminal process, do the following:
 - If one of the far processes is an *AUTOPLEX*TM (APX) system process and the class of service of that far processes points to a portless APX process and the keep-alive bit (pcblaka) is set to DBYES, send an MGAXCHK message to the APX system process on the far process' SM so the system process verifies that the far party is still active.
 - If the far process is an APX terminal process conference circuit, send an MGAXCHK message to the APX system process on the far process' SM so the system process verifies that the far party is still active.
 - If this process is an APX terminal process and owns a conference circuit, send an MGAXCHK message to the APX terminal process so that process checks its far parties to see if they are still active.

4. ERROR HANDLING

If a process control block linkage area is found to have a far process ID that does not point to a valid far process, the corresponding process is purged. If the far process ID of a far process process control block linkage area does not point back to the original process, the original process is purged.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

Table 1 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The rIPCBLA tuple associated with this process could not be accessed.
Possible Error Effect:	The rIPCBLA tuple may be a lost resource, unusable by call processing.
Corrective Action Taken:	Related audit PCBLA is scheduled to check for consistent states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPCBLA.
<i>Bad Data =</i>	The process number of the rIPCBLA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The process number of the rIPCBLA tuple.
<i>Dump =</i>	The rIPCBLA tuple.

Table 2 Error Code: FPID_LINK

Error Code:	FPID_LINK
Error Description:	The far process ID (FARPID) being checked does not claim to be associated with the pcbla being checked. No lists owned by the far terminal process' PCBLA have the near terminal process process ID (PID) specified in it. This process is thus considered lost.
Possible Error Effect:	A terminal associated with this process is invalidly denied service. In addition, many occurrences of this type could cause the system to run out of processes.
Corrective Action Taken:	The process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPCBLA tuple when Good Data is zero else the bad process ID.
<i>Bad Data =</i>	The key of the tuple containing the far process.
<i>Good Data =</i>	If Dump 2 indicates GPAXDB, 1 for FAR1 2 for FAR2 3 for FAR3 otherwise it is 0.
<i>Logical Index =</i>	procno of this pcbla.
<i>Dump 1 =</i>	rIPCBLA.
<i>Dump 2 =</i>	Additional data requested, identified in the dump.
<i>Dump 3 =</i>	When Dump 2 indicates GPAXDB, rIPCBLA for far party.

Table 3 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The OSDS ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUfar_time, used for transient timing may have occurred.
Possible Error Effect:	The audit's transient timing has been interrupted. Lost resources may take longer to recover because transient timing must be restarted.
Corrective Action Taken:	AUfar_time will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the transient timing global variable (AUfar_time).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Content of the transient timing variable (AUfar_time).
<i>Dump 2 =</i>	Content of the current OSDS ITIME (OSnow).

FCASITBL

Software Release: 5E10 and later

1. FEATURE CONTROL ASI UTD TABLE (FCASITBL) AUDIT

Advanced service access (ASI) is a switch based capability which will provide the telephone operating companies with the ability to connect an intelligent peripheral (IP) to subscribers on the switch. The telephone operating companies would utilize IP to support a variety of services for the purpose of providing automated call handling for the subscriber. The FCASITBL audit will insure the integrity of the FCasitbl[] array structure that is used to hold the PIDs of processes using ASI universal tone decoder (UTD) resources.

2. DATA STRUCTURES

The FCASITBL audit references the following attributes of the FCasitbl[] global array:

pcrid = The processor where the owning UTD process is resident.
 procno = The process number of the owning UTD process.
 uniq = The uniqueness attribute of the owning UTD process.

In addition the audit references the following global values:

AUutdindx = Pointer to the entry in FCasitbl[] currently being audited or FCTBLSTR.
 AUutd_use = Audits count of the valid processes owning a UTD currently in the FCasitbl[] array.
 FCHUNDRD = Global constant used to determine the number of UTDs in use as a percentage.
 FCMAXUTD = Global constant used to determine the maximum number of number of entries in the FCasitbl[] array.
 FCTBLALL = Global constant used to define the first index into the FCasitbl[] array.
 FCTBLSTR = Global constant used to define the first index into the FCasitbl[] array that may contain a valid PID.
 FCTBLEND = Global constant used to define the last index into the FCasitbl[] array that may contain a valid PID.
 FClstrel = Pointer to the last relased entry in FCasitbl[] or the last available entry.
 FCutd_use = Count of the valid processes owning a UTD currently in the FCasitbl[] array.
 Glasiulim = The maximum number of UTDs that can be allocated in the FCasitbl[] array.
 GI_utd = Global constant used to determine number of UTDs in use as a percentage.

All entries in the Fcasitbl[] array should be marked OSnullpid unless they are currently in use. A maximum of (Glasiulim * GI_utd) entries may be in use at any one time. The FCasitbl[FCTBLALL] entry is the first entry in the table and is never used (must always be set to OSnullpid). FClstrel should contain the index into FCasitbl[] array whose entry is OSnullpid or FCTBLALL. If FClstrel is set to FCTBLALL, a search for the next OSnullpid entry in the FCasitbl[] array will be done. FCutd_use should always contain the number of valid PIDs in the FCasitbl[] array using UTD resources.

3. AUDIT CHECKS

The FCASITBL audit makes the following checks: If the FCASITBL audit is entered:

- NOTIDLE = Check that the entry FCasitbl[FCTBLALL] is OSnullpid.
- OUTOFRANGE = Check that the FClstrel pointer is within range.
- ZAP = Check to see if audit was entered as the result of a reset.

For every entry in the FCasitbl[] array:

- If the entry is not OSnullpid:
 - BDPID = Check that the process in the entry is a valid process.
 - INDEX = Check that the FClstrel pointer is pointing to a valid null entry in FCasitbl[].
- If the entry is OSnullpid:
 - BDCOUNT = Check that the FCutd_use count is valid.
 - RANGE = Check that the FCutd_use is within a valid range.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The global count, FCutd_use, which counts the number of valid PIDs in the FCasitbl[] array did not match the audit count, AUutd_use, after stepping through the array entries.
Possible Error Effect:	A wrong usage count could lead to over or under allocation of UTD resources in the switch.
Corrective Action Taken:	The usage count, FCutd_use, is set to the value of the audit count, AUutd_use.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the global count, FCutd_use.
<i>Bad Data =</i>	The incorrect value of FCutd_use.
<i>Good Data =</i>	The correct value of the global count, AUutd_use.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: BDPID

Error Code:	BDPID
Error Description:	A PID no longer associated with a valid process was found in an FCasitbl[] array entry.
Possible Error Effect:	A process that has abnormally ended failed to release its UTD resource. This may prevent other processes from using this UTD resource.
Corrective Action Taken:	The FCasitbl[] array entry found to be in error is set to OSnullpid.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the FCasitbl[] array entry found to contain an invalid PID.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The value of OSnullpid.
<i>Logical Index</i> =	The index into the FCasitbl[] array containing the entry in error.
<i>Dump</i> =	The value of the FCasitbl[] array entry at the time of error.

Table 3 Error Code: INDEX

Error Code:	INDEX
Error Description:	In stepping through the FCasitbl[] array, it was found that the global variable FClstrel contains an index of a non-null entry (not OSnullpid). Since FClstrel is used to assign the next null entry in FCasitbl[] array a valid PID, it should always point to a null entry or contain FCTBLALL (indicating a search must be done for a null entry).
Possible Error Effect:	This error could cause the reallocation of a UTD resource that is currently allocated to another process.
Corrective Action Taken:	FClstrel is set to FCTBLALL to indicate a search needs to be done for the next OSnullpid entry in the FCasitbl[] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the global variable, FClstrel.
<i>Bad Data</i> =	The value of FClstrel that was in error.
<i>Good Data</i> =	The value of FCTBLALL.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The entry FCasitbl[FCTBLALL] was not set to OSnullpid. This entry should always be set to OSnullpid since it is never used.
Possible Error Effect:	The FCasitbl[] array may be corrupt leading to lost UTD resources.
Corrective Action Taken:	The FCasitbl[FCTBLALL] entry is set to OSnullpid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the FCasitbl[FCTBLALL] entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The value of OSnullpid.
<i>Logical Index</i> =	The value of FCTBLALL.
<i>Dump</i> =	The value of FCasitbl[FCTBLALL] at the time of error.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A check on the index pointing to the last entry released, FClstrel, was made, and this global variable was found to be outside the range of the FCasitbl[] array. FClstrel should contain the index into FCasitbl[] containing an OSnullpid entry or FCTBLALL.
Possible Error Effect:	The FCasitbl[] array may be corrupt leading to lost UTD resources.
Corrective Action Taken:	The global variable, FClstrel, is set to FCTBLALL to indicate a search needs to be done to find the next OSnullpid entry in the FCasitbl[] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the FClstrel global variable.
<i>Bad Data</i> =	The value of FClstrel at the time of error.
<i>Good Data</i> =	The value of FCTBLALL.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 6 Error Code: RANGE

Error Code:	RANGE
Error Description:	The count of valid PIDs in FCasitbl[], FCutd_use, was found to be out of range

	with the limits set by the global variables Glasiulim and Gl_ utd.
Possible Error Effect:	This condition may lead to over allocation of UTD resources.
Corrective Action Taken:	None.
Manual Action Required:	Users should verify that the UTD usage parameters set by recent change are correct or need to be increased. Also, the global limit parameters Glasiulim and Gl_ utd may have been corrupted. Normally, this condition would indicate a severe sanity problem in the effected SM.
Dump Description:	
<i>Error Address</i> =	The address of the FCutd_ use variable.
<i>Bad Data</i> =	The value of (FCutd_ use * FCHUNRD) at the time of error.
<i>Good Data</i> =	The value of (Glasiulim * Gl_ utd) at the time of error.
<i>Logical Index</i> =	The value of Gl_ utd at the time of error.
<i>Dump</i> =	NA.

FCLINE

Software Release: 5E10 and later

1. INDIVIDUAL LINE FEATURE DATA (FCLINE) AUDIT

Local area signaling services (LASS) features make use of a list of telephone numbers, usually directory numbers (DNs), associated with the user. When the user receives a call from a DN on such a list, special treatment is provided to the call. The "client" features requiring such lists are:

Computer Access Restriction (CAR) = Enables a customer to receive circuit switched data calls only from calling parties whose identity attributes have been put on the customer's screening list.

Selective Call Acceptance (SCA) = Allows customers to accept calls only from parties whose identity attributes are on the customers SCA list.

Selective Call Forwarding (SCF) = Enables a customer to designate a SCF list whereby call parties whose identity attributes are contained on the list are forwarded to individually designated remote numbers as specified in the SCF list.

Selective Call Rejection (SCR) = Enables customers to receive voice calls only from parties whose identity attributes are contained in an SCR list.

Selective Distinct Alert (SDA) = Enables customers to select, through the use of an SDA list, which calling parties will be provided distinctive alerting treatment.

Screen List Editing (SLE) = Enables users to interactively create, activate, edit and delete screening lists of identity attributes associated with the customer's DN.

The RLFC_LINE relation is used by several LASS features to indicate whether or not a particular line should have access to a LASS feature. The FCLINE audit runs in all switching modules (SMs) in segmented mode and is also scheduled as a post initialization audit as part of a selective initialization.

2. DATA STRUCTURES

The RLFC_LINE relation is a static relation used in conjunction with the static global variables Glac_access (automatic callback), Glar_access (automatic recall), Glcot_access (customer originated trace), Glscrac (selective call rejection), Glscac (selective call acceptance), Glscfac (selective call forwarding), and Glstdaac (selective distinctive alert), and the static relation RLTRMFEATS [where rIFC_LINE.lpt_type (key) = rITRMTFEATS.lpt_type (key)] to determine if tuples in the dynamic line history block relations RLAN_ILHB, RLAN_OLHB, RLIS_ILHB, and RLIS_OLHB should exist. If any of the global variables are marked UNAVAILABLE or TEMP_UNAVAIL then the dynamic relations should not exist and no auditing is done. Otherwise, static-to-dynamic existence relationships exist between RLFC_LINE and the dynamic relations RLAN_ILHB, RLAN_OLHB, RLIS_ILHB, and RLIS_OLHB according to the following rules:

- If Glar_access is marked GEN_AVAIL (generally available) then RLAN_OLHB or RLIS_OLHB should exist depending on whether the port is analog or ISDN. If ISDN, then rIFC_LINE.lpt_type = LPT_ISDN; if analog, rIFC_LINE.lpt_type = LPT_OTHER. In both cases the RLFC_LINE brc_trm and trmbfg attributes are DBYES and the key for RLFC_LINE (rIFC_LINE.lport) is that same as lport (key) in RLAN_OLHB or RLIS_OLHB.
- If Glac_access is marked GEN_AVAIL (generally available) then RLAN_ILHB or RLIS_ILHB should exist depending on whether the port is analog or ISDN. If ISDN, then rIFC_LINE.lpt_type = LPT_ISDN; if analog, rIFC_LINE.lpt_type = LPT_OTHER. In both cases the RLFC_LINE brc_trm and trmbfg attributes are DBYES and the key for RLFC_LINE (rIFC_LINE.lport) is that same as lport (key) in RLAN_ILHB or RLIS_ILHB.
- If any of the remaining global variables listed are marked GEN_AVAIL (generally available) then RLAN_ILHB or

RLIS_ILHB should exist depending on whether the port is analog or ISDN. If ISDN, then rIFC_LINE.lpt_type = LPT_ISDN; if analog, rIFC_LINE.lpt_type = LPT_OTHER. In both cases the RLFC_LINE brc_trm and trmbfg attributes are DBYES and the key for RLFC_LINE (rIFC_LINE.lport) is that same as lport (key) in RLAN_ILHB or RLIS_ILHB.

3. AUDIT CHECKS

The FCLINE audit uses the DRSTATIC driver, and for each rIFC_LINE tuple makes the following checks: If any of the global variables are marked UNAVAILABLE or TEMP_UNAVAIL then no checks are done.

If **none** of the global variables are marked UNAVAILABLE or TEMP_UNAVAIL:

ANILHBKEY	= Check that the keys agree for RLFC_LINE and RLAN_ILHB.
ANOLHBKEY	= Check that the keys agree for RLFC_LINE and RLAN_OLHB.
CL	= Check for correct population of Glcot_access.
HTYPE	= Check that RLFC_LINE lpt_type is LPT_OTHER or LPT_ISDN when brc_trm and trmbfg are both DBYES for AR features.
ISILHBKEY	= Check that the keys agree for RLFC_LINE and RLIS_ILHB.
ISOLHBKEY	= Check that the keys agree for RLFC_LINE and RLIS_OLHB.
LLOSTISILHB	= Check that an rIIS_ILHB tuple exists for this rIFC_LINE tuple for AC, COT, LSF, and ICLID features.
LOSTANOLHB	= Check that an rIAN_OLHB tuple exists for this rIFC_LINE tuple.
LOSTISOLHB	= Check that an rIIS_OLHB tuple exists for this rIFC_LINE static tuple with the AR feature assigned.
RANGE	= Check that RLFC_LINE lpt_type is LPT_OTHER or LPT_ISDN when brc_trm and trmbfg are both DBYES for AC, COT and LSF (Lass Selective) features.
SGNODD	= Check for correct population of Glac_access.
TYPE	= Check for correct population of Glar_access.
XLOSTANILHB	= Check that an rIAN_ILHB tuple exists for this rIFC_LINE tuple for AC, COT, LSF, and ICLID features.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ANILHBKEY

Error Code:	ANILHBKEY
Error Description:	The audit has detected an inconsistency between the lport attribute in the rIFC_LINE tuple and the lport attribute of the associated dynamic rIAN_ILHB

	tuple.
Possible Error Effect:	The rIAN_ILHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS features ICLID, COT, LSF, and AC.
Corrective Action Taken:	The audit will report an error and correct the dynamic rIAN_ILHB to agree with the static rIFC_LINE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Local address of the rIAN_ILHB tuple.
<i>Bad Data</i> =	lport attribute of the dynamic rIAN_ILHB tuple.
<i>Good Data</i> =	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index</i> =	port.member attribute of the rIFC_LINE tuple in question.
<i>Dump</i> =	The static data rIFC_LINE tuple in question.

Table 2 Error Code: ANOLHBKEY

Error Code:	ANOLHBKEY
Error Description:	The audit has detected an inconsistency between the lport attribute in the rIFC_LINE tuple and the lport attribute of the associated dynamic rIAN_OLHB tuple.
Possible Error Effect:	The rIAN_OLHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS feature AR.
Corrective Action Taken:	The audit will report an error and correct the dynamic rIAN_OLHB to agree with the static rIFC_LINE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Local address of the rIAN_OLHB tuple.
<i>Bad Data</i> =	lport attribute of the dynamic rIAN_OLHB tuple.
<i>Good Data</i> =	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index</i> =	port.member attribute of the rIFC_LINE tuple in question.
<i>Dump</i> =	The static data rIFC_LINE tuple in question.

Table 3 Error Code: CL

Error Code:	CL
Error Description:	The audit has found the static global variable Glocot_access is populated incorrectly.
Possible Error Effect:	The audit will not be able to complete its checks of related line history blocks (rIAN_ILHB, rIIS_ILHB) associated with the LASS feature COT or LSF.
Corrective Action Taken:	The audit will report an error and continue on to the next rIFC_LINE tuple.
Manual Action Required:	The global parameter Glocot_access must be checked for correctness. If the parameter is incorrect, the audit will continue to print this error.
Dump Description:	
<i>Error Address</i> =	Not an address; value of local acflag or lsfflag used indicate if an associated rIAN_ILHB or rIIS_ILHB should exist to support the LASS feature AC.
<i>Bad Data</i> =	Value of Glocot_access.
<i>Good Data</i> =	cot_alw attribute of the static rIFC_LINE tuple; used to indicate whether COT or LSF is available for a line.
<i>Logical Index</i> =	lport field of the static rIFC_LINE tuple.
<i>Dump</i> =	The static data rIFC_LINE tuple.

Table 4 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The audit has detected an inconsistency between the static data in rIFC_LINE. The inconsistency indicates that the lpt_type attribute is not consistent with the brc_trm and/or the trmbfg attributes.
Possible Error Effect:	The audit cannot complete its checks between a rIFC_LINE tuple and any related line history block tuple (rIAN_OLHB, rIIS_ILHB etc.).
Corrective Action Taken:	The audit will report an error and continue on to the next rIFC_LINE tuple.
Manual Action Required:	The rIFC_LINE tuple in question must be examined to determine whether the brc_trm, trmbfg, and lpt_type attributes are populated correctly. If they are not correct, the audit will continue to report this error until they are corrected.
Dump Description:	
<i>Error Address</i> =	Not an address, this is the value of the trmbfg attribute of the rIFC_LINE tuple in

	question.
<i>Bad Data</i> =	lpt_type attribute of the rIFC_LINE tuple in question.
<i>Good Data</i> =	brc_trm attribute of the rIFC_LINE tuple in question.
<i>Logical Index</i> =	lport attribute of the rIFC_LINE tuple in question.
<i>Dump</i> =	The static data rIFC_LINE tuple in question.

Table 5 Error Code: ISILHBKEY

Error Code:	ISILHBKEY
Error Description:	The audit has detected an inconsistency between the lport attribute in the rIFC_LINE tuple and the lport attribute of the associated dynamic rIIS_ILHB tuple.
Possible Error Effect:	The rIIS_ILHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS features ICLID, COT, LSF, and AC.
Corrective Action Taken:	The audit will report an error and correct the dynamic rIIS_ILHB to agree with the static rIFC_LINE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Local address of the rIIS_ILHB tuple.
<i>Bad Data</i> =	lport attribute of the dynamic rIIS_ILHB tuple.
<i>Good Data</i> =	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index</i> =	port.member attribute of the rIFC_LINE tuple in question.
<i>Dump</i> =	The static data rIFC_LINE tuple in question.

Table 6 Error Code: ISOLHBKEY

Error Code:	ISOLHBKEY
Error Description:	The audit has detected an inconsistency between the lport attribute in the rIFC_LINE tuple and the lport attribute of the associated dynamic rIIS_OLHB tuple.
Possible Error Effect:	The rIIS_OLHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS feature AR.
Corrective Action Taken:	The audit will report an error and correct the dynamic rIIS_OLHB to agree with the static rIFC_LINE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Local address of the rIIS_OLHB tuple.
<i>Bad Data</i> =	lport attribute of the dynamic rIIS_OLHB tuple.
<i>Good Data</i> =	lport attribute of the static rIIS_LINE tuple.
<i>Logical Index</i> =	port.member attribute of the rIFC_LINE tuple in question.
<i>Dump</i> =	The static data rIFC_LINE tuple in question.

Table 7 Error Code: LLOSTISILHB

Error Code:	LLOSTISILHB
Error Description:	The audit has found that an associated dynamic rIIS_ILHB tuple is either missing or is garbaged to 0xffff.
Possible Error Effect:	The rIIS_ILHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS features ICLID, COT, LSF, and AC.
Corrective Action Taken:	The audit will report an error and correct/insert the dynamic rIIS_ILHB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index</i> =	lport field of the dynamic rIIS_ILHB tuple in question.
<i>Dump</i> =	The static data rIFC_LINE tuple.

Table 8 Error Code: LOSTANOLHB

Error Code:	LOSTANOLHB
Error Description:	The audit has found that an associated dynamic rIAN_OLHB tuple is either missing or is garbaged to 0xffff.
Possible Error Effect:	The rIAN_OLHB tuple may not be able to be accessed correctly. This may affect

	calls associated with LASS feature AR.
Corrective Action Taken:	The audit will report an error and correct/insert the dynamic rIAN_OLHB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index =</i>	lport field of the dynamic rIAN_OLHB tuple in question.
<i>Dump =</i>	The static data rIFC_LINE tuple.

Table 9 Error Code: LOSTISOLHB

Error Code:	LOSTISOLHB
Error Description:	The audit has found that an associated dynamic rIIS_OLHB tuple is either missing or is garbaged to 0xffff.
Possible Error Effect:	The rIIS_OLHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS feature AR.
Corrective Action Taken:	The audit will report an error and correct/insert the dynamic rIIS_OLHB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index =</i>	lport field of the dynamic rIIS_OLHB tuple in question.
<i>Dump =</i>	The static data rIFC_LINE tuple.

Table 10 Error Code: RANGE

Error Code:	RANGE
Error Description:	The audit has detected an inconsistency between the static data in rIFC_LINE. The inconsistency indicates that the lpt_type attribute is not consistent with the brc_trm and/or the trmbfg attributes.
Possible Error Effect:	The audit cannot complete its checks between a rIFC_LINE tuple and any related line history block tuple (rIAN_OLHB, rIIS_ILHB etc.)
Corrective Action Taken:	The audit will report an error and continue on to the next rIFC_LINE tuple.
Manual Action Required:	The rIFC_LINE tuple in question must be examined to determine whether the brc_trm, trmbfg, and lpt_type attributes are populated correctly. If they are not correct, the audit will continue to report this error until they are corrected.
Dump Description:	
<i>Error Address =</i>	Not an address, this is the value of the brc_trm attribute of the rIFC_LINE tuple in question.
<i>Bad Data =</i>	lpt_type attribute of the rIFC_LINE tuple in question.
<i>Good Data =</i>	trmbfg attribute of the rIFC_LINE tuple in question.
<i>Logical Index =</i>	lport attribute of the rIFC_LINE tuple in question.
<i>Dump =</i>	The static data rIFC_LINE tuple in question.

Table 11 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	The audit has found the static global variable Glar_access is populated incorrectly.
Possible Error Effect:	The audit will not be able to complete its checks of related line history blocks (rIAN_ILHB, rIIS_ILHB) associated with the LASS feature AC.
Corrective Action Taken:	The audit will report an error and continue on to the next rIFC_LINE tuple.
Manual Action Required:	The global parameter Glac_access must be checked for correctness. If the parameter is incorrect, the audit will continue to print this error.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Value of Glac_access.
<i>Good Data =</i>	ac_alw attribute of the static rIFC_LINE tuple; used to indicate whether AC is available for a line.
<i>Logical Index =</i>	lport field of the static rIFC_LINE tuple.
<i>Dump =</i>	The static data rIFG_LINE tuple.

Table 12 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit has found the static global variable Glar_access is populated incorrectly.
Possible Error Effect:	The audit will not be able to complete its checks of related line history blocks (rIAN_OLHB, rIIS_OLHB) associated with the LASS feature AR.
Corrective Action Taken:	The audit will report an error and continue on to the next rIFC_LINE tuple.
Manual Action Required:	The global parameter Glar_access must be checked for correctness. If the parameter is incorrect, the audit will continue to print this error.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Value of Glar_access.
<i>Good Data =</i>	ar_alw attribute of the static rIFC_LINE tuple; used to indicate whether AR is available for a line.
<i>Logical Index =</i>	lport field of the static rIFC_LINE tuple.
<i>Dump =</i>	The static data rIFC_LINE tuple.

Table 13 Error Code: XLOSTANILHB

Error Code:	XLOSTANILHB
Error Description:	The audit has found that an associated dynamic rIAN_ILHB tuple is either missing or is garbaged to 0xffff.
Possible Error Effect:	The rIAN_ILHB tuple may not be able to be accessed correctly. This may affect calls associated with LASS features ICLID, COT, LSF, and AC.
Corrective Action Taken:	The audit will report an error and correct/insert the dynamic rIAN_ILHB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	lport attribute of the static rIFC_LINE tuple.
<i>Logical Index =</i>	lport field of the dynamic rIAN_ILHB tuple in question.
<i>Dump =</i>	The static data rIFC_LINE tuple.

FCTRUNK

Software Release: 5E10 and later

1. FEATURE DATA FOR TRUNKS RELATION (FCTRUNK) AUDIT

The feature data for trunks relation, RLFC_TRUNK, is used by call processing operations in the switch. The relations contains trunk related information such as trunk port, group and member data, equipment unit type, and unit number. Since RLFC_TRUNK is a static relation's contents are used as the master data for comparison against various dynamic relations. The FCTRUNK audit provides this audit functionality. The FCTRUNK audit resides in all switching modules (SMs) and runs in routine and elevated segmented modes.

2. DATA STRUCTURES

One tuple exists in the RLFC_TRUNK relation for every trunk member in the SM. The FCTRUNK audit performs checks based on the following fields:

- port* = The key of the rIFC_TRUNK tuple. For every trunk port assigned the automated call assurance and hundred second scan (ACA and HSS) feature, a corresponding tuple should exist in the rIACA_HSS dynamic relation, where rIFC_TRUNK.port.member == rIACA_HSS.port and rIFC_TRUNK.port.module == GI_pcrd.
- tgroup* = The trunk group to which the member assigned to the port belongs, where rIFC_TRUNK.tgroup represents a tuple in the RLRT_TRKG static relation, and rIFC_TRUNK.tgroup == rIRT_TRKG.grpnum.

3. AUDIT CHECKS

the following checks: If the FCTRUNK audit gets entered:

- KEY = Verify that the key stored in a rIACA_HSS tuple is valid.
- LOSTTUP = Verify that a corresponding tuple in the rIACA_HSS dynamic relation exists for a given tuple in the RLFC_TRUNK relation, where rIFC_TRUNK.port.member == rIACA_HSS.port.
- TRKG = Obtain the corresponding tuple in the RLRT_TRKG relation for a given tuple in the RLFC_TRUNK relation, where rIFC_TRUNK.tgroup == rIRT_TRKG.grpnum.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: KEY

Error Code:	KEY
Error Description:	A rIACA_HSS tuple has been found with a corrupted key.
Possible Error Effect:	Asserts out of measurement scanning applications, and invalid trunk measurements.
Corrective Action Taken:	The key of the rIACA_HSS tuple is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACA_HSS tuple found with the corrupted key.
<i>Bad Data =</i>	The key of the rIACA_HSS tuple (rIACA_HSS.port).

<i>Good Data</i> =	The key of the corresponding tuple in the RLFC_TRUNK static relation (rlFC_TRUNK.port.member).
<i>Logical Index</i> =	The key of the corresponding tuple in the RLFC_TRUNK static relation (rlFC_TRUNK.port.member).
<i>Dump 1</i> =	The contents of the rlACA_HSS dynamic tuple found with the corrupted key.
<i>Dump 2</i> =	The contents of the corresponding tuple in the RLFC_TRUNK static relation with key equal to rlFC_TRUNK.port.member.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A rlACA_HSS tuple has been found missing from the database. For every tuple in the RLFC_TRUNK relation with the ACA and HSS feature activated, there should exist a corresponding dynamic tuple in the RLACA_HSS relation (where rlFC_TRUNK.port.member == rlACA_HSS.port, rlRT_TRKG.grpnum == rlFC_TRUNK.tgroup and rlRT_TRKG.nlactive == DBYES OR rlRT_TRKG.sht_thres > 0 OR rlRT_TRKG.lht_thres > 0).
Possible Error Effect:	Asserts out of measurement scanning applications, and invalid trunk measurements.
Corrective Action Taken:	The missing rlACA_HSS tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The key of the corresponding tuple in the RLFC_TRUNK static relation (rlFC_TRUNK.port.member).
<i>Logical Index</i> =	The key of the corresponding tuple in the RLFC_TRUNK static relation (rlFC_TRUNK.port.member).
<i>Dump</i> =	The contents of the tuple in the RLFC_TRUNK static relation with key equal to rlFC_TRUNK.port.member.

Table 3 Error Code: TRKG

Error Code:	TRKG
Error Description:	A rlFC_TRUNK static tuple was found for which no rlRT_TRKG tuple exists.
Possible Error Effect:	Trunk routing failures.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, determine if the trunk group represented by the rlACA_HSS tuple key (trunk port) exists, by reading the RLGROUP_PORT relation with rlGROUP_PORT.portgrp.grpnum set to the group number in RLFC_TRUNK "Bad Data" dumped with the audit message, and the rlGROUP_PORT.portgrp.grptype set to TM_TRK. Set rlGROUP_PORT.member to the member number dumped in the RLACA_HSS dump which comes out with the audit message. If a tuple exists, insert a corresponding tuple in the RLRT_TRKG relation with rlRT_TRKG.grpnum set to rlGROUP_PORT.portgrp.grpnum. If a tuple does not exist, use recent change to delete the ACA and HSS feature from this trunk port. Then remove the trunk port and insert it again. Verify that the rlGROUP_PORT static tuple is created. Finally, add the ACA and HSS feature to the trunk port.
Dump Description:	
<i>Error Address</i> =	The address of the rlACA_HSS tuple.
<i>Bad Data</i> =	The tgroup field in the rlFC_TRUNK static tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rlACA_HSS tuple (rlACA_HSS.port).
<i>Dump 1</i> =	The contents of the rlACA_HSS tuple.
<i>Dump 2</i> =	The contents of the associated tuple in the RLFC_TRUNK relation.

FD

Software Release: 5E10 and later

1. FRAME DESCRIPTOR (FD) AUDIT

Frames are used to transmit data through the quad linked packet switched (QLPS) network. Frame descriptors (FDs), elements of CMfduse[], are used to represent per-frame information such as type and ownership. FDs are passed between data structures by linking them in different lists.

2. DATA STRUCTURES

The following data is used by the FD audit:

CMbufidle	= Buffer idle list.
CMfduse[]	= SM-2000 frame descriptors.
CMspcdb[]	= SPYDER-T channel data block.
CMsptduse[]	= SPYDER-T transmit descriptor.

3. AUDIT CHECKS

The following checks are performed by the FD audit:

ALNK2QUE	= Check for corrupt linkage for an FD in transmit or message queue.
BADLINK	= Check linkages on non-idle FD's.
BADUSE	= Check for out of range FD usage type.
CORRUPT_IDLE	= Check for default attribute values for an idle FD.
HEAD	= Check for corrupt index for an FD in CMbufidle.
IDLE_LINK	= Check for corrupt FD in-use idle stack entry.
INVUSAGE	= Check for owned FD by level 1 not being touched for several minutes.
IS_NOTUSED	= Check for unused entries being non-null.
QUE	= Check for corrupt linkage for an FD in transmit or message queue.
RCVACCBLK	= Check for invalid level 1 linkages for an in-use receive FD.
ZAP	= A reset was taken during the audit.

4. ERROR HANDLING

In general, the audit will clear the FD idle stack and rebuild the idle list.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALNK2QUE

Error Code:	ALNK2QUE
Error Description:	An FD (entry in CMfduse[]) appears to be in either the transmit queue (CMBUFUSE.usetype = CMXMTQ) or message queue (CMBUFUSE.usetype = CMMSGQ), but the linkage to the queue is corrupt: CMXMTQ: • CMBUFUSE.link is not CM_LINK0 or CM_LINK1. • CMBUFUSE.qindx is greater than CM_X25MOD. CMMSGQ: • CMBUFUSE.link is not CM_LINK0 or CM_LINK1. • CMBUFUSE.qindx is greater than CM_OUTQLEN.
Possible Error Effect:	The FD is not available for use. If enough FDs became unavailable, the QLPS network would lose capacity.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Head of the FD idle stack (CMbufidle.toppt).
<i>Bad Data =</i>	FD's user bits (CMBUFUSE.bf_owner).
<i>Good Data =</i>	FD's usage type (CMBUFUSE.usetype).
<i>Logical Index =</i>	FD's index (index into CMfduse[]).
<i>Dump =</i>	FD (entry in CMfduse[]).

Table 2 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	One or more of the following is true of a non-idle (usetype != CMFREES) FD (entry in CMfduse[]): • FD.llcb is out of range ($\geq$ QLNLLCBS). • FD.next_ifd = CMNIL_NDX, but QLllcb[FD.llcb].l_ifd != the FD's index. • FD.back_ifd = CMNIL_NDX, but QLllcb[FD.llcb].f_ifd != the FD's index. • FD.next_ifd != CMNIL_NDX, but is not valid (is $\geq$ CM_NUMBUF). • FD.back_ifd != CMNIL_NDX, but is not valid (is $\geq$ CM_NUMBUF). • FD.next_ifd is $<$ CM_NUMBUF, but CMfduse[FD.next_ifd].back_ifd != the FD's index. • FD.back_ifd is $<$ CM_NUMBUF, but CMfduse[FD.back_ifd].next_ifd != the FD's index.
Possible Error Effect:	The FD is not available for use. If enough FDs became unavailable, the QLPS network would lose capacity.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Head of the FD idle stack (CMbufidle.toppt).
<i>Bad Data</i> =	FD's user bits (CMBUFUSE.bf_owner).
<i>Good Data</i> =	FD's usage type (CMBUFUSE.usetype).
<i>Logical Index</i> =	FD's index (index into CMfduse[]).
<i>Dump 1</i> =	FD (entry in CMfduse[]).
	Up to two of the following dumps will also be provided in the order listed depending on the values of the FD linkage fields:
<i>Dump 2</i> =	FD indexed by CMBUFUSE.next_ifd if next_ifd is < CM_NUMBUF.
<i>Dump 3</i> =	FD indexed by CMBUFUSE.back_ifd if back_ifd is < CM_NUMBUF.
<i>Dump 4</i> =	The first 64 bytes of CMmsgbuff[FD index].
<i>Dump 5</i> =	The owning QLLCB (QLlcb[CMBUFUSE.lcb]) if lcb is < QLNLLCBS.

Table 3 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An FD's usage type is out of range (CMBUFUSE.usetype ≥ CMMAXFDUSE).
Possible Error Effect:	The out-of-range value may cause asserts. In any case, the FD is not processed correctly. Any frame associated with the FD may be dropped.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	FD's usage type (CMBUFUSE.usetype).
<i>Good Data</i> =	CMMAXFDUSE.
<i>Logical Index</i> =	FD's index (index into CMfduse[]).
<i>Dump</i> =	FD (entry in CMfduse[]).

Table 4 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	An idle FD (CMBUFUSE.usetype = CMFREES) does not contain the expected default value in one or more of its attributes: • CMBUFUSE.qindx is out of range (> CMbufidle.toppt). • CMbufidle.fdindx[CMBUFUSE.qindx] != the FD's index. • CMBUFUSE.link != 0. • One or more of the following are != CMNIL_NDX: • l1_prev • l1_next • td_indx [message handler (MH) only] • back_ifd • next_ifd • CMBUFUSE.l1_channel != CMSPCHNULL. • CMBUFUSE.pack_type != -1. • CMBUFUSE.lcb != QLLIXNULL. • CMBUFUSE.length != 0.

	• CMBUFUSE.bf_owner != CM_NO_OWNER.
Possible Error Effect:	The FD is not available for use or the FD idle stack is corrupt. If enough FDs became unavailable, the QLPS network would lose capacity.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Head of the FD idle stack (CMbufidle.toppt).
<i>Bad Data =</i>	FD's user bits (CMBUFUSE.bf_owner).
<i>Good Data =</i>	FD's usage type (CMBUFUSE.usetype).
<i>Logical Index =</i>	FD's index (index into CMfduse[]).
<i>Dump =</i>	FD (entry in CMfduse[]).

Table 5 Error Code: HEAD

Error Code:	HEAD
Error Description:	The FD (entry in CMfduse[]) is corrupt. One or more of the following are true: • CMbufidle.toppt is out-of-range (> CM_NUMBUFS). • CMbufidle.fdindx[CMbufidle.toppt] = CMNIL_NDX. • CMbufidle.fdindx[CMbufidle.toppt+1] != CMNIL_NDX.
Possible Error Effect:	FDs can not be allocated. No new frames can be transmitted. If the processor on which the audit is running is an MH, the switching module (SM) associated with the MH may lose communication with the QLPS network. If the processor the audit is running on is a QLPS gateway processor (QGP), the QLPS network may lose capacity or may lose communication to some SMs.
Corrective Action Taken:	The audit clears the FD idle stack (CMbufidle.toppt = 0) and rebuilds the idle list (links all FDs whose usetype is CMFREES to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMbufidle.toppt.
<i>Good Data =</i>	qlcib_idx QLLIXNULL.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 6 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An FD in-use idle stack entry (index <= CMbufidle.toppt) is corrupt. One or more of the following is true ("idle_fd" refers to the contents of the idle stack entry: CMbufidle.fdindx[index]): • idle_fd ≥ CM_NUMBUF • idle_fd < 0 • CMfduse[idle_fd].qindx != index into CMbufidle.fdindx[]
Possible Error Effect:	FDs may not be allocatable. If so, no new frames can be transmitted. If the processor the audit is running on is an MH, the SM associated with the MH may lose communication with the QLPS network.

	If the processor the audit is running on is a QGP, the QLPS network may lose capacity or may lose communication to some SMs.
Corrective Action Taken:	If this is the first in-use idle list entry (index = CMbufidle.toppt), the audit clears the entry (CMbufidle.fdindx[index] = CMNIL_NDX) and decrements the count of idle FDs (decrements CMbufidle.toppt). Otherwise, the audit puts a good value into the entry. That is, the audit sets CMbufidle.fdindx[indx] = CMbufidle.fdindx[CMbufidle.toppt], relinks the FD (CMfduse[CMbufidle.fdindx[CMbufidle.toppt]].qindx = index), clears the old entry (CMbufidle.fdindx[CMbufidle.toppt] = CMNIL_NDX), and decrements the number of idle FDs (decrements CMbufidle.toppt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Index into CMbufidle.fdindx[].
<i>Good Data</i> =	CM_NUMBUF.
<i>Logical Index</i> =	Contents of CMbufidle.fdindx[index] ("idle_fd").
<i>Dump</i> =	NA.

Table 7 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An FD (entry in CMfduse[]) is owned only by level 1 (CMBUFUSE.bf_owner & CM_L1_OWNER != 0, but CMBUFUSE.bf_owner and CM_L2_OWNER = 0) and has not been touched for several minutes (CMBUFUSE.au_detect = 1 and CMBUFUSE.au_recover = 1).
Possible Error Effect:	The FD is not available for use. If enough FDs became unavailable, the QLPS network would lose capacity.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Head of the FD idle stack (CMbufidle.toppt).
<i>Bad Data</i> =	FD's user bits (CMBUFUSE.bf_owner).
<i>Good Data</i> =	FD's usage type (CMBUFUSE.usetype).
<i>Logical Index</i> =	FD's index (index into CMfduse[]).
<i>Dump</i> =	FD (entry in CMfduse[]).

Table 8 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused entry (index > CMbufidle.toppt) in the FD (entry in CMfduse[]) is not null (CMbufidle.fdindx[index] != CMNIL_NDX).
Possible Error Effect:	FDs may not be allocatable. If so, no new frames can be transmitted. If the processor the audit is running on is an MH, the SM associated with the MH may lose communication with the QLPS network. If the processor the audit is running on is a QGP, the QLPS network may lose capacity or may lose communication to some SMs.
Corrective Action Taken:	The audit clears the FD idle stack (CMbufidle.toppt = 0) and rebuilds the idle list (links all FDs whose usetype is CMFREES to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Contents of CMbufidle.fdindx[index].
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into CMbufidle.fdindx[].
<i>Dump</i> =	NA.

Table 9 Error Code: QUE

Error Code:	QUE
Error Description:	An FD (entry in CMfduse[]) appears to be in either the transmit or message queue (CMBUFUSE.usetype = CMXMTQUE or CMMSGQUE) but the linkage to the queue is corrupt. Specifically: CMXMTQ = cmCOMLKDATA.fdindx != the FD's index, or the FD's queue slot can not be reached from the load and unload queue indexes. CMMSGQ = cmMSGQ.fdindx[CMBUFUSE].qindx != the FD's index, or the FD's queue slot cannot be reached from the load and unload queue indexes.
Possible Error Effect:	The FD is not available for use. If enough FDs became unavailable, the QLPS network would lose capacity.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Head of the FD idle stack (CMbufidle.toppt).
<i>Bad Data =</i>	FD's user bits (CMBUFUSE.bf_owner).
<i>Good Data =</i>	FD's usage type (CMBUFUSE.usetype).
<i>Logical Index =</i>	FD's index (index into CMfduse[]).
<i>Dump =</i>	FD (entry in CMfduse[]).

Table 10 Error Code: RCVACBLK

Error Code:	RCVACBLK
Error Description:	An FD (entry in CMfduse[]) is in use as a receive FD (CMBUFUSE.usetype = CMRCVBUF) and is owned by an unused channel data block (CMspcdb[CMBUFUSE.I1_channel].state = CMCHSTOFF), but has invalid level 1 linkages. That is, one or more of the following are true: • CMBUFUSE.I1_next = CMNIL_NDX, but CMCDBENTRY.rcv_unload != the FD's index. • CMBUFUSE.I1_prev = CMNIL_NDX, but CMCDBENTRY.rcv_load != the FD's index. • CMBUFUSE.I1_next != CMNIL_NDX, but is not valid (is ≥ CM_NUMBUF). • CMBUFUSE.I1_prev != CMNIL_NDX, but is not valid (is ≥ CM_NUMBUF). • CMBUFUSE.I1_next is < CM_NUMBUF, but CMfduse[CMBUFUSE.I1_next].I1_prev != the FD's index. • CMBUFUSE.I1_prev is < CM_NUMBUF, but CMfduse[CMBUFUSE.I1_prev].I1_next != the FD's index.
Possible Error Effect:	The FD is not available for use. If enough FDs became unavailable, the QLPS network would lose capacity.
Corrective Action Taken:	The audit unlinks the FD from any other data structures, sets all fields to default values and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Head of the FD idle stack (CMbufidle.toppt).
<i>Bad Data =</i>	FD's user bits (CMBUFUSE.bf_owner).
<i>Good Data =</i>	FD's usage type (CMBUFUSE.usetype).
<i>Logical Index =</i>	FD's index (index into CMfduse[]).
<i>Dump 1 =</i>	FD (entry in CMfduse[]).

	Up to two of the following dumps will also be provided in the order listed depending on the values of the FD linkage fields:
<i>Dump 2</i> =	FD indexed by CMBUFUSE.next_ifd if next_ifd is < CM_NUMBUF.
<i>Dump 3</i> =	FD indexed by CMBUFUSE.back_ifd if back_ifd is < CM_NUMBUF.
<i>Dump 4</i> =	The first 64 bytes of CMmsgbuff[FD index].

Table 11 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: - clears (set = CMNIL_NDX) the entry (if any) in CMbufidle.fdindx[] which was being audited when the stimulus occurred. - recovers the FD, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the FD from any other data structures, sets all fields to default values, and puts the FD on the FD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMmhrunning (the current base-level loop job).
<i>Bad Data</i> =	Current entry in CMbufidle.fdindx[].
<i>Good Data</i> =	Current index into CMbufidle.fdindx[].
<i>Logical Index</i> =	Current FD index (index into CMfduse[]).
<i>Dump</i> =	FD (entry in CMfduse[]).

FMGR

Software Release: 5E10 and later

1. FILE MANAGERS (FMGR) AUDIT

The file manager (FMGR) audits verify the data in the file manager task table and other internal buffers. Nine correcting, frequency-group audits make up the FMGR family.

- FMGR1 - Task queue audit. The FMGR task table audit checks the validity of certain fields that describe an active task. If more than 10 tasks are active, it looks for active tasks that have been road blocked for more than 7 minutes or less. When a road-blocked task is found, it is torn down, its resources are freed, and an acknowledgment is sent to the requester. Besides being scheduled routinely with the other FMGR audits, this audit is requested by the file manager when its task table is in or near overload.
- FMGR2 - Internal inode table audit. The FMGR inode audit verifies the integrity of the incore inode table by checking certain fields that track the current state of the inode (such as the status, use count, and time stamp fields). FMGR 2 compares the use count field against the number of opens on a file and then checks the count field's status. FMGR 2 also checks the time stamp field and, if it indicates that the disk inode has not been updated with the past three hours, FMGR compares the disk inode with the incore inode.
- FMGR3 - Hash table and internal inode pointer table audit. The FMGR inode hash table audit verifies the integrity of the inode hash table and the inode free chain. The inode hash table chains inodes with the same hash value together to speed up the search for an inode. FMGR validates these chains and ensures that each incore inode is either on the free chain or on one of the hash chains.
- FMGR4 - File table audit. The FMGR file table audit verifies the integrity of the file manager's internal file table. FMGR 4 validates the fields that link a file table entry to an incore inode, to other file table entries, and to capability table entries. (A file table entry is made for each open on a file.) FMGR 4 also compares its count field with the number of capability table entries that are linked to the count field.
- FMGR5 - Internal capability table audit. The FMGR capability table audit validates the capability table. A major purpose of this audit is to detect and remove capabilities that belong to dead processes. FMGR 5 also checks the integrity of the chain of capabilities that are linked to a file table entry.
- FMGR6 - Buffer audit. The FMGR buffer audit validates the management of the internal buffers that buffer input/output (I/O) to or from the file manager's data or a requester's address space. The internal buffers are linked together into devices hash chains to facilitate the search and the free buffers are also linked into the free chain. FMGR 6 validates these linked chains and the buffer status. Additionally, FMGR 6 checks the validity of locks on the file system super blocks.
- FMGR7 - Mount table audit. The FMGR mount table audit validates the inode pointers and buffer pointers in the mount table.
- FMGR8 - Delayed queue audit. The FMGR delayed queue audit ensures that the open and mount delayed queues are consistent with the current number of active, open, and mount tasks.
- FMGR9 - Message queue audit. THE FMGR delayed queue audit checks for message buffer overload in the file manager's message queue.

The file manager audits are routinely scheduled in the correcting mode. Any of these audits may be manually requested, particularly if routine audit scheduling has been disabled and there is reason to suspect that a resource

has been disabled.

Any errors that occur are corrected automatically by each audit.

CAUTION: FMGR1 and FMGR5 are recovery audits and should not be inhibited without consulting your next level of technical support.

1.1 Requesting FMGR Audits

To request an audit of the FMGR resources, use the following input message:

```
AUD:FMGR=a[:b|:c|:b,c];
```

where:

a = Specifies the member number of the FMGR audit. Valid values are:

- 1 = Task queue audit.
- 2 = Internal inode table audit.
- 3 = Hash table and internal inode pointer table audit.
- 4 = File table audit.
- 5 = Internal capability table audit.
- 6 = Buffer audit.
- 7 = Mount table audit.
- 8 = Delayed queue audit.
- 9 = Message queue audit.

b = Indicates whether audit should correct errors. Valid values are:

- CORR = Specifies that an audit should correct errors (default).
- NCG = Specifies that an audit does not correct errors.

c = Indicates what type of report or messages should be generated. Valid values are:

- DETL = Generates a raw data report for each error found plus an audit completion message (default). (A limited number of raw data reports are generated.)
- SUM = Generates only an audit completion message when the audit terminates. Raw data reports are not generated for any errors that are found.

To interpret system responses to the AUD:FMGR input messages, see the APP:AUD Appendix in the Appendixes section of Input Messages manual (235-600-700).

1.2 Interpreting FMGR Output Messages and Error Reports

The AUD FMGR output messages report the termination status of the FMGR audits and/or any errors found while executing the audit. For detailed explanations of the FMGR output messages, see the Output Messages manual (235-600-750).

Table 1 lists the FMGR audit error messages and their associated categories. Error categories include the following:

- Category 0 = Errors that are easily corrected (for example, resetting a bit or individual value). There is no recovery action when the maximum is exceeded.
- Category 1 = Errors that involve multiple structures of the same audited resource (for example, rebuilding the inode pointer chain). The audit aborts when the maximum is exceeded.
- Category 2 = Errors that require searches of structures other than the type being audited and usually require restoring part of the audited resource, category 1 (for example, rebuilding the file pointers of an inode table entry). There is no recovery action when the maximum is exceeded.
- Category 3 = Errors that require input/output (for example, rereading the superblock of a mounted file system). The audit aborts when the maximum is exceeded.
- Category 4 = Errors that require a total rebuild of a structure (for example, restoring a mount table entry). The recovery action when the maximum is exceeded - phase level 1.
- Category 5 = Errors that require a task tear down (for example, a task that has timed out). The recovery action when the maximum is exceeded - phase level 1.

Table 1 FMGR AUDIT ERRORS

Category	Error Number	Meaning
5	2	FMGR 1 - Task queue audit
0	3	Task header - invalid task ID.
5	4	Task header - invalid sleep address.
5	5	Task header - invalid task stack address.
5	6	Task header - invalid message address.
0	7	Task header - task time-out.
		Task header - invalid time stamp.
2	12	FMGR 2 - Inode audit
0	13	Inode - bad i_count value.
0	14	Inode - bad i_wcount value.
15	0	Inode - invalid task ID.
3	16	Inode - invalid time stamp.
2	17	Inode - disk/incore mismatch.
2	19	Inode - invalid file table pointer.
2	25	Inode - invalid mount pointer.
0	26	Inode - incorrect inuse count.
0	27	Invalid inode table size.
0	28	Invalid degrowth page marker.
0	28	Inode - incorrect mount table pointer.
0	92	Inode - invalid open count for driver.
2	21	FMGR 3 - Inode hash table audit.
4	22	Inode hash - bad value.
4	24	Inode pointer - bad chain.
		Inode pointer - invalid chain pointer.
0	31	FMGR 4 - File table audit.
2	32	File - invalid FLOCK flag.
2	33	File - invalid inode pointer.
2	34	File - invalid capability pointer.
0	35	File - invalid file pointer.
2	36	File - invalid task ID.
0	39	File - bad use count of open files.
2	40	File - incorrect count of file table entries in use.
0	90	File - invalid free list.
		File - invalid file table size.

0	91	File - invalid degrowth page marker.
1	41	FMGR 5 - Capability table audit.
2	42	Capability - nonexistent process.
2	43	Capability - invalid file pointer.
2	44	Capability - invalid capability pointer.
2	45	Capability - invalid capability pointer in file table entry.
0	46	Capability - invalid capability chain.
2	47	Capability - incorrect count of capability table entries in use.
0	48	Capability - invalid free list.
0	49	Capability - incorrect capability table size.
0	49	Capability - incorrect degrowth page marker.
4	51	FMGR 6 - Buffer audit.
4	52	Buffer - bad available list.
0	54	Buffer - bad device chain.
0	55	Buffer - invalid available list pointer.
0	56	Buffer - invalid device pointer.
2	57	Buffer - invalid buffer address.
0	58	Buffer - invalid mount pointer.
5	59	Buffer - invalid task ID.
0	60	Buffer - input/output time-out.
0	61	Buffer - bad time stamp.
0	62	Buffer - bad superblock ilist lock.
0	62	Buffer - bad superblock free lock.
0	64	Buffer - invalid total number of free inodes.
0,2	72	FMGR 7 - Mount table audit.
0	74	Mount - invalid superblock buffer pointer.
2	75	Mount - invalid mount inode pointer.
0	77	Mount - invalid root inode pointer.
2	82	Mount - invalid task ID.
2	83	FMGR 8 - Delayed queue audit.
		Delayed queue - bad number, open.
		Delayed queue - bad number, mount.
5	81	FMGR 9 - Message queue audit.
		Overload - large message queue.

1.3 Responding to AUD FMGR Error Reports

If any FMGR audit reports an error, take the appropriate action suggested. Use the correcting (default) mode on all manual requests. If errors are still present after taking the indicated action, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

Audit Reporting Error	Action
FMGR 1	Rerun audit.
FMGR 2	Rerun FMGR 2, then run FMGR 3.
FMGR 3	Rerun FMGR 3, then run FMGR 2.
FMGR 4	Rerun FMGR 4, then run FMGR 5.
FMGR 5	Rerun FMGR 5, then run FMGR 4.
FMGR 6	Rerun audit.
FMGR 7	Rerun audit.
FMGR 8	Rerun audit.
FMGR 9	Run FMGR 1.

FSBLK

Software Release: 5E10 and later

1. FILE SYSTEM BLOCK AUDIT (FSBLK)

The file system block (FSBLK) audit verifies the integrity consistency of the free block list, the block usage bit map, and the inodes of the file system specified by the instance name.

It detects out-of-range block assignments in inodes and duplicate ownership of blocks by two or more inodes. When run in the correcting mode, it rebuilds the bitmap, free block list, and free inode list.

FSBLK also detects bad blocks and duplicate blocks and verifies the size and mode of each file in a file system. It is one of two dynamic file system audits (refer to the FSLINK audit). FSBLK is a complete file system audit and can run on mounted or unmounted file systems.

Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual (235-600-700) for a list of the file systems that the *UNIX*® RTR operating system supplies ECD audit-instance records for.

The FSBLK audit for these file systems should be scheduled as a timed audit to be run at a specified time during the day and week when the application load is minimal.

Since the correction of errors found may alter a file in a way the application did not intend, it is suggested that the audit be scheduled to run in non-correcting mode. The audit may then be run in the correcting mode after the identity of the affected files is determined. To run the FSBLK audit on another file system, the application must add an audit-instance record for that file system.

The audit may use considerable processing time, especially for a large system or on a file system with a lot of current activity.

The FSBLK audit may be run on mounted or unmounted file systems.

NOTE: Any instance of this audit will be blocked during a field update or a program update involving the file system named in the instance. Similarly, a field or program update may not be started while the FSBLK audit of a file system affected by the update is running.

When you have reason to suspect internal damage of a file system, you may manually request FSBLK 1. Run the audit first in the non-correcting mode. If errors are reported and they are correctable, you can then rerun the audit in the correcting mode.

1.1 Requesting the FSBLK Audit

To request an FSBLK audit, use the following input message:

```
AUD:FSBLK=a,INS="b"[:c|:d|:c,d][PARAM="e"];
```

where:

a = Member number to be audited.

NOTE: The value input for a is dependent upon the instance. Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual (235-600-700).

b = Instance to be audited. Instance names are the special device files corresponding to the file system audited. Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual (235-600-700) for a complete list of device file names.

- c = Audit action. Valid values are:
- CORR = Audit should correct errors.
- NCG = Audit should not correct errors (default).
- d = Type of report generated. Valid values are:
- DETL = Generates a detailed raw data report for each error found followed by an audit completion message (default). (A limited number of reports is generated.)
- SUM = Generates only a summary audit completion message when the audit terminates. Raw data reports are not created for any errors that are found.
- e = Backup parameter. This field is used to check the file system for the appropriateness of generating a backup tape. When specified, the audit may report error conditions that, normally, would not be considered errors, but could result in the creation of an improper backup tape. These errors are not correctable.

To interpret system responses to the AUD:FSBLK input message, refer to the APP:AUD Appendix in the Appendix section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD FSBLK Output Messages and Error Reports

The FSBLK audit prints an output message after each audit. It reports the termination status of the FSBLK audit and/or any errors found while executing the audit. For more detailed information about the AUD:FSBLK output message, see the Output Messages manual (235-600-750). Table 1 is a list of error messages.

The FSBLK audit uses the following error categories:

- Category 1 - Errors generally indicate noncorrectable system errors apart from the audit. The errors will cause the audit to terminate prematurely. No recover action is required.
- Category 3 - Errors indicate problems with an individual file in the file system being audited. File system corruption has occurred.

Table 1 FSBLK AUDIT ERRORS

Category	Error Number	Meaning
1	1	Bad superblock label.
3	2	Bad file mode.
3	3	Bad file size.
3	4	Bad block. The block is out of the file system block range.
3	5	Duplicate block number.
3	6	Bad address blocks. The address block contains invalid block addresses.
2	7	Temporary file found in file system when audit invoked with PARAM="BACKUP". This condition does not represent an actual file system error, but if a backup tape is made with this condition present, the backup tape could contain file system errors. There is no corrective action possible or needed for this condition.
1	8	Audit library initialization failed.
1	9	Unable to open device file.
1	10	Unable to start device file.
1	11	Device file not block type.
1	12	Unable to allocate message.
1	13	Unable to queue message.
1	14	Unable to add segments.

1	15	Unable to read superblock.
1	16	Unable to access block or get buffer to read block.
1	17	Unable to get free block.
1	18	Unable to access inode.
1	19	Unable to remove a segment.
1	20	Cannot mount device.
1	21	Unable to get bitmap segment ID.
1	22	Bitmap I/O failed.
1	26	Device file is open for writing.
1	35	File manager segment list is full.

1.3 Responding to AUD FSBLK Error Reports

Even in the correcting mode, FSBLK does not correct all the errors it finds. Errors that are not corrected and the action that should be taken are listed in Table 2.

Table 2 SPECIAL ERROR MESSAGES

Error Number	Action to be Taken
1-2	Error is not correctable. Back up the file system from the backup source.
3-6	Use the input message OP:FNAME [refer to the OP:FNAME input message in the Input Messages manual (235-600-700)] to print the full file pathname of the damaged file. To correct the error, rerun the audit in the correcting mode.
7	No action is required.
8, 12-22	An error in accessing the disk or in getting system resources has occurred. Check the disk status and/or system resource usage. Rerun the FSBLK audit when the system is quiet and disks are active.
9-11	The divide file for the file system does not exist or it is the wrong type. Check and repair/create the device file that defines the file system.
26	The device file is open for writing. Try again later.
35	This will occur when the maximum number of file manger tasks are running and using the available file manager segment table slots. Rerun the audit at different times during the day when there are no other file system audits running. If the problem persists, execute a level 3 initialization. If the problem still persists, then create an off-line memory dump tape and refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

FSCMPT

Software Release: 5E10 and later

1. FILE SYSTEM COMPACTION (FSCMPT) AUDIT

The file system compaction (FSCMPT) audit provides a facility for merging free disk blocks of a file system to form a large free contiguous space. This will alleviate the problem of fragmentation of free disk space in a file system that interferes with the allocation of space for contiguous files. The facility to compact a file system will be executed as a file system correcting audit that can be invoked by the user.

An extent is a set of contiguous blocks within a file system. A file system becomes fragmented when the total number of free blocks available for extent files is larger than the size of an allocation request, but the size of each free extent is less than the required amount.

After the *UNIX*[®] RTR operating system has been updated several times, the free blocks of the file system become distributed into small contiguous sets separated by used blocks. From this fragmented state, the compaction audit will create a large set of contiguous free blocks by merging some of the free blocks together.

To compact the file system, the compaction audit first surveys the file system to determine the following:

- The total number of free blocks in the file system
- The portions of used blocks allocated to regular and contiguous/extent files
- The size of the largest extent file in the file system.

This information together with the *max/min* parameter is used to set a compaction goal (the number of free blocks that it will attempt to merge into a single free extent). This number will be less than the total number of free blocks.

The number and distribution of *regular* file blocks has a large impact on the compaction overhead and may further limit the compaction goal. Because of the cost in system time and resources, the audit tries to avoid moving *regular* files, preferring to move contiguous/extent files instead. The *cond* parameter restricts the number of regular file blocks that will be moved to 511, the size of the free block chain from which blocks for regular files are taken. If the *ucl* parameter is specified, the number of regular file blocks that will be moved is limited only by availability of relocation space.

Each file system on which the compaction audit will be run should be carefully reviewed and thoroughly tested under appropriate load conditions. Both the outcome and performance of the file system compaction audit is dependent on the application environment.

The following rules apply when compacting a file system:

- The audit should be run during periods of light system activity.
- The amount of activity occurring in the file system undergoing compaction should be kept to a minimum if possible.
- The audit should **not** be run as a routine audit.

The compaction audit may be run in a noncorrecting mode or in a correcting mode. If the compaction audit is run in noncorrecting mode, the compaction goal that has been determined is reported with an error code 2. The information in the report includes the following:

DATA1 = Total number of free blocks in the file system.

- DATA2 = Number of contiguous free blocks to be created by compaction.
- DATA3 = Current largest number of contiguous free blocks.
- DATA4 = A projection of the number of I/O jobs required to do the compaction.

Each of the DATA items is reported in hexadecimal notation.

In the correcting mode, after mapping the file system and setting the compaction goal, used blocks are relocated in order to merge the free blocks into a large free extent. At the conclusion of the compaction, an error code 3 report contains the following information.

- DATA1 = total number of free blocks in the file system.
- DATA2 = number of contiguous free blocks created by the compaction effort.
- DATA3 = largest number of contiguous free blocks before compaction.
- DATA4 = the number of I/O jobs involved in doing the compaction.

All error codes except 2 and 3 report error conditions that prevent the audit from completing. For the interpretation and recommended action for each error code, refer to the AUD:FSCMPT output message page in the Output Message Manual (235-600-750).

1.1 When to Use the Compaction Audit

If an attempt to allocate a number of blocks for a contiguous or extent file fails, that is, the number allocated is less than the amount requested, the following procedures should be used.

- (1) Using one of the following two methods, determine whether there are sufficient free blocks in the file system that contains the file.

- At the *UNIX*® system terminal, enter:

```
df<filesystem>
```

- At the user terminal, enter:

```
OP:STATUS,FREEDISK,FN="a";
```

Where *a* is the full pathname to the file system.

If the number of free blocks is not smaller than the desired amount, continue to the next step. Otherwise, some files may need to be removed from the file system to make more free space.

- (2) Run the FSBLK audit in noncorrecting mode, by entering:

```
AUD:FSBLK=1,INS="<filesystem>":NCG;
```

If no errors are reported, proceed to the next step. If errors are reported, refer to the Output Message Manual (235-600-750) for interpretation and recommended action to be taken.

- (3) Run the FSBLK audit in correcting mode by entering:

```
AUD:FSBLK=1,INS="<filesystem>":CORR;
```

This causes all free blocks to be returned to the pool of free blocks available for both regular and extent files. If the required number of contiguous blocks still cannot be allocated, the file system compaction audit may be able to merge enough free blocks for the allocation.

- (4) Perform the steps listed in the How to Use the Compaction Audit section.

1.2 How to Use the Compaction Audit

The input message that runs the file system compaction audit can be entered manually on the maintenance terminal or craft shell terminal.

- (1) Run the compaction audit in noncorrecting (NCG) mode with increasingly liberal parameters as listed below:

```
AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MIN,COND";
AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MAX,COND";
AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MIN,UCL";
AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MAX,UCL";
```

The projected number of contiguous free blocks that the compaction audit will be able to create when run in correcting mode is reported in DATA 2 of the error code 2 report. For most file systems, this projected number will increase as you progress through the list. When DATA2 in the output message is large enough, proceed to the next step.

- (2) Run the audit in correcting (CORR) mode with the pair of parameters selected. An error 3 reports the compaction completed with DATA 2 specifying the number of contiguous free blocks now available in the file system. This may be less than the amount projected in step 1. In this case, you may rerun the audit in correcting mode with the same parameters or return to step 1, using the next pair of parameters in the list.

The file system compaction audit can run on mounted or unmounted file systems.

1.3 Requesting the FSCMPT Audit

When running the FSCMPT audit, it should first be run in the noncorrecting mode to determine the results and cost of compacting the file system. Then it should be run in the correcting mode with the selected parameters to compact the file system. To request an FSCMPT audit, use the following input message:

```
AUD:FSCMPT=a,INS="b" [ ; c ; d | ; c, d ] [ : PARAM="{ e | f | e, f } " ] ;
```

where:

- a = Member number of the FSCMPT audit. The administrative module (AM) audit is member number 1.
- b = The instance to be audited. Instance names are the special devices files corresponding to the file system audited. Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual (235-600-700) for specific device file names.

To run the FSCMPT audit on another file system, the application must add an audit instance record for the file system.

- c = Correcting or not correcting. Valid values are:

CORR = Audit should correct errors and complete the compaction.

NCG = Audit should not correct errors and should not do the compaction (default).

d = Type of report generated. Valid values are:

DETL = Generates both a raw data report for each error found and an audit-completion message (default). Only a limited number of raw data reports are generated.

SUM = Generates only an audit-completion message when the audit terminates. Raw data reports are not generated for any errors found.

e = Type of effort compaction. Valid values are:

MIN = Perform or report for a minimum effort compaction (default).

MAX = Perform or report for a maximum effort compaction.

f = Number of file blocks relocated. Valid values are:

COND = Limit number of regular file blocks relocated to achieve compaction (default).

UCL = An unlimited number of regular file blocks may be relocated to achieve compaction.

1.4 Interpreting AUD FSCMPT Output Messages and Error Reports

The AUD FSCMPT 1 output messages report the termination status of the FSCMPT audit and provide information about errors that were found while the audit was executing.

For more detailed information about the AUD:FSCMPT output message, refer to the Output Messages manual (235-600-750). Table 1 is a list of error messages.

The FSCMPT audit uses three error categories. These categories are defined as follows.

Category 1 = Errors generally indicate noncorrectable system errors apart from the audit. These errors will cause the audit to terminate prematurely. No recovery action is required.

Category 3 = Errors indicate problems with an individual file in the file system being audited. File system corruption has occurred.

Category 5 = Errors are informational only, and are not actual errors.

Table 1 FSCMPT AUDIT ERRORS

Category	Error Number	Meaning
1	1	Bad superblock label.
5	2	Not an error; contains information. Refer to How to Use the Compaction Audit.
5	3	Not an error; contains information. Refer to How to Use the Compaction Audit.
3	4	Bad block. The block is out of the file system block range.
3	5	Duplicate block number.
3	6	Bad address block.
1	8	Audit library initialization failed.
1	9	Unable to open device file.
1	10	Unable to start device file.
1	11	Device file not a block device.

1	12	Unable to allocate a message.
1	13	Unable to queue a message.
1	14	Unable to add segments.
1	16	Unable to access block (or get buffer to read block).
1	17	Unable to get a free block.
1	18	Unable to access inode.
1	19	Unable to remove a segment.
1	20	Cannot mount the device.
1	21	Unable to get bitmap segment ID.
1	22	Bitmap I/O failed.
1	23	No available structure for mapping allocated blocks for the compaction.
1	24	Concurrent file system activity is interfering with the compaction effort.
1	25	I/O attempt failed.
1	26	The device file is open for writing.
1	35	Partition for the file system on which the device file resides.

1.5 Responding to AUD FSCMPT Error Reports

Even in the correcting mode, FSCMPT does not correct all the errors it finds. Errors that are not corrected and the action that should be taken are listed in Table 2 .

Table 2 SPECIAL ERROR MESSAGES

Error Number	Action to be Taken
1	Error may not be corrected. Back up the file system from backup source.
4-6	For block errors run the block audit in the noncorrecting mode. If it finds an error, run the block audit in the correcting mode.
8-23, 25, 26	Error may not be corrected; refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
24	Run this audit with reduced file system activity.
35	This will occur when the maximum number of file manger tasks are running and using the available file manager segment table slots. Rerun the audit at different times during the day when there are no other file system audits running. If the problem persists, execute a level 3 initialization. If the problem still persists, then create an off-line memory dump tape and refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

FSLINK

Software Release: 5E10 and later

1. FILE SYSTEM LINK (FSLINK) AUDIT

The file system link (FSLINK) audit verifies the linkage among the directories and inodes of the file system specified by the instance name.

It detects a disagreement between the link count in an inode and the number of references to the inode in the directories of the file system. In the correcting mode, it corrects the inode link count that does not agree with the number of references, invalidates a directory entry with an invalid inode number, and moves an inode that has no directory references to a *lost&found* directory.

Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual (235-600-700) for a list of the file systems that the *UNIX*® RTR operating system supplies ECD audit-instance records for.

It is recommended that the FSLINK audit for these file systems be scheduled as a timed audit, to be run at specified times during the day and week, at a time when the application load is minimal.

Since the correction of errors found may alter a file in a way the application may not intend, it is suggested that the audit be scheduled to run in the non-correcting mode. The audit may then be run in the correcting mode after the identity of the affected files is determined. To run the FSLINK audit on another file system, the application must add an audit-instance record file for that file system.

This audit may use considerable processing time especially for a large system, or on a file system with a lot of current activity.

The FSLINK audit can be run on a mounted or unmounted file system.

1.1 Requesting the FSLINK Audit

When you have reason to suspect internal damage of a file system, you may manually request an FSLINK audit. Run the audit first in the non-correcting mode. If errors are reported, and they are correctable, you can then rerun the audit in the correcting mode. To request an FSLINK audit, use the following input message:

```
AUD:FSLINK=a,INS="b"[:c|:d|:c,d][:PARAM="e"];
```

Where:

a = Member number to be audited.

NOTE: The value input for a is dependent upon the instance. Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual, 235-600-700.

b = Instance to be audited. Instance names are the special device files corresponding to the file system audited. Refer to the APP:AUD-SDF appendix in the Appendixes section of the Input Messages manual, 235-600-700, for a complete list of special device file names.

c = Audit action. Valid value(s):

CORR = Audit should correct errors.

NCG = Audit does not correct errors (default).

d = Type of report. Valid value(s):

DETL = Generate a raw data report for each error found followed by an audit completion message (default). (A limited number of reports is generated.)

SUM = Generate only an audit completion message when the audit terminates. Raw data reports are not created for any errors that are found.

e = BACKUP parameter. This field is used to check the file system for the appropriateness of generating a backup tape. When specified, the audit may report error conditions that, normally, would not be considered errors, but could result in the creation of an improper backup tape. These errors are not correctable.

Refer to the APP:AUD appendix in the Appendixes section of the Input Messages manual, 235-600-700, to interpret system responses to the AUD:FSLINK input message.

1.2 Interpreting AUD:FSLINK Output Message and Error Reports

The FSLINK audit prints an output message after each audit. The output message reports the termination status of the FSLINK audit and provides information about errors that were found while the audit was executing. For more detailed information about the AUD:FSLINK output message, refer to the 5ESS[®] Switch Output Messages manual, 235-600-750.

The FSLINK audit uses two error categories.

Category 1 = Errors generally indicate noncorrectable system errors apart from the audit. These errors will cause the audit to terminate prematurely. No recovery action is required.

Category 2 = Errors indicate problems with the file system directory structure. File system corruption has occurred.

Refer to Table 1.

Table 1 FSLINK Audit Errors

Category	Error Number	Meaning
1	1	Bad super block label.
2	2	Mutilated inode in the inode table.
2	3	Bad directory entry.
2	4	Bad directory block (block out of file system block range).
2	5	Bad link count (link count in inode not same as number of references in directories).
2	6	Unreferenced inode.
2	7	Link count of the director inode is less than 2.
1	8	Audit library initialization failed.
1	9	Unable to open device file.
1	10	Unable to stat device file.
1	11	Device file not block device.
1	12	Unable to allocate a message.
1	13	Unable to queue a message.
1	14	Unable to add segments.
1	16	Unable to access block (or get buffer to read block).
1	18	Unable to access inode.
1	19	Unable to remove a segment.
1	20	Cannot mount the device.
1	26	The device file is open for writing.
2	27	Current directory entry could not be referenced from the parent directory.
2	28	Current directory entry cannot reference its parent directory.
2	29	Current directory entry has a bad inode.
2	30	References to a free inode.

2	31	Current directory has more than one parent directory.
2	32	Current directory is not reachable from file system root.
1	33	Active but unreferenced file in file system when audit invoked with PARAM="BACKUP".
2	35	File manager segment list is full.

1.3 Responding to AUD:FSLINK Error Reports

Even in the correcting mode, FSLINK may not correct all the errors it finds. Another process may be using the inode, so that the audit may not need to be run more than once to correct the errors. Errors that are detected and/or corrected and the action that should be take are listed in Table 2

Table 2 Special Error Messages

Error Number	Action to be Taken
1, 2, 4, 7	These error codes are generally not correctable. Before backing up the file system from the backup source, rerun the audit in the correcting (CORR) mode; this may correct some of the errors.
3, 5, 27 - 32	Use the input message OP:FNAME to print the full path name of file. Refer to the Input Messages manual, 235-600-700. To correct errors, rerun the audit in the CORR mode.
6	The unreferenced file cannot be reached in the file system by any name. To correct errors, rerun the audit in the CORR mode.
8, 12 - 14, 16, 18 - 20	Check disk status and/or system resource usage. Rerun audit when the system is quiet and disks active. Disk-access error or system-resource-access error has occurred.
9 - 11	Check and repair or create a device file that defines the file system. Device file is non-existent or the wrong type.
15, 17	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
26	Rerun the audit later. If error persists, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
33	This condition does not represent an actual file system error, but if a backup tape is made with this condition present, the backup tape could contain file system errors. There is no corrective action for this condition.
35	This will occur when the maximum number of file manager tasks are running and using all available file manager segment table slots. Rerun the audit at different times during the day when there are no other file system audits running. If the problem persists, execute a level 3 initialization. If the problem still persists, then create an off-line memory dump tape and refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

GCKTI

Software Release: 5E10 and later

1. GLOBAL SERVICE CIRCUIT (GCKTI) AUDIT

Conference circuits are used for conference calling and for verifying that busy lines are actually busy. The conference circuits are part of the global digital service unit (GDSU) and are resources that can be accessed from any switching module (SM) in an office. Each office has a specified number of conference circuits; therefore, it is important that these resources be operational.

The peripheral control subsystem uses the static (write-protected) global conference circuit relation (RLGCKTI) to initialize the conference circuit access table dynamic relation (RLCCATBL) and the global conference circuit access parameters (PCccap and PC6ccap). The global service circuit audit (GCKTI) performs static-to-dynamic existence checks between RLGCKTI and RLCCATBL, and verifies the index and number of circuits listed in PCccap and PC6ccap.

This audit is part of the peripheral control audits subgroup. It runs in the communication module processor (CMP) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The GCKTI audit verifies the dynamic conference circuit access table relation (RLCCATBL) against the static global conference circuit relation (RLGCKTI). The RLGCKTI relation contains information about all conference circuits, such as the switching module number (im), the remote switching module number (rsm), the GDSU service group key (sg), the service group number (sgn), the digital service unit number (dsu), the number of circuits (nckts), and an index to the conference circuit access table (index). The RLCCATBL relation should contain the same information.

The PCccap and PC6ccap structures contain the maximum index to RLCCATBL for any global digital services unit service group 0, and the maximum number of three- or six-port circuits in any global digital services unit service group. (Refer to Figure 1.)

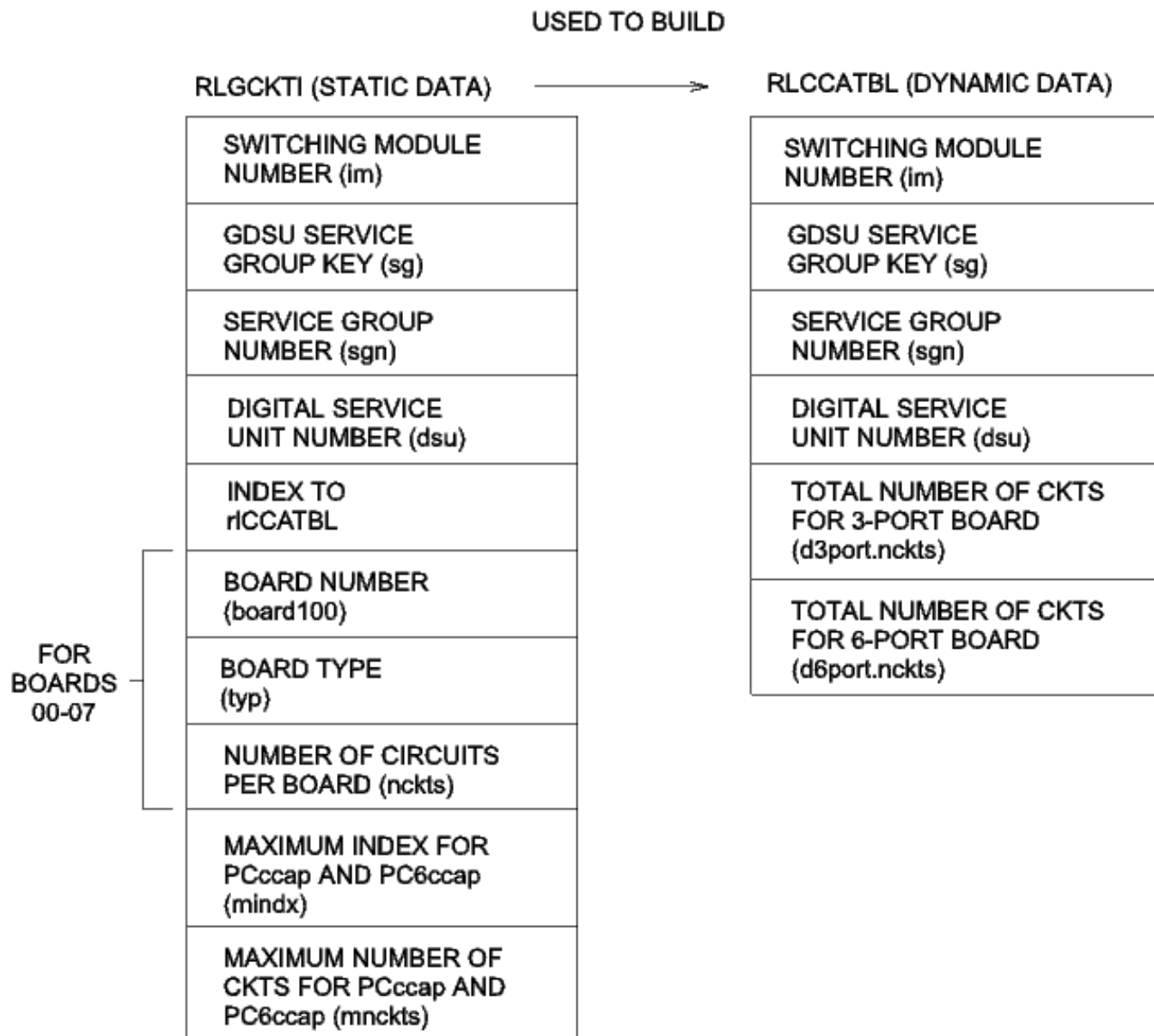


Figure 1 (GCKTI-1) Data Structures

3. AUDIT CHECKS

The global service circuit audit uses the static driver (DRSTATIC) to step through the tuples in rLGCKTI. The DRSTATIC driver calls a semantic check function to perform the following checks:

- DSUGCKT = Check the digital service unit (dsu) in the rlCCATBLE tuple against rLGCKTI.
- DYN_STAT = Check the key (ccadx) of the rlCCATBL tuple against rLGCKTI.
- IMGCKT = Check the module number (im) in the rlCCATBL tuple against rLGCKTI.
- NCKTS3P = Check the number of 3-port conference circuits (d3port.nckts) in the rlCCATBLE tuple against rLGCKTI.

- NCKTS6P = Check the number of 6-port conference circuits (d6port.nckts) in the rICCATBLE tuple against rIGCKTI.
- RSMBIT = Check the remote switch module (RSM) bit (rsmbit) in the rICCATBL tuple against rIGCKTI.
- SGGCKT = Check the service group (sg) in the rICCATBL tuple against rIGCKTI.
- SNGCKT = Check the service group number (sgn) in the rICCATBL tuple against rIGCKTI.

If the audit is about to complete the following checks are made:

- CKTCNT = Check the maximum number of circuits (mnckts) in the PCccap and PC6ccap arrays against rIGCKTI.
- MINDX = Check the minimum index (mindx) in the PCccap and PC6ccap arrays against rIGCKTI.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CKTCNT

Error Code:	CKTCNT
Error Description:	If the logical index is 0 then the maximum number of 3-port conference circuits (mnckts) in PCccap is less than the number of circuits counted in the rIGCKTI tuple. If the logical index is 1 then the maximum number of 6-port conference circuits (mnckts) in PC6ccap is less than the number of circuits counted in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls due to skip flags being set when not necessary.
Corrective Action Taken:	Set PCccap.mnckts or PC6ccap.mnckts to the number of circuits counted in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCccap (logical index = 0) or PC6ccap (logical index = 1).
<i>Bad Data =</i>	Maximum number of circuits found in PCccap or PC6ccap.
<i>Good Data =</i>	Circuit count in rIGCKTI tuple.
<i>Logical Index =</i>	0 indicates PCccap was in error 1 indicates PC6ccap was in error.
<i>Dump =</i>	NA.

Table 2 Error Code: DSUGKCT

Error Code:	DSUGKCT
Error Description:	The digital service unit field (dsu) in the rICCATBL tuple does not match the corresponding field (dsu) in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls due to calls being routed to wrong digital service unit.
Corrective Action Taken:	The digital service unit field (dsu) in the rICCATBL tuple is set to the value of the corresponding field (dsu) in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	Digital service unit field (dsu) from the rICCATBL tuple.
<i>Good Data =</i>	Digital service unit field (dsu) from the rIGCKTI tuple.
<i>Logical Index =</i>	Key of the rICCATBL tuple found in error.
<i>Dump =</i>	rICCATBL tuple found in error.

Table 3 Error Code: IMGCKT

Error Code:	IMGCKT
Error Description:	The switching module field (im) in the rICCATBL tuple does not match the corresponding field (im) in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls due to calls being routed to the wrong SM.
Corrective Action Taken:	The switching module field (im) in the rICCATBL tuple is set to the value of the corresponding field (im) in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICCATBL tuple found in error.
<i>Bad Data</i> =	Switching module field (im) from the rICCATBL tuple.
<i>Good Data</i> =	Switching module field (im) from the rIGCKTI tuple.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple found in error.

Table 4 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The key (ccadx) of the rICCATBL tuple does not match the corresponding field (index) in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls, due to garbage in the rICCATBL tuple.
Corrective Action Taken:	The key (ccadx) field of the rICCATBL tuple is set to the value of the corresponding field (index) in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICCATBL tuple found in error.
<i>Bad Data</i> =	Logical index (ccadx) of the rICCATBL tuple found in error.
<i>Good Data</i> =	Logical index (index) of the rIGCKTI tuple.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple found in error.

Table 5 Error Code: MINDX

Error Code:	MINDX
Error Description:	If the logical index is 0 then the minimum index field (mindx) in PCccap does not match the minimum index calculated from the RLGCKTI relation. If the logical index is 1 then the minimum index field (mindx) in PC6ccap does not match the minimum index calculated from the RLGCKTI relation.
Possible Error Effect:	Longer allocation time for conference circuits.
Corrective Action Taken:	Set PCccap.mindx or PC6ccap.mindx to the value calculated from the RLGCKTI relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCccap (logical index = 0) or PC6ccap (logical index = 1).
<i>Bad Data</i> =	Minimum index (mindx) value from PCccap (logical index = 0) or PC6ccap (logical index = 1).
<i>Good Data</i> =	Minimum index value obtained from RLGCKTI relation.
<i>Logical Index</i> =	Valid values are: 0 = PCccap was in error. 1 = PC6ccap was in error.
<i>Dump</i> =	NA.

Table 6 Error Code: NCKTS3P

Error Code:	NCKTS3P
Error Description:	The number of 3-port conference circuits (d3port.nckts) in the rICCATBL tuple does not agree with the number of 3-port conference circuits (boardXX.nckts) found in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls, due to garbage in the rICCATBL tuple.

Corrective Action Taken:	The number of 3-port circuits (d3port.nckts) in the rICCATBL tuple is set to the number of 3-port conference circuits (boardXX.nckts) found in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	Number of 3-port conference circuits (d3port.nckts) from the rICCATBL tuple.
<i>Good Data =</i>	Number of 3-port conference circuits (boardXX.nckts) from the rIGCKTI tuple.
<i>Logical Index =</i>	Key of the rICCATBL tuple found in error.
<i>Dump =</i>	rICCATBL tuple found in error.

Table 7 Error Code: NCKTS6P

Error Code:	NCKTS6P
Error Description:	The number of 6-port conference circuits (d6port.nckts) in the rICCATBL tuple does not agree with the number of 6-port conference circuits (boardXX.nckts) found in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls, due to garbage in the rICCATBL tuple.
Corrective Action Taken:	The number of 6-port circuits (d6port.nckts) in the rICCATBL tuple is set to the number of 6-port conference circuits (boardXX.nckts) found in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	Number of 6-port conference circuits (d6port.nckts) from the rICCATBL tuple.
<i>Good Data =</i>	Number of 6-port conference circuits (boardXX.nckts) from the rIGCKTI tuple.
<i>Logical Index =</i>	Key of the rICCATBL tuple found in error.
<i>Dump =</i>	rICCATBL tuple found in error.

Table 8 Error Code: RSMBIT

Error Code:	RSMBIT
Error Description:	The remote switching module field (rsm) in the rICCATBL tuple does not match the corresponding field (rms) in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls.
Corrective Action Taken:	The remote switching module field (rsm) in the rICCATBL tuple is set to the value of the corresponding field (rsm) in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	The value of the remote switching module field (rsm) from the rICCATBL tuple.
<i>Good Data =</i>	The value of the remote switching module field (rsm) from the rIGCKTI tuple.
<i>Logical Index =</i>	Key of the rICCATBL tuple found in error.
<i>Dump =</i>	rICCATBL tuple found in error.

Table 9 Error Code: SGGKCT

Error Code:	SGGKCT
Error Description:	The service group field (sg) in the rICCATBL tuple does not match the corresponding field (sg) in the rIGCKTI tuple.
Possible Error Effect:	Possible blocking of conference calls due to calls being routed to wrong service group.
Corrective Action Taken:	The service group field (sg) in the rICCATBL tuple is set to the value of the corresponding field (sg) in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICCATBL tuple found in error.
<i>Bad Data =</i>	The service group field (sg) from the rICCATBL tuple.
<i>Good Data =</i>	The service group field (sg) from the rIGCKTI tuple.
<i>Logical Index =</i>	Key of the rICCATBL tuple found in error.
<i>Dump =</i>	rICCATBL tuple found in error.

Table 10 Error Code: SNGKCT

Error Code:	SNGKCT
Error Description:	The service group number field (sgn) in the rICCATBL tuple does not match the corresponding field (sgn) in the rIGCKTI tuple.

Possible Error Effect:	Possible blocking of conference calls due to calls being routed to wrong service group number.
Corrective Action Taken:	The service group number field (sgn) in the rICCATBL tuple is set to the value of the corresponding field (sgn) in the rIGCKTI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICCATBL tuple found in error.
<i>Bad Data</i> =	The service group number field (sgn) from the rICCATBL tuple.
<i>Good Data</i> =	The service group number field (sgn) from the rIGCKTI tuple.
<i>Logical Index</i> =	Key of the rICCATBL tuple found in error.
<i>Dump</i> =	rICCATBL tuple found in error.

GPAXDB-A

Software Release: 5E10 - 5E13

1. GENERAL PURPOSE ANNEXED DATA BLOCK (GPAXDB) AUDIT

The RLGPAXDB relation is used by the 5E-AMPS feature to define the sub-processes (NEAR, FAR1, FAR2, FAR3) of an *AUTOPLEX*[®] (APX) call. Tuples in RLGPAXDB are linked either to an idle list from rIDB_HEADC, to an rIPCBLA tuple (when APX is using a terminal process), or to an rIAPXCTBL tuple (when APX is using the system process and two non-cell site trunks are involved).

The purpose of the GPAXDB audit is to ensure the integrity of the semantic data and linkages of the rIGPAXDB tuples in the RLGPAXDB relation. The GPAXDB audit runs in the switching module environment in segmented mode. It is also called during selective initialization.

2. DATA STRUCTURES

The RLGPAXDB dynamic relation contains information on near and far parties involved in an APX call, including the user (rIGPAXDB.client_tag) to identify whether the call is CL_APX1 (holding information for the near and first far leg), or CL_APX2 [with information for the second and third far leg(s)]. The GPAXDB audit references the following dynamic relations:

- RLDB_HEADC = One tuple of this relation acts as the idle list headcell for relation RLGPAXDB, linked through *gpaxdb_mem*.
- RLAPXCTBL = A busy rIGPAXDB tuple may be linked through *spidx* to an rIAPXCTBL tuple, and/or
- RLPCBLA = A busy rIGPAXDB tuple may be linked through *gpaxdb_mem.st_ownk* to an rIPCBLA tuple.

3. AUDIT CHECKS

For each rIGPAXDB tuple found in the relation, the audit makes the following checks:

- MEMBAD = Some of the linkage in rIGPAXDB.gpaxdb_mem is corrupt.
- NOPCBLA = If the rIGPAXDB tuple is neither idle nor APX, it must be linked to an rIPCBLA tuple.
- OWNLINK = When the rIGPAXDB tuple is idle, it must be owned by the idle list.
- WRONGLIST = Check that the user of the previous rIGPAXDB tuple is appropriate for this one.

If the rIGPAXDB tuple is busy:

- CALLID_BAD = The rIGPAXDB tuple should have a valid call record index.
- DEADPID = The tuple should be linked to an active rIPCBLA tuple.
- IDLE_LINK = If the user of the rIGPAXDB tuple is an APX process (rIGPAXDB.client_tag = CL_APX1 or CL_APX2), it must be linked to an rIPCBLA or rIAPXCTBL tuple.
- PID = When a far process is in the same SM, it must be an active process.
- UNEQUAL = When the rIGPAXDB tuple is linked to an APX system process, the key of the rIGPAXDB tuple (rIGPAXDB.gpaxdb) must match one of the semantic links in rIAPXCTBL (rIAPXCTBL.gpaxidx1 and gpaxidx2).

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CALLID_BAD

Error Code:	CALLID_BAD
Error Description:	The call record index is out of range. This field should reference a valid call record (zero to CRMAX - 1).
Possible Error Effect:	Either the tuple (and any other resources linked to it) is lost or the call it was supposed to be associated with will not be able to reliably access information about call legs.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The invalid call record index (<i>rIGPAXDB.gpaxdata.far.cid.crindx</i>).
<i>Good Data =</i>	The number of the Far party (1=far1, 2=far2, 3=far3).
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump =</i>	The rIGPAXDB tuple where the error was found.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rIGPAXDB tuple is not linked to its headcell nor to an active rIPCBLA tuple, nor is it zero. It must be linked to one of them.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The linkage to the rIPCBLA owning the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb mem.st ownk - DBSFPCBLA</i>).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump 1 =</i>	The rIGPAXDB tuple where the error was found.
<i>Dump 2 =</i>	The process control block.

Table 3 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The user of an rIGPAXDB tuple is an APX process (<i>rIGPAXDB.client_tag</i> = CL_APX1 or CL_APX2), but the tuple has no link to rIPCBLA tuple or to an rAPXCTBL tuple. An active rIGPAXDB APX tuple must be linked to a terminal or system process.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The linkage to the rIPCBLA tuple (<i>rIGPAXDB.gpaxdb mem.st ownk</i>).

<i>Good Data</i> =	The semantic link to the APX system process (<i>rIGPAXDB.gpaxdata.spidx</i>).
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 4 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The linkage test for the rIGPAXDB tuple failed.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data</i> =	The user of the rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 5 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	The user of the rIGPAXDB tuple is other than null or APX, but it is not linked to a rIPCBLA tuple.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data</i> =	The linkage to the rIPCBLA tuple (<i>rIGPAXDB.gpaxdb_mem.st_ownk</i>).
<i>Good Data</i> =	The user of the rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 6 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The user of the rIGPAXDB tuple is null, but it is not linked to the idle list.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data</i> =	The linkage to the idle list in rIDB_HEADC (<i>rIGPAXDB.gpaxdb_mem.st_ownk</i>).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 7 Error Code: PID

Error Code:	PID
Error Description:	The far party is in the same switching module as the rIGPAXDB tuple, but the process of the far party is not active.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.

Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The process number of the far party.
<i>Good Data =</i>	The number of the Far party (1=far1, 2=far2, 3=far3).
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump 1 =</i>	The rIGPAXDB tuple where the error was found.
<i>Dump 2 =</i>	The process control block.

Table 8 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The rIGPAXDB tuple is linked to an APX system process, but neither of the semantic links in the rIAPXCTBL tuple (rIAPXCTBL.gpaxidx1 or gpaxidx2) holds the key to this rIGPAXDB tuple.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The two semantic links in the rIAPXCTBL tuple to the rIGPAXDB tuples (the RLAPXCTBL <i>gpaxidx1</i> field << 16 the RLAPXCTBL <i>gpaxidx2</i> field).
<i>Good Data =</i>	The semantic link in the rIGPAXDB tuple to the rIAPXCTBL tuple (<i>rIGPAXDB.gpaxdata.spidx</i>).
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump 1 =</i>	The rIGPAXDB tuple where the error was found.
<i>Dump 2 =</i>	The rIAPXCTBL tuple where the error was found.

Table 9 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rIGPAXDB tuple has a backward link on the active list, but the user of the previous rIGPAXDB tuples is: • Not CL_NULL, if the current rIGPAXDB tuple has no user, or • CL_NULL if the current rIGPAXDB tuple has a user.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The user of the previous rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump =</i>	The rIGPAXDB tuple where the error was found.

GPAXDB-B

Software Release: 5E14 and later

1. GENERAL PURPOSE ANNEXED DATA BLOCK (GPAXDB) AUDIT

The RLGPAXDB relation is used by the 5E-AMPS feature to define the sub-processes (NEAR, FAR1, FAR2, FAR3) of an *AUTOPLEX*[®] (APX) call. Tuples in RLGPAXDB are linked either to an idle list from rIDB_HEADC, to an rIPCBLA tuple (when APX is using a terminal process), or to an rIAPXCTBL tuple (when APX is using the system process and two non-cell site trunks are involved).

The purpose of the GPAXDB audit is to ensure the integrity of the semantic data and linkages of the rIGPAXDB tuples in the RLGPAXDB relation. The GPAXDB audit runs in the switching module environment in segmented mode. It is also called during selective initialization.

2. DATA STRUCTURES

The RLGPAXDB dynamic relation contains information on near and far parties involved in an APX call, including the user (rIGPAXDB.client_tag) to identify whether the call is CL_APX1 (holding information for the near and first far leg), or CL_APX2 [with information for the second and third far leg(s)]. The GPAXDB audit references the following dynamic relations:

- RLDB_HEADC = One tuple of this relation acts as the idle list headcell for relation RLGPAXDB, linked through *gpaxdb_mem*.
- RLAPXCTBL = A busy rIGPAXDB tuple may be linked through *spidx* to an rIAPXCTBL tuple, and/or
- RLPCBLA = A busy rIGPAXDB tuple may be linked through *gpaxdb_mem.st_ownk* to an rIPCBLA tuple.

3. AUDIT CHECKS

For each rIGPAXDB tuple found in the relation, the audit makes the following checks:

- MEMBAD = Some of the linkage in rIGPAXDB.gpaxdb_mem is corrupt.
- NOPCBLA = If the rIGPAXDB tuple is neither idle nor APX, it must be linked to an rIPCBLA tuple.
- OWNLINK = When the rIGPAXDB tuple is idle, it must be owned by the idle list.
- WRONGLIST = Check that the user of the previous rIGPAXDB tuple is appropriate for this one.

If the rIGPAXDB tuple is busy:

- CALLID_BAD = The rIGPAXDB tuple should have a valid call record index.
- DEADPID = The tuple should be linked to an active rIPCBLA tuple.
- IDLE_LINK = If the user of the rIGPAXDB tuple is an APX process (rIGPAXDB.client_tag = CL_APX1 or CL_APX2), it must be linked to an rIPCBLA or rIAPXCTBL tuple.
- PID = When a far process is in the same SM, it must be an active process.
- UNEQUAL = When the rIGPAXDB tuple is linked to an APX system process, the key of the rIGPAXDB tuple (rIGPAXDB.gpaxdb) must match one of the semantic links in rIAPXCTBL (rIAPXCTBL.gpaxidx1 and gpaxidx2).

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CALLID_BAD

Error Code:	CALLID_BAD
Error Description:	The call record index is out of range. This field should reference a valid call record (zero to CRMAX - 1).
Possible Error Effect:	Either the tuple (and any other resources linked to it) is lost or the call it was supposed to be associated with will not be able to reliably access information about call legs.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The invalid call record index (<i>rIGPAXDB.gpaxdata.far.cid.crindx</i>).
<i>Good Data =</i>	The number of the Far party (1=far1, 2=far2, 3=far3).
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump =</i>	The rIGPAXDB tuple where the error was found.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rIGPAXDB tuple is not linked to its headcell nor to an active rIPCBLA tuple, nor is it zero. It must be linked to one of them.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The linkage to the rIPCBLA owning the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb mem.st ownk - DBSFPCBLA</i>).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump 1 =</i>	The rIGPAXDB tuple where the error was found.
<i>Dump 2 =</i>	The process control block.

Table 3 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The user of an rIGPAXDB tuple is an APX process (<i>rIGPAXDB.client_tag = CL_APX1 or CL_APX2</i>), but the tuple has no link to rIPCBLA tuple or to an rAPXCTBL tuple. An active rIGPAXDB APX tuple must be linked to a terminal or system process.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The linkage to the rIPCBLA tuple (<i>rIGPAXDB.gpaxdb mem.st ownk</i>).

<i>Good Data</i> =	The semantic link to the APX system process (<i>rIGPAXDB.gpaxdata.spidx</i>).
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 4 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The linkage test for the rIGPAXDB tuple failed.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data</i> =	The user of the rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 5 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	The user of the rIGPAXDB tuple is other than null or APX, but it is not linked to a rIPCBLA tuple.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data</i> =	The linkage to the rIPCBLA tuple (<i>rIGPAXDB.gpaxdb_mem.st_ownk</i>).
<i>Good Data</i> =	The user of the rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 6 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The user of the rIGPAXDB tuple is null, but it is not linked to the idle list.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data</i> =	The linkage to the idle list in rIDB_HEADC (<i>rIGPAXDB.gpaxdb_mem.st_ownk</i>).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump</i> =	The rIGPAXDB tuple where the error was found.

Table 7 Error Code: PID

Error Code:	PID
Error Description:	The far party is in the same switching module as the rIGPAXDB tuple, but the process of the far party is not active.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.

Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The process number of the far party.
<i>Good Data =</i>	The number of the Far party (1=far1, 2=far2, 3=far3).
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump 1 =</i>	The rIGPAXDB tuple where the error was found.
<i>Dump 2 =</i>	The process control block.

Table 8 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The rIGPAXDB tuple is linked to an APX system process, but neither of the semantic links in the rIAPXCTBL tuple (rIAPXCTBL.gpaxidx1 or gpaxidx2) holds the key to this rIGPAXDB tuple.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The two semantic links in the rIAPXCTBL tuple to the rIGPAXDB tuples (the RLAPXCTBL <i>gpaxidx1</i> field << 16 the RLAPXCTBL <i>gpaxidx2</i> field).
<i>Good Data =</i>	The semantic link in the rIGPAXDB tuple to the rIAPXCTBL tuple (<i>rIGPAXDB.gpaxdata.spidx</i>).
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump 1 =</i>	The rIGPAXDB tuple where the error was found.
<i>Dump 2 =</i>	The rIAPXCTBL tuple where the error was found.

Table 9 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rIGPAXDB tuple has a backward link on the active list, but the user of the previous rIGPAXDB tuples is: - Not CL_NULL, if the current rIGPAXDB tuple has no user, or - CL_NULL if the current rIGPAXDB tuple has a user.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIGPAXDB tuples.
Corrective Action Taken:	The rIGPAXDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIGPAXDB tuple where the error was found.
<i>Bad Data =</i>	The user of the previous rIGPAXDB tuple (<i>rIGPAXDB.client_tag</i>).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key to the rIGPAXDB tuple (<i>rIGPAXDB.gpaxdb</i>).
<i>Dump =</i>	The rIGPAXDB tuple where the error was found.

GPFTIMER

Software Release: 5E12 and later

1. GENERAL PURPOSE FUNCTION TIMERS (GPFTIMER) AUDIT

The RLGPFTIMER relation is used to implement general purpose function timers.

2. DATA STRUCTURES

For details on the fields contained in the RLGPFTIMER relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description."

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	A general purpose function timer client relation tuple is not linked correctly to the rIGPFTIMER tuple being audited. Specifically, the relation ID and tuple key stored in rIGPFTIMER.udata indicate a tuple which does not contain the rIGPFTIMER tuple key in any of its timer linkage fields.
Possible Error Effect:	The rIGPFTIMER tuple is not available for use. Additional tuples also may not be available depending on conditions. If enough tuples are not available, it could cause timer allocation failures due to a lack of resources.
Corrective Action Taken:	The audit reinitializes the rIGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIGPFTIMER.timer_type.
<i>Bad Data =</i>	The client relation ID.
<i>Good Data =</i>	The (logical) tuple key of the client relation.
<i>Logical Index =</i>	The rIGPFTIMER tuple key (rIGPFTIMER.gpftimer).
<i>Dump 1 =</i>	The rIGPFTIMER tuple.
<i>Dump 2 =</i>	The client relation tuple.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The client relation ID in an rIGPFTIMER tuple's user data (rIGPFTIMER.udata) is corrupt or contains a value not expected by the audit.
Possible Error Effect:	The rIGPFTIMER tuple may not be available for use. Additional tuples also may not be available depending on conditions. If enough tuples are not available, it could cause timer allocation failures due to a lack of resources. The other possibility is that a new client type was introduced but the audit was not updated. This error should be escalated to the next level of support. Refer

	to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Corrective Action Taken:	The audit reinitializes the rGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rGPFTIMER.timer_type.
<i>Bad Data =</i>	The client relation ID.
<i>Good Data =</i>	The (logical) tuple key of the client relation.
<i>Logical Index =</i>	The rGPFTIMER tuple key (rGPFTIMER.gpftimer).
<i>Dump =</i>	The rGPFTIMER tuple.

Table 3 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An rGPFTIMER tuple is not idle (rGPFTIMER.timer_type != TMTTNULL) but is either linked to the idle list (rGPFTIMER.link_mem.st_ownk = DBIDLEGPFTIMER+DBSFOHTIMER) or has an invalid timer type (rGPFTIMER.timer_type ≥ TMTMRMAX).
Possible Error Effect:	Varies according to conditions. If the timer is linked to the idle list the effect is minimal. If the timer is in use but contains an invalid timer type, the timer will not be processed correctly when it expires.
Corrective Action Taken:	The audit reinitializes the rGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rGPFTIMER.timer_type.
<i>Bad Data =</i>	rGPFTIMER.link_mem.st_ownk.
<i>Good Data =</i>	TMTMRMAX.
<i>Logical Index =</i>	The rGPFTIMER tuple key (rGPFTIMER.gpftimer).
<i>Dump =</i>	The rGPFTIMER tuple.

Table 4 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An rGPFTIMER tuple is idle (rGPFTIMER.timer_type = TMTTNULL) but is not linked to the idle list (rGPFTIMER.link_mem.st_ownk != DBIDLEGPFTIMER+DBSFOHTIMER).
Possible Error Effect:	The rGPFTIMER tuple may not be available for use. Additional tuples also may not be available depending on conditions. If enough tuples are not available, it could cause timer allocation failures due to a lack of resources. Otherwise, the timer is in use but with a corrupted timer type which means the timer will not be processed correctly when it expires.
Corrective Action Taken:	The audit reinitializes the rGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rGPFTIMER.timer_type.
<i>Bad Data =</i>	rGPFTIMER.link_mem.st_ownk.
<i>Good Data =</i>	DBIDLEGPFTIMER.
<i>Logical Index =</i>	The rGPFTIMER tuple key (rGPFTIMER.gpftimer).
<i>Dump =</i>	The rGPFTIMER tuple.

Table 5 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve a pointer to the client relation tuple which appears to own the rGPFTIMER tuple being audited.
Possible Error Effect:	The rGPFTIMER tuple may not be available for use. Additional tuples also may not be available depending on conditions. If enough tuples are not available, it could cause timer allocation failures due to a lack of resources.
Corrective Action Taken:	The audit reinitializes the rGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rGPFTIMER.timer_type.

<i>Bad Data</i> =	The client relation ID.
<i>Good Data</i> =	The (logical) tuple key of the client relation.
<i>Logical Index</i> =	The rIGPFTIMER tuple key (rIGPFTIMER.gpftimer).
<i>Dump 1</i> =	The rIGPFTIMER tuple.
<i>Dump 2</i> =	The invalid client tuple data retrieved by the audit.

Table 6 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rIGPFTIMER tuple is on a timing or timeout list but was not been processed by the timer process during its last pass through the lists. If the rIGPFTIMER tuple is not processed, the timer it represents will never expire.
Possible Error Effect:	Varies according to conditions including what timer client relation is involved and what the timer is being used for.
Corrective Action Taken:	The audit reinitializes the rIGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIGPFTIMER.timer_type.
<i>Bad Data</i> =	The timer's increment field (rIGPFTIMER.tcount).
<i>Good Data</i> =	The timer's check field (rIGPFTIMER.shadow_cnt).
<i>Logical Index</i> =	The rIGPFTIMER tuple key (rIGPFTIMER.gpftimer).
<i>Dump 1</i> =	The rIGPFTIMER tuple.
<i>Dump 2</i> =	The client relation tuple (the relation ID and tuple key can be read out of rIGPFTIMER.udata).

Table 7 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	An rIGPFTIMER is being held in local (non-relation) data (the relation ID in the rIGPFTIMER.udata is TMNO_REL_ID) but has been held there too long.
Possible Error Effect:	The rIGPFTIMER tuple may not be available for use. Additional tuples also may not be available depending on conditions. If enough tuples are not available, it could cause timer allocation failures due to a lack of resources.
Corrective Action Taken:	The audit reinitializes the rIGPFTIMER tuple and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIGPFTIMER.timer_type.
<i>Bad Data</i> =	rIGPFTIMER.quarstate.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIGPFTIMER tuple key (rIGPFTIMER.gpftimer).
<i>Dump</i> =	The rIGPFTIMER tuple.

GRPPORT

Software Release: 5E10 and later

1. GROUP AND MEMBER NUMBER TO PORT (GRPPORT) AUDIT

To process a call during stand alone (SA) condition, the routing and terminal allocation (RTA) routing process in the remote switching module (RSM) and in the optically remote module (ORM) must determine the ports that are available for terminating calls. For every member of the global multi-line hunt group (MLHG), a rIGROUP_PORT tuple must exist in the switching module (SM) which owns the port in the designated hunting SM and in the designated routing SM. During SA, related tuples of RLGROUP_PORT relation, tuples for which the RSM/ORM are the owners or the hunting SM, have an associated active member in the multiline hunt group member status relation (RLMEM) or in the multiline hunt large group member status relation (RLLGMEM) see MEM/LGMEM audits).

The RLGROUP_PORT relation works with both rLMEM and RLLGMEM relations to ensure that missing tuples are added to these relations. The relation also makes sure that for all related tuples of rIGROUP_PORT, the corresponding member in rLMEM/rLLGMEM is active (set to "1").

In the RSM/ORM, the audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode. This audit is scheduled only during the SA mode.

2. DATA STRUCTURES

The data structures checked by the GRPPORT audit are the relations RLMEM and RLLGMEM.

The RLMEM/RLLGMEM relations are dynamic relations that contain information about small/large multi-line hunt group. This information is kept on bit maps, with each bit representing a member of the MLHG. The bit map is included in each tuple in the status relations RLMEM/RLLGMEM. When a member is populated for a MLHG, the corresponding bit must be set to "1" to indicate the member is active. If the member is not active, the bit must be set to "0".

3. AUDIT CHECKS

The GRPPORT audit uses the static driver (DRSTATIC) to step through the RLGROUP_PORT relation and sequentially retrieve tuples. The DRSTATIC driver invokes a semantic check function that verifies each rIGROUP_PORT tuple against the rLMEM/rLLGMEM tuples by using the key (group number) of rIGROUP_PORT to read the dynamic relations. To determine which dynamic relation to read (RLMEM or RLLGMEM), the semantic function reads RLRT_MHG relation to get the hunt type.

The audit looks specifically for missing tuples, and for members that are marked as not active in rLMEM/rLLGMEM tuple, but for which there is a corresponding tuple in RLGROUP_PORT relation.

4. ERROR HANDLING

Missing tuples are created in RLMEM/RLLGMEM relation. Bits that are set to "0" (not active) and that are associated with a MLHG member are corrected to "1" (active). If a tuple of the member status relation contains an invalid data in its key, the key is corrected. If rIRT_MHG tuple contains an improper hunt type, an error message requesting manual action is printed on the read-only printer (ROP).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The multi-line hunt group information relation, RLRT_MHG, contains a tuple which has an invalid hunt type. This error will repeat until fixed by the user.
Possible Error Effect:	Inability to hunt to lines in this group and possible ASSERT failures.
Corrective Action Taken:	None.
Manual Action Required:	The hunt type in rIRT_MHG is invalid. Determine the nature of the database inconsistency and correct it. This message will repeat until corrective action is taken by the user.
Dump Description:	
<i>Error Address</i> =	Address of the tuple buffer.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	rIRT_MHG tuple.

Table 2 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	The corresponding member in the dynamic relations RLMEM/RLMGEM is initialized incorrectly to invalid ("0").
Possible Error Effect:	Initialization was wrong and calls will fail to that member.
Corrective Action Taken:	The member will be set to valid ("1").
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the dynamic tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	1 (valid member).
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rIMEM/rILGMEM tuple.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	The tuple of the relation RLMEM has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIMEM tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	The correct value of the key.
<i>Dump</i> =	The contents of the rIMEM tuple with the wrong key.

Table 4 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	The tuple of the relation rILGMEM has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rILGMEM tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	The correct value of the key.
<i>Dump</i> =	The contents of the rILGMEM tuple with the wrong key.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing from the dynamic relation RLMEM/RLMGEM.
Possible Error Effect:	MLHG members will be initialized incorrectly and call will fail.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the dynamic tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	0 = Missing tuple in rMEM relation. 1 = Missing tuple in rLGMEM relation.
<i>Logical Index</i> =	Key of the lost tuple.
<i>Dump</i> =	Dump of the lost tuple.

Table 6 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit has found an inconsistency between the static relations RLGRPPORT and RLRT_MHG. The key or rIGRPPORT tuple is used to access a tuple in RLRT_MHG relation but no matching tuple is found and the audit cannot continue to do checks.
Possible Error Effect:	Audit continue to the next tuple or RLGRPPORT relation.
Corrective Action Taken:	Audit report an error.
Manual Action Required:	Verify tuples rIRT_MHG and rIGRPPORT in the ODD, using the rIGRPPORT key found in the "Logical Index" field of the audit report. Corresponding tuple rIRT_MHG with the same key should exist on the same SM or the tuple rIGRPPORT may be invalid. Correct any inconsistencies found.
Dump Description:	
<i>Error Address</i> =	The address of the rIGRPPORT tuple.
<i>Bad Data</i> =	The key of rIRT_MHG tuple returned from the base.
<i>Good Data</i> =	The key used to access rIRT_MHG tuple.
<i>Logical Index</i> =	The key used to access rIRT_MHG tuple.
<i>Dump</i> =	The contents of the rIGRPPORT tuple.

GRPTGC

Software Release: 5E10 and later

1. GROUP TRUNK GROUP CONTROLS (GRPTGC) AUDIT

Trunk group controls are a set of regulators used to protect an office from excessive routing requests and to expand routing patterns during busy periods. These regulators control the number of calls that can access a trunk group. A hierarchy exists within the trunk group controls. The group trunk group controls relation (RLGRP_TGC) maps active trunk groups in an office to trunk groups that have controls associated with them. The trunk group controls relation (RLTGC) maps the trunk groups and their associated manual controls to a specific control. The extended reroute relation (RLEXRR) maps a specific reroute control to a series of trunk groups designated to be the trunks on which the call attempts termination. The dynamic overload control (RLDOC) and trunk reservation (RLTR) relations map the TGs into their associated automatic controls.

The group trunk group controls audit (GRPTGC) verifies that busy tuples in the group trunk group controls relation (RLGRP_TGC) are correctly linked to one of the following relations: the trunk owner relation (RLTKOWNER), the trunk status relation (RLTKSTAT), or the large trunk status relation (RLLGTKSTAT) to ensure that each busy tuple has a valid trunk group associated with it. The audit verifies that the control(s) are properly linked to the trunk group controls relation (RLTGC), which may specify from one to four controls, and/or the dynamic overload control (RLDOC) and/or trunk reservation (RLTR) relations. The GRPTGC audit also checks for lost resources by verifying linkages to the database headcells relation (RLDB_HEADC).

This audit is part of the routing and terminal allocation audits subgroup. It operates in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The data structures checked by the GRPTGC audit are the group trunk group controls relation (RLGRP_TGC), the trunk group controls relation (RLTGC), the dynamic overload control relation (RLDOC), the trunk reservation relation (RLTR), the trunk status relation (RLTKSTAT), the large trunk status relation (RLLGtKstat), the trunk owner relation (RLTKOWNER), and the database head cells relation (RLDB_HEADC). Figure 1 illustrates the layout of RLGRP_TGC and shows the items checked by the audit. The RLGRP_TGC is a dynamic relation that contains information about controls placed on individual trunk groups. Included in the information within the tuples is the point-back to one of the RLTKSTAT, RLLGtKstat, or RLTKOWNER relations, and the trunk group controls fields (cant, skip, cro, canf, rr, tr, and doc) that point the to RLTGC, RLTR, and RLDOC relations.

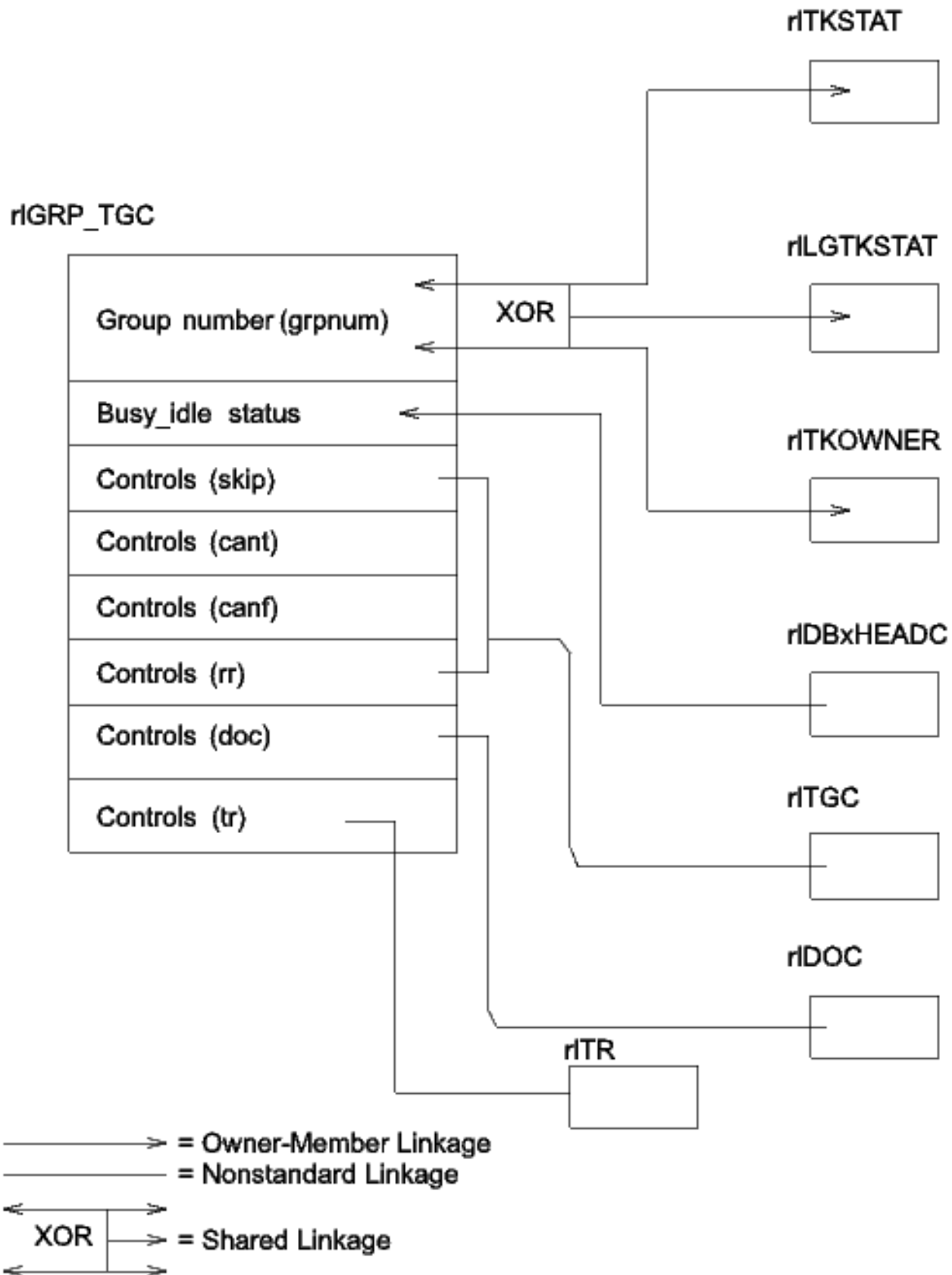


Figure 1 (GRPTGC-1) rIGRP_TGC Linkages**3. AUDIT CHECKS**

To verify the linkages in the group trunk group controls relation, the GRPTGC audit calls the linkage driver (DRLINK) once for each audit segment. The DRLINK driver ensures that each tuple in RLGRP_TGC has at least one valid linkage to the appropriate control relations. The DRLINK driver also verifies the owner-member linkage to the RLDB_HEADC relation, and the one-to-one linkage (grpnum) to one of the following relations: RLTKSTAT, RLLGTKSTAT, or RLTKOWNER. The DRLINK driver invokes the semantic check function after verifying the linkages of each rIGRP_TGC tuple. The semantic check function performs the following checks:

- (a) The audit verifies that each of the control fields is either linked to the appropriate control relation or contains a null pointer.
 - If they are not null, the cant, skip, cro, canf, and rr attributes should be linked to rITGC.
 - If it is not null, the tr attribute should be linked to rITR.
 - If it is not null, the doc attribute should be linked to rIDOC.

If a link exists, the audit verifies that the type of control associated with the link is the type of control represented in the rITGC, rITR, or rIDOC relation.

- (b) The audit verifies that all tuples in rIGRP_TGC are either busy or idle.
- (c) The audit verifies that all active DOC controls are owned by a network management timing list.
- (d) The audit verifies that all tuples owned by a network management timing list have valid timing links.

4. ERROR HANDLING

The audit reports errors when any linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples and links them to the rIGRP_TGC idle list.

The GRPTGC audit schedules the DBHEADC, DOC, LGTKSTAT, NMTR, TGC, TKOWNER, and TKSTAT audits as related audits when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: CANFCNTL

Error Code:	CANFCNTL
Error Description:	The canf field pointer is not pointing to a rITGC tuple with the canf control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type it would be possible for the trunk group to have multiple controls of the same type.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of canf field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rITGC tuple.

Table 2 Error Code: CANTCNTL

Error Code:	CANTCNTL
Error Description:	The cant field pointer is not pointing to a rITGC tuple with the cant control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type it would be possible for the trunk group to have multiple controls of the same type.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of cant field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rITGC tuple.

Table 3 Error Code: CROCNTL

Error Code:	CROCNTL
Error Description:	The cro field pointer is not pointing to a rITGC tuple with the cro control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type, it would be possible for the trunk group to have multiple controls of the same type.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of cro field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rITGC tuple.

Table 4 Error Code: DOCCNTL

Error Code:	DOCCNTL
Error Description:	The doc field pointer is not pointing to a rIDOC tuple with the cant or skip control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type it would be possible for the trunk group to have an incorrect response to a DOC signal.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of doc field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rIDOC tuple.

Table 5 Error Code: DOCLVL

Error Code:	DOCLVL
Error Description:	The doc_lvl field indicates that a DOC signal has been received for this DOC, but no timer is timing the control.
Possible Error Effect:	The DOC would never time out. More calls would be controlled then was necessary.
Corrective Action Taken:	The doc_lvl field is set to zero (the DOC control is turned off).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of doc_lvl field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump</i> =	Contents of rIGRP_TGC tuple.

Table 6 Error Code: GSCACT

Error Code:	GSCACT
Error Description:	The gsc_act field indicates that a GSC control is active, but no timer is timing the control.
Possible Error Effect:	The GSC control would never time out. More calls than intended would be forced to seek alternate routes.
Corrective Action Taken:	The gsc_act field is set to zero (the GSC control is turned off).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIGRP_TGC tuple.
<i>Bad Data =</i>	Contents of gsc_act field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Contents of grpkey field.
<i>Dump =</i>	Contents of rIGRP_TGC tuple.

Table 7 Error Code: NMTIML

Error Code:	NMTIML
Error Description:	The timing list used to turn off automatic NM controls is corrupted.
Possible Error Effect:	The control will never time out.
Corrective Action Taken:	The audit removes the tuple from the corrupted timing list (the control is turned off).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIGRP_TGC tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Contents of grpkey field.
<i>Dump =</i>	Contents of rIGRP_TGC tuple.

Table 8 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A tuple with valid linkages and data to qualify as busy is not on the busy list.
Possible Error Effect:	If this tuple is on the idle list the data could be lost when the supposedly idle tuple is made active.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIGRP_TGC tuple.
<i>Bad Data =</i>	Contents of st_ownk field.
<i>Good Data =</i>	Busy list headcell.
<i>Logical Index =</i>	Key of rIGRP_TGC tuple.
<i>Dump =</i>	Contents of rIGRP_TGC tuple.

Table 9 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A tuple has null pointer and data to indicate that it is idle but it is not on the idle list.
Possible Error Effect:	This is an idle tuple which can never be made active since it is not on the idle list.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIGRP_TGC tuple.
<i>Bad Data =</i>	Contents of st_ownk field.
<i>Good Data =</i>	Idle list headcell.
<i>Logical Index =</i>	Key of rIGRP_TGC tuple.
<i>Dump =</i>	Contents of rIGRP_TGC tuple.

Table 10 Error Code: RRCNTL

Error Code:	RRCNTL

Error Description:	The rr field pointer is not pointing to a rITGC tuple with the rr control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type it would be possible for the trunk group to have multiple controls of the same type.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of rr field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rITGC tuple.

Table 11 Error Code: SKIPCNTL

Error Code:	SKIPCNTL
Error Description:	The skip field pointer is not pointing to a rITGC tuple with the skip control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type it would be possible for the trunk group to have multiple controls of the same type.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of skip field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rITGC tuple.

Table 12 Error Code: TRCNTL

Error Code:	TRCNTL
Error Description:	The tr field pointer is not pointing to a rITR tuple with the cant or skip control type.
Possible Error Effect:	If the link points to a tuple with the incorrect control type it would be possible for the trunk group to have an incorrect response to a TR signal.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGRP_TGC tuple.
<i>Bad Data</i> =	Contents of tr field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Contents of grpkey field.
<i>Dump 1</i> =	Contents of rIGRP_TGC tuple.
<i>Dump 2</i> =	Contents of rITR tuple.

GSDB-A

Software Release: 5E10 only

1. GROUP SCAN DATA BLOCK (GSDB) AUDIT

A group scan data block is linked to an active concentrated buffer headcell when a line origination is buffered. Each group scan data block represents a group of eight lines. Each line in a group is represented by one bit. If the bit is set, there is an origination on the line waiting to be processed.

The group scan data block audit (GSDB) ensures that group scan data block tuples (rIGSDBs) are properly linked to one of the concentrated buffer headcell tuples (rICBHC). The rICBHC tuples represent the idle list or one of several active lists. The audit also checks the name of the first port in the scan group that is stored in an rIGSDB tuple.

This audit is part of the peripheral control audits subgroup.

2. DATA STRUCTURES

The data structures checked by the GSDB audit are the group scan data block relation (RLGSDB) and the concentrated buffer headcell relation (RLCBHC). The RLGSDB relation is owned by RLCBHC. Figure 1 illustrates the layout of a group scan data block, and Figure 2 illustrates the owner-member linkage between rICBHC and rIGSDB.

3. AUDIT CHECKS

The GSDB audit uses the static driver (DRSTATIC) to step through the group scan data block relation (RLGSDB). The DRSTATIC driver calls a dynamic to static existence check function to perform the following check.

- (1) For each non-SLC[®] 96 related tuple in RLGSDB, the existence check function calculates a grid name. The grid name is used to verify that an associated tuple exists in the circuit data relation (RLCKTDATA).
- (2) Next, a semantic check function is called to perform the following checks:
 - (a) The audit ensures that all port names stored in rIGSDB tuples are correct. The port name (cktname) is algorithmically derived from the logical key; if there is an error, an incorrect circuit will be addressed.
 - (b) The audit checks the member linkage of each rIGSDB to an rICBHC tuple to verify that it belongs to a list owned by an rICBHC tuple.

4. ERROR HANDLING

If a tuple that should exist is missing in the rLCKTDATA relation, the audit inserts that tuple into the relation.

Any tuples that fail these checks are reinitialized and placed on the idle list. Any error codes used within the drivers (drstatic, drlink) also apply here.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	An rIGSDB tuple has an incorrect cktname stored in it. The cktname is

	algorithmically derived from the logical key (grpname). Therefore, something has overwritten it.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit reinitializes the rIGSDB tuple and places it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGSDB tuple.
<i>Bad Data</i> =	rIGSDB tuple's ckname.
<i>Good Data</i> =	Derived ckname.
<i>Logical Index</i> =	Key of rIGSDB tuple.
<i>Dump</i> =	rIGSDB tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIGSDB tuple was found. Either the read of rICKTDATA was successful and the equipage status was SMUNEQ, or the read failed.
Possible Error Effect:	Garbage left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGSDB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIGSDB.
<i>Dump</i> =	rIGSDB tuple.

Table 3 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	A dummy rIGSDB tuple contains nonzero data. Since the dummy tuple is never used for anything, something has written into it in error.
Possible Error Effect:	Invalid attempts to use the dummy tuple.
Corrective Action Taken:	The audit reinitializes the dummy tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGSDB.
<i>Bad Data</i> =	Nonzero tuple data.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rIGSDB.
<i>Dump</i> =	Dummy rIGSDB.

GSDB-B

Software Release: 5E11 and later

1. GROUP SCAN DATA BLOCK (GSDB) AUDIT

A group scan data block is linked to an active concentrated buffer headcell when a line origination is buffered. Each group scan data block represents a group of eight lines. Each line in a group is represented by one bit. If the bit is set, there is an origination on the line waiting to be processed.

The group scan data block audit (GSDB) ensures that group scan data block tuples (rIGSDBs) are properly linked to one of the concentrated buffer headcell tuples (rICBHC). The rICBHC tuples represent the idle list or one of several active lists. The audit also checks the name of the first port in the scan group that is stored in an rIGSDB tuple.

This audit is part of the Peripheral Control audits subgroup.

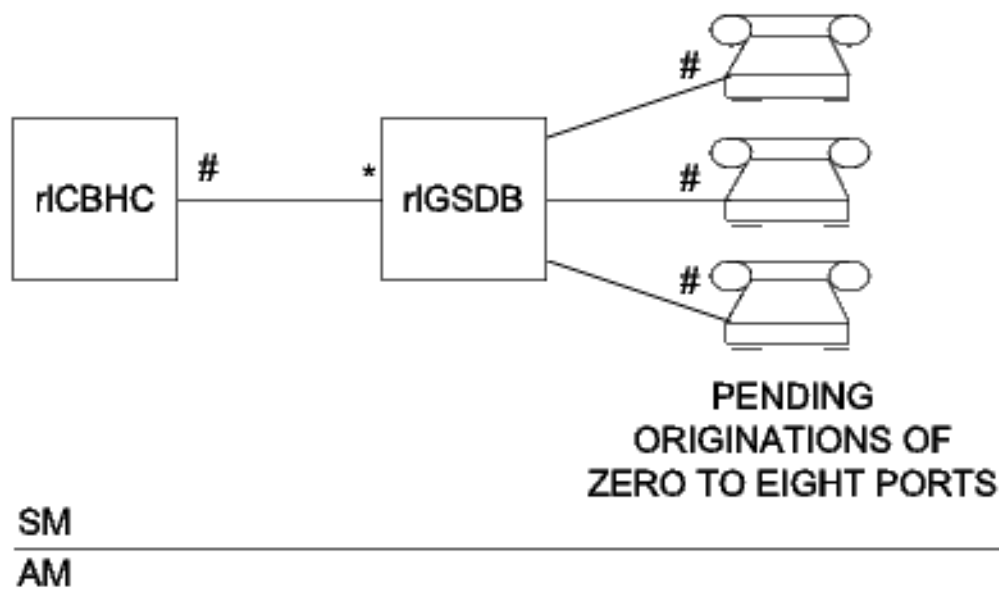
2. DATA STRUCTURES

The data structures checked by the GSDB audit are the group scan data block relation (RLGSDB) and the concentrated buffer headcell relation (RLCBHC). The RLGSDB relation is owned by RLCBHC. Figure 1 illustrates the layout of a group scan data block, and Figure 2 illustrates the owner-member linkage between RLCBHC and RLGSDB.

rIGSDB

LOGICAL KEY (grpname)
PORT NAME OF FIRST PORT IN SCAN GROUP (cktname)
TIME WHEN ORIG REPORT IS DISPENSED FROM REPORT QUEUE (origtime)
SCAN MASK BITS OF 8 LINES IN GROUP (lineihbt)
LAST LOOK BITS OF 8 LINES IN GROUP (line11)

Figure 1 (GSDB-1) Group Scan Data Block



— * = Owner-Member Linkage

Figure 2 (GSDB-2) rIGSDB Linkages

3. AUDIT CHECKS

The GSDB audit uses the static driver (DRSTATIC) to step through the group scan data block relation (RLGSDB). The DRSTATIC driver calls a dynamic to static existence check function to perform the following check.

- (1) For each non-SLC[®] 96 related tuple in RLGSDB, the existence check function calculates a grid name. The grid name is used to verify that an associated tuple exists in the circuit data relation (RLCKTDATA).
- (2) Next, a semantic check function is called to perform the following checks:
 - (a) The audit ensures that all port names stored in rIGSDB tuples are correct. The port name (cktname) is algorithmically derived from the logical key; if there is an error, an incorrect circuit will be addressed.
 - (b) The audit checks the member linkage of each rIGSDB to an rICBHC tuple to verify that it belongs to a list owned by an rICBHC tuple.

4. ERROR HANDLING

If a tuple that should exist is missing in the rICKTDATA relation, the audit inserts that tuple into the relation.

Any tuples that fail these checks are reinitialized and placed on the idle list. Any error codes used within the drivers (drstatic, drlink) also apply here.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	An rIGSDB tuple has an incorrect cktname stored in it. The cktname is algorithmically derived from the logical key (grpname). Therefore, something has overwritten it.
Possible Error Effect:	An incorrect circuit will be addressed.
Corrective Action Taken:	The audit reinitializes the rIGSDB tuple and places it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGSDB tuple.
<i>Bad Data</i> =	rIGSDB tuple's cktname.
<i>Good Data</i> =	Derived cktname.
<i>Logical Index</i> =	Key of rIGSDB tuple.
<i>Dump</i> =	rIGSDB tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIGSDB tuple was found. Either the read of rICKTDATA was successful and the equipage status was SMUNEQ, or the read failed.
Possible Error Effect:	Garbage left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGSDB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIGSDB.
<i>Dump</i> =	rIGSDB tuple.

Table 3 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	A dummy rIGSDB tuple contains nonzero data. Since the dummy tuple is never used for anything, something has written into it in error.
Possible Error Effect:	Invalid attempts to use the dummy tuple.
Corrective Action Taken:	The audit reinitializes the dummy tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIGSDB.
<i>Bad Data</i> =	Nonzero tuple data.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rIGSDB.
<i>Dump</i> =	Dummy rIGSDB.

GSLK-A

Software Release: 5E11 - 5E13

1. GROUP SUMMARY SCAN DATA LINKAGE BLOCK (GSLK) AUDIT

A group summary scan data linkage block is used by peripheral control to link different types of group summary data block (GSDB) relations (that is, RLGSDb, RLAI_GSDb, RLDNGSDb, and so forth) to the concentrated buffer head cell relation (RLCBHC).

2. DATA STRUCTURES

For details on the fields contained in the RLGSLK relation, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	An rGSLK tuple was found with a bad one-to-one linkage to a group summary scan data block (that is, rIGSDb, rIAI_GSDb). This rGSLK tuple points to another tuple, but that tuple does not point back.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rGSLK tuple.
<i>Bad Data =</i>	Linkage map identification.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump =</i>	rGSLK tuple.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rGSLK tuple is queued for origination, but the unit type attribute (rGSLK.utype) is not recognized by the audit.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rGSLK tuple.
<i>Bad Data =</i>	Unit type value (rGSLK.unittyp) from the rGSLK tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump =</i>	rGSLK tuple.

Table 3 Error Code: INVOWNER

Error Code:	INVOWNER

Error Description:	The rGSLK tuple is linked to an invalid list.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSLK tuple.
<i>Bad Data</i> =	Contents of the owner linkage (rGSDB.gscbm.st_ownk).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump</i> =	rGSLK tuple.

Table 4 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The rGSLK tuple is linked to the idle list, but the unit type attribute (rGSLK.unitttype) is not set to SMNULLUNIT.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSLK tuple.
<i>Bad Data</i> =	Unit type (rGSLK.unitttype).
<i>Good Data</i> =	SMNULLUNIT.
<i>Logical Index</i> =	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump</i> =	rGSLK tuple.

Table 5 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rGSLK tuple is on the idle list, but is still linked to a group summary scan data block (that is, rGSDB, rIAI_GSDB).
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSLK tuple.
<i>Bad Data</i> =	Nonnull data found in the rGSLK tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump</i> =	rGSLK tuple.

GSLK-B

Software Release: 5E14 and later

1. GROUP SUMMARY SCAN DATA LINKAGE BLOCK (GSLK) AUDIT

A group summary scan data linkage block is used by peripheral control to link different types of group summary data block (GSDB) relations (that is, RLGSDb, RLAI_GSDb, RLDNGSDb, and so forth) to the concentrated buffer head cell relation (RLCBHC).

2. DATA STRUCTURES

For details on the fields contained in the RLGSLK relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	An rIGSLK tuple was found with a bad one-to-one linkage to a group summary scan data block (that is, rIGSDb, rLAI_GSDb). This rIGSLK tuple points to another tuple, but that tuple does not point back.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rIGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIGSLK tuple.
<i>Bad Data =</i>	Linkage map identification.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (rIGSLK.linknum) of the rIGSLK tuple.
<i>Dump =</i>	rIGSLK tuple.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rIGSLK tuple is queued for origination, but the unit type attribute (rIGSLK.utype) is not recognized by the audit.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rIGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIGSLK tuple.
<i>Bad Data =</i>	Unit type value (rIGSLK.unittyp) from the rIGSLK tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (rIGSLK.linknum) of the rIGSLK tuple.
<i>Dump =</i>	rIGSLK tuple.

Table 3 Error Code: INVOWNER

Error Code:	INVOWNER

Error Description:	The rGSLK tuple is linked to an invalid list.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSLK tuple.
<i>Bad Data</i> =	Contents of the owner linkage (rGSDB.gscbm.st_ownk).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump</i> =	rGSLK tuple.

Table 4 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The rGSLK tuple is linked to the idle list, but the unit type attribute (rGSLK.unitttype) is not set to SMNULLUNIT.
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSLK tuple.
<i>Bad Data</i> =	Unit type (rGSLK.unitttype).
<i>Good Data</i> =	SMNULLUNIT.
<i>Logical Index</i> =	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump</i> =	rGSLK tuple.

Table 5 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rGSLK tuple is on the idle list, but is still linked to a group summary scan data block (that is, rGSDB, rAI GSDB).
Possible Error Effect:	May cause errors in processing the queue for line originations.
Corrective Action Taken:	The audit reinitializes the rGSLK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rGSLK tuple.
<i>Bad Data</i> =	Nonnull data found in the rGSLK tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key (rGSLK.linknum) of the rGSLK tuple.
<i>Dump</i> =	rGSLK tuple.

16. H - IM

HCCALLTQ

Software Release: 5E10 and later

1. HEAD CELLS FOR CALL TYPE QUEUES (HCCALLTQ) AUDIT

Features of the operator services position system (OSPS) include basic automatic call distribution (ACD) and call handling. These features contain the call processing software needed for an operator to log onto a basic services terminal (BST) and handle requests for operator services. The ACD device routes calls to available positions. Audits are needed to maintain the integrity of the data structures needed to perform these OSPS functions. The HCCALLTQ audit verifies call type related information.

2. DATA STRUCTURES

The HCCALLTQ audit uses three relations, the static call type information relation (RLCTINFO), the dynamic head cells for call type queues relation (RLHCCALLTQ), and the call queue data block relation (RLCQDB). These data structures are shown in Figure 1.

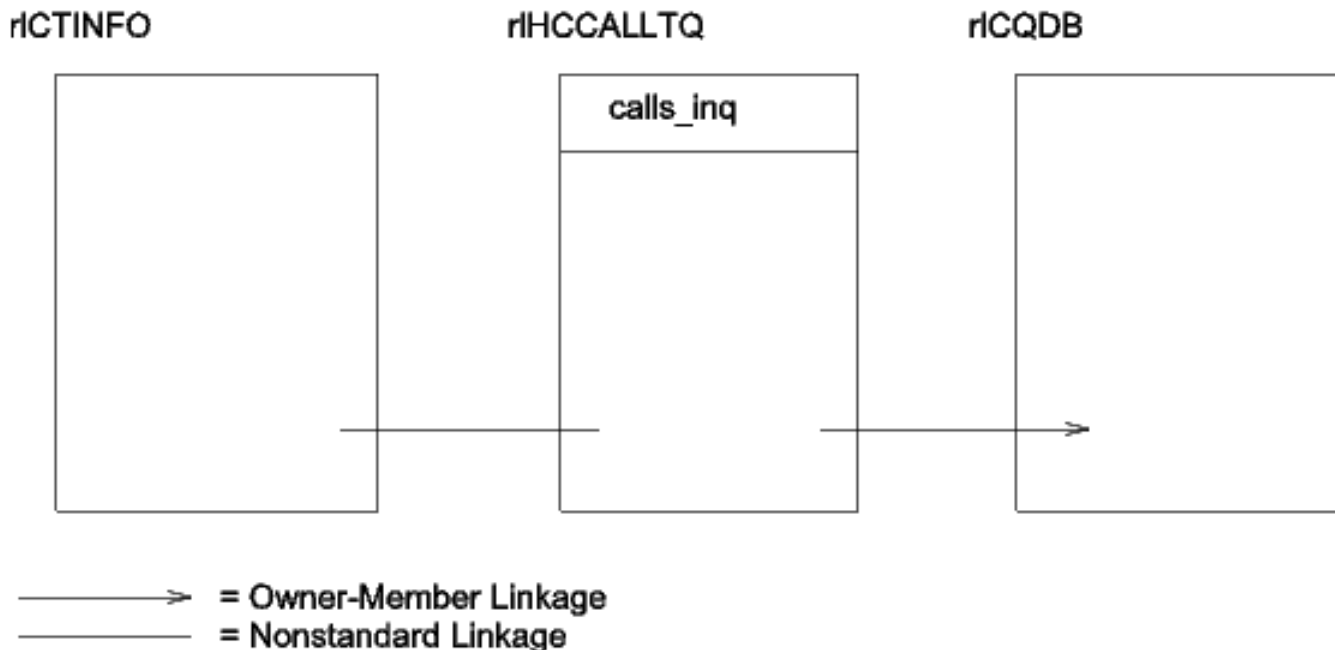


Figure 1 (HCCALLTQ-1) Data Structures of the HCCALLTQ Audit

3. AUDIT CHECKS

The HCCALLTQ audit calls the dynamic driver (DRDYN) to step through the call queue data block head cells. The driver examines the links for each tuple. If they are valid, it invokes the semantic check function to perform the following checks:

- The audit ensures that the fields in rLHCCALLTQ contain valid values. Most fields should contain the same values as the corresponding fields in rLCTINFO.
- The audit makes a count of the call queue data blocks that are linked to the rLHCCALLTQ tuple and checks the count with the **calls_inq** field of rLHCCALLTQ.
- The audit also validates the head cell linkages to rLCQDB.

- (d) The audit also checks for extra rIHCCALLTQ tuples. If there is not a corresponding rICTINFO tuple, the rIHCCALLTQ tuple is zeroed out.

4. ERROR HANDLING

If any inconsistencies are found in rIHCCALLTQ, the inconsistent values are replaced with the correct data from rICTINFO or, in the case of a count error, the calculated count.

The HCCALLTQ audit schedules the CQDB audit as a related audit when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of rICQDB tuples linked to this HCCALLTQ is inconsistent with the value in the calls_inq attribute.
Possible Error Effect:	The tuple contains invalid information.
Corrective Action Taken:	The value in the calls_inq attribute is replaced with the correct value obtained by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIHCCALLTQ.
<i>Bad Data =</i>	cltq_ptr→calls_inq.
<i>Good Data =</i>	calls_count.
<i>Logical Index =</i>	Key of rIHCCALLTQ.
<i>Dump =</i>	rIHCCALLTQ tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra tuple was found.
Possible Error Effect:	The extra tuple contains garbage, which may be invalidly accessed and presumed to be valid data.
Corrective Action Taken:	The extra rIHCCALLTQ tuple that was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIHCCALLTQ.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIHCCALLTQ.
<i>Dump =</i>	rIHCCALLTQ tuple.

Table 3 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static tuple rICTINFO does not exist for the dynamic tuple rIHCCALLTQ.
Possible Error Effect:	The data in rIHCCALLTQ is based upon data in rICTINFO, but the rICTINFO tuple doesn't exist when it should.
Corrective Action Taken:	The rICTINFO must be created and initialized.
Manual Action Required:	Action must be taken to create and initialize the missing rICTINFO tuple.
Dump Description:	
<i>Error Address =</i>	Address of rIHCCALLTQ tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIHCCALLTQ.
<i>Dump =</i>	rIHCCALLTQ.

Table 4 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The dynamic data for this tuple in rHCCALLTQ is corrupted. It is compared against the static data in rICTINFO.
Possible Error Effect:	The tuple contains invalid information.
Corrective Action Taken:	The audit will replace the incorrect data with valid data from the rICTINFO tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHCCALLTQ.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rHCCALLTQ.
<i>Dump =</i>	rHCCALLTQ tuple.

HCCLLQ

Software Release: 5E11 and later

1. HEAD CELLS OF CALL QUEUES (HCCLLQ) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. Each call queue data block whether it is idle or representing a call is associated with the data in the RLHC_CLLQ (head cells of call queues) relation. Data in this relation is audited by the HCCLLQ audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLHC_CLLQ relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the idle rHC_CLLQ tuple's maximum number of calls count, max_calls, value is incorrect.
Possible Error Effect:	Calls may not be queued properly.
Corrective Action Taken:	The rHC_CLLQ max_calls field will be corrected to the value of the call queue maximum calls Glcqmcs_global parameter.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the idle rHC_CLLQ tuple.
<i>Bad Data =</i>	The tuple's maximum number of calls count (rHC_CLLQ.max_calls).
<i>Good Data =</i>	The value of the call queue maximum calls global parameter (Glcqmcs).
<i>Logical Index =</i>	The logical key of the idle rHC_CLLQ tuple.
<i>Dump =</i>	The idle rHC_CLLQ tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the rHC_CLLQ tuple's calls in queue count, calls_inq, value is incorrect.
Possible Error Effect:	Calls in this queue may not be processed properly.
Corrective Action Taken:	The rHC_CLLQ calls_inq field will be corrected to the value calculated by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rHC_CLLQ tuple.
<i>Bad Data =</i>	The tuple's calls in queue count (rHC_CLLQ.calls_inq).
<i>Good Data =</i>	The value for calls in queue calculated by the audit.
<i>Logical Index =</i>	The logical key of the rHC_CLLQ tuple.

<i>Dump</i> =	The rIHC_CLLQ tuple.

Table 3 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The audit found an inconsistency between the dynamic rIHC_CLLQ tuple and the static rICQPARM tuple.
Possible Error Effect:	This may result in call queuing not functioning properly.
Corrective Action Taken:	The rIHC_CLLQ tuple is recovered based on the corresponding static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIHC_CLLQ tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rIHC_CLLQ tuple.
<i>Dump 1</i> =	The rIHC_CLLQ tuple.
<i>Dump 2</i> =	The rICQPARM tuple.

HCSERVTM

Software Release: 5E10 and later

1. HEAD CELLS FOR SERVING TEAM POSITION QUEUES (HCSERVTM) AUDIT

Features of the operator services position system (OSPS) include basic automatic call distribution (ACD) and call handling. These features contain the call processing software needed for an operator to log onto a basic services terminal (BST) and handle requests for operator services. The ACD device routes calls to available positions. Audits are needed to maintain the integrity of the data structures needed to perform these OSPS functions. The HCSERVTM audit ensures the accuracy of the position lists on a serving team basis and verifies the number of positions logged into each team field.

2. DATA STRUCTURES

The HCSERVTM audit uses three data structures: the static serving team information relation (RLSTINFO), the dynamic serving team head cells relation (RLHCSERVTM), and the serving team queue data block relation (RLSTQDB). These data structures are shown in Figure 1.

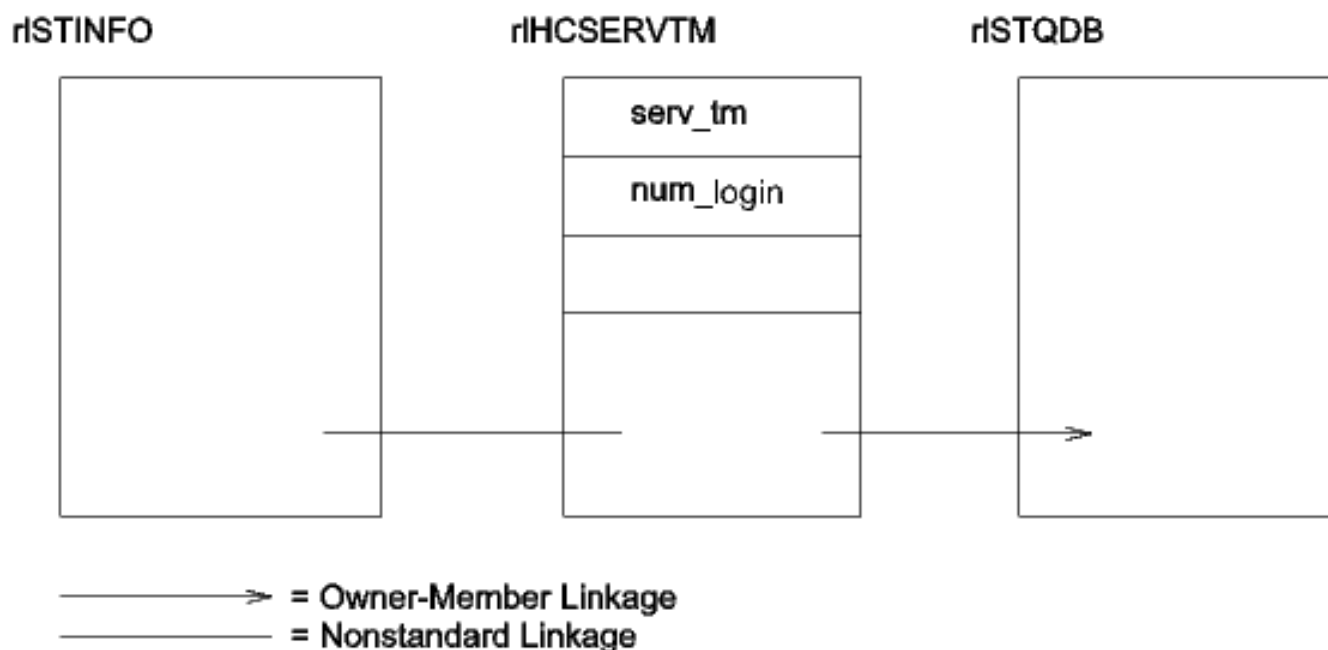


Figure 1 (HCSERVTM-1) Data Structures of the HCSERVTM Audit

3. AUDIT CHECKS

The HCSERVTM audit calls the dynamic driver (DRDYN) to step through the serving team queue data block head cells relation (rLHCSERVTM). If the linkages are valid, the audit will invoke the semantic check function, which performs the following checks:

- (a) The serving team field (serv_tm), which is the key of rLHCSERVTM, should have a match in the corresponding field of rLSTINFO. If it does not, the rLHCSERVTM tuple is deleted as an extra tuple.
- (b) The audit counts the serving team queue data blocks in rLSTQDB and ensures that the count equals the count in the number of logins field (num_login) in rLHCSERVTM.

4. ERROR HANDLING

The audit deletes rHCSEVTM tuples without matching rISTINFO tuples. The audit replaces erroneous counts in the num_login field with the calculated count.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of rISTQDB tuples linked to this HCSEVTM is inconsistent with the value in the num_login attribute.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The value in rHCSEVTM tuple attribute (num_login) is replaced with the calculated value obtained by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rHCSEVTM tuple.
<i>Bad Data</i> =	rHCSEVTM tuple attribute num_login.
<i>Good Data</i> =	Calculated value of login_count.
<i>Logical Index</i> =	rHCSEVTM tuple.
<i>Dump</i> =	rHCSEVTM tuple.

Table 2 Error Code: DUPTUP

Error Code:	DUPTUP
Error Description:	An extra rHCSEVTM tuple was found that does not have a corresponding rIST_PARM tuple.
Possible Error Effect:	The extra rHCSEVTM tuple contains invalid information, which may be invalidly accessed and presumed to be valid data.
Corrective Action Taken:	The extra rHCSEVTM tuple that was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rHCSEVTM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rHCSEVTM tuple.
<i>Dump</i> =	rHCSEVTM tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rHCSEVTM tuple was found that does not have a corresponding rISTINFO tuple.
Possible Error Effect:	The extra rHCSEVTM tuple contains invalid information, which may be invalidly accessed and presumed to be valid data.
Corrective Action Taken:	The extra rHCSEVTM tuple that was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rHCSEVTM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rHCSEVTM tuple.
<i>Dump</i> =	rHCSEVTM tuple.

Table 4 Error Code: ODD_COUNT

Error Code:	ODD_COUNT
Error Description:	The number of rISTQDB tuples shown to be handling calls from supporting queues does not equal the num_sup attribute in rHCSEVTM tuple.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The value in the rIHCSERVMT tuple attribute (num_sup) is replaced with the calculated value obtained by the audit (sup_count).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIHCSERVMT tuple.
<i>Bad Data =</i>	rIHCSERVMT tuple attribute num_sup.
<i>Good Data =</i>	The calculated value sup_count.
<i>Logical Index =</i>	rIHCSERVMT tuple.
<i>Dump =</i>	rIHCSERVMT tuple.

Table 5 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The maximum number of rISTQDB tuple that may be handling calls from supporting queues, as indicated by the mxnumsup attribute, is not equal to the product of the num_login and mxpcntsup attributes divided by 100.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The value in rIHCSERVMT tuple attribute (mxnumsup) is replaced with the calculated value obtained by the audit (mxnumsup).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIHCSERVMT tuple.
<i>Bad Data =</i>	rIHCSERVMT tuple attribute (mxnumsup).
<i>Good Data =</i>	Calculated value of mxnumsup.
<i>Logical Index =</i>	rIHCSERVMT tuple.
<i>Dump =</i>	rIHCSERVMT tuple.

Table 6 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static tuple rISTINFO does not exist for the dynamic tuple rIHCSERVMT.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None
Manual Action Required:	Action must be taken to insert the missing static rISTINFO tuple with the logical key of rIHCSERVMT tuple.
Dump Description:	
<i>Error Address =</i>	Address of rIHCSERVMT tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIHCSERVMT tuple.
<i>Dump =</i>	rIHCSERVMT tuple.

Table 7 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The number of rISTQDB tuples linked to the busy list of this rIHCSERVMT tuple is inconsistent with the value in the num_busy attribute.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The value in the num_busy attribute is replaced with the calculated value (busy_count).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIHCSERVMT tuple.
<i>Bad Data =</i>	rIHCSERVMT tuple attribute num_busy.
<i>Good Data =</i>	Calculated busy count (busy_count).
<i>Logical Index =</i>	rIHCSERVMT tuple.
<i>Dump =</i>	rIHCSERVMT tuple.

Table 8 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The dynamic data for this tuple in rIHCSERVMT is corrupted. It is compared against the static data in rISTINFO tuple attributes (intra_alwd and mxpcntsup).
Possible Error Effect:	Varies according to conditions.

Corrective Action Taken:	The audit will replace the incorrect data with valid data from the rISTINFO tuple attributes (intra_alwd and mxpcntsup).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rHCSESVTM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rHCSESVTM tuple.
<i>Dump</i> =	rHCSESVTM tuple.

Table 9 Error Code: SPLOSTUP

Error Code:	SPLOSTUP
Error Description:	The static rINTERFLW tuple does not exist for the dynamic rHCSESVTM tuple.
Possible Error Effect:	The data in rHCSESVTM is based upon data in rINTERFLW, but the rINTERFLW tuple doesn't exist when it should.
Corrective Action Taken:	None
Manual Action Required:	Action must be taken to insert the missing static rINTERFLW tuple with the logical key of rHCSESVTM tuple.
Dump Description:	
<i>Error Address</i> =	Address of rHCSESVTM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rHCSESVTM tuple.
<i>Dump</i> =	rHCSESVTM tuple.

Table 10 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The dynamic data for this tuple in rHCSESVTM is corrupted. It is compared against the static data in rINTERFLW tuple attributes (if_on and rr_on).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will replace the incorrect data with valid data from the rINTERFLW tuple attributes (if_on and rr_on).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rHCSESVTM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rHCSESVTM tuple.
<i>Dump</i> =	rHCSESVTM tuple.

HCSRVQ

Software Release: 5E11 and later

1. HEAD CELLS OF SERVING TEAM QUEUE (HCSRVQ) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. Each serving team queue block whether it is unoccupied or being used by an agent is associated with the data in the RLHC_SRVQ (head cells of serving team queues) relation. Data in this relation is audited by the HCSRVQ audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLHC_SRVQ relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The audit found that the value for the rLHC_SRVQ count of the rLHC_CLLQ cwait attributes in the call queues associated with this serving team, cw_lamp, is incorrect.
Possible Error Effect:	This serving team's call wait lamp may not function properly.
Corrective Action Taken:	The rLHC_SRVQ cw_lamp field will be corrected to the value calculated by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	N/A
<i>Bad Data =</i>	The tuple's count of rLHC_CLLQ cwait attributes (rLHC_SRVQ.cw_lamp).
<i>Good Data =</i>	The audit calculated count.
<i>Logical Index =</i>	The logical key of the rLHC_SRVQ tuple.
<i>Dump 1 =</i>	The rLHC_SRVQ tuple.
<i>Dump 2 =</i>	The rLCQMASK tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the rLHC_SRVQ number of busy serving team positions count, num_busy, value is incorrect.
Possible Error Effect:	This serving team's call processing may not function properly.
Corrective Action Taken:	The rLHC_SRVQ num_busy field will be corrected to the value calculated by this audit.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. This distinguishes this COUNTS error from others.
<i>Bad Data</i> =	The tuple's number of busy positions count (rIHC_SRVQ.num_busy).
<i>Good Data</i> =	The number of busy positions calculated by the audit.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the rIHC_SRVQ number of supporting queues count, num_sup, value is incorrect.
Possible Error Effect:	This serving team's call processing may not function properly.
Corrective Action Taken:	The rIHC_SRVQ num_sup field will be corrected to the value calculated by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1. This distinguishes this COUNTS error from others.
<i>Bad Data</i> =	The tuple's number of supporting queues count (rIHC_SRVQ.num_sup).
<i>Good Data</i> =	The number of supporting queue calls being served calculated by the audit.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 4 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the rIHC_SRVQ number of agents logged in count, num_login, value is incorrect.
Possible Error Effect:	This serving team's call processing may not function properly.
Corrective Action Taken:	The rIHC_SRVQ num_login field will be corrected to the value calculated by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	2. This distinguishes this COUNTS error from others.
<i>Bad Data</i> =	The tuple's number of logged in count (rIHC_SRVQ.num_login).
<i>Good Data</i> =	The number of logged in count calculated by the audit.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 5 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the rIHC_SRVQ maximum number of team positions allowed to handle calls from supporting queues count, mxnumsup, value is incorrect.
Possible Error Effect:	This serving team's supporting queue call processing may not function properly.
Corrective Action Taken:	The rIHC_SRVQ mxnumsup field will be corrected to the value calculated by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	3. This distinguishes this COUNTS error from others.
<i>Bad Data</i> =	The tuple's maximum number of supporting queue calls (rIHC_SRVQ.mxnumsup).
<i>Good Data</i> =	The audit calculated count.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 6 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the value for the count of call queues associated with principle serving team, cq_count, was inconsistent between the dynamic rIHC_SRVQ tuple and its corresponding static rICQMASK tuple.
Possible Error Effect:	Principle serving team call processing may not function properly.

Corrective Action Taken:	The rIHC_SRVQ cq_count field will be corrected to the rICQMASK cq_count value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	4. This distinguishes this COUNTS error from others.
<i>Bad Data =</i>	The tuple's count of principle serving team's call queues (rIHC_SRVQ.cq_count).
<i>Good Data =</i>	The rICQMASK count of principle serving team's call queues (rICQMASK.cq_count).
<i>Logical Index =</i>	The logical key of the rIHC_SRVQ tuple.
<i>Dump 1 =</i>	The rIHC_SRVQ tuple.
<i>Dump 2 =</i>	The rICQMASK tuple.

Table 7 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found that the value for the count of call queues associated with the supporting serving team, scq_count, was inconsistent between the dynamic rIHC_SRVQ tuple and its corresponding static rICQMASK tuple.
Possible Error Effect:	Supporting serving team call processing may not function properly.
Corrective Action Taken:	The rIHC_SRVQ scq_count field will be corrected to the rICQMASK cq_count value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	5. This distinguishes this COUNTS error from others.
<i>Bad Data =</i>	The tuple's count of supporting serving team's call queues (rIHC_SRVQ.scq_count).
<i>Good Data =</i>	The rICQMASK count of supporting serving team's call queues (rICQMASK.cq_count).
<i>Logical Index =</i>	The logical key of the rIHC_SRVQ tuple.
<i>Dump 1 =</i>	The rIHC_SRVQ tuple.
<i>Dump 2 =</i>	The rICQMASK tuple.

Table 8 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The audit found that the rIHC_SRVQ tuple corresponding to the unoccupied positions head cell has a non-zero count value for either the count of call queues associated with principle serving team, cq_count or the count of call queues associated with the supporting serving team, scq_count, but this tuple's supporting queue number field, supq, was not zero. These counts should always be zero, since they are associated with the unoccupied positions list.
Possible Error Effect:	This may cause call processing resources to be used incorrectly.
Corrective Action Taken:	The audit will set the rIHC_SRVQ tuple's cq_count and scq_count fields to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIHC_SRVQ tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key of the rIHC_SRVQ tuple.
<i>Dump =</i>	The rIHC_SRVQ tuple.

Table 9 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit found the static principle serving team rICQMASK (call queue bit masks) tuple missing.
Possible Error Effect:	The audit cannot correct the count, so the principle serving team call processing may not function properly.
Corrective Action Taken:	None.
Manual Action Required:	The missing static rICQMASK tuple must be inserted for the principle serving team. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.

Dump Description:	
<i>Error Address</i> =	0. This distinguishes this ODDLOSTTUP error from others.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rICQMASK tuple.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 10 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit found the static supporting serving team rICQMASK (call queue bit masks) tuple missing.
Possible Error Effect:	The audit cannot correct the count, so the supporting serving team call processing may not function properly.
Corrective Action Taken:	None.
Manual Action Required:	The missing static rICQMASK tuple must be inserted for the supporting serving team. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.
Dump Description:	
<i>Error Address</i> =	1. This distinguishes this ODDLOSTTUP error from others.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rICQMASK tuple.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 11 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit found the static rISTPARM (serving team parameters) tuple missing.
Possible Error Effect:	The audit cannot recover the rIHC_SRVQ intra_alwd and mxpcntsup fields due to the missing rISTPARM tuple, so the supporting serving team call processing may not function properly.
Corrective Action Taken:	None.
Manual Action Required:	The missing static rISTPARM tuple must be inserted for the serving team. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.
Dump Description:	
<i>Error Address</i> =	2. This distinguishes this ODDLOSTTUP error from others.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rISTPARM tuple.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 12 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit found the static principle serving team rICQMASK (call queue bit masks) tuple missing.
Possible Error Effect:	The audit cannot recover the count, so the principle serving team call processing may not function properly.
Corrective Action Taken:	None.
Manual Action Required:	The missing static rICQMASK tuple must be inserted for the principle serving team. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.
Dump Description:	
<i>Error Address</i> =	3. This distinguishes this ODDLOSTTUP error from others.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rICQMASK tuple.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 13 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP

Error Description:	The audit found the static supporting serving team rICQMASK (call queue bit masks) tuple missing.
Possible Error Effect:	The audit cannot recover the count, so the supporting serving team call processing may not function properly.
Corrective Action Taken:	None.
Manual Action Required:	The missing static rICQMASK tuple must be inserted for the supporting serving team. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.
Dump Description:	
<i>Error Address</i> =	4. This distinguishes this ODDLOSTTUP error from others.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rICQMASK tuple.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump</i> =	The rIHC_SRVQ tuple.

Table 14 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The audit found an inconsistency between the dynamic rIHC_SRVQ tuple and the static rISTPARM tuple.
Possible Error Effect:	This may result in call processing not functioning properly.
Corrective Action Taken:	The rIHC_SRVQ tuple is recovered based on the corresponding static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIHC_SRVQ tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rIHC_SRVQ tuple.
<i>Dump 1</i> =	The rIHC_SRVQ tuple.
<i>Dump 2</i> =	The rISTPARM tuple.

HLSC

Software Release: 5E10 and later

1. HIGH-LEVEL SERVICE CIRCUIT (HLSC) AUDIT

Line units (both models LU 1 and LU 2) contain from four to six high-level service circuits that serve 64 channel circuits each. The integrated services line unit Z_card (ISLU) contains six high-level service circuit that service up to 512 analog line cards. The high-level service circuit performs three major functions:

- Testing for a satisfactory origination.
- Testing for a satisfactory termination.
- Ringing.

The software representing the high-level service circuit contains operational code and status tables such as the long job occupancy map and the short job occupancy map. The long job occupancy map contains one bit per half second of time a process may be on the high-level service circuit. The short job occupancy map contains one bit per 50 milliseconds of time. The high-level service circuit may be time-shared among several processes. The high-level service circuit audit (HLSC) ensures that the high-level service circuits can be allocated properly. The audit checks all the high-level service circuits in each line unit.

This audit is part of the peripheral control audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The HLSC audit checks the dynamic high-level service circuit relation (RLHLSC), the static high-level service circuit relation (RLHS_CKT), and the circuit data relation (RLCKTDATA). Within each tuple of RLHLSC is the channel control block user array (ccbusr), which indicates the channels using the high-level service circuit. The audit obtains information from the circuit data block common area relation (RLCDBCOM) and the channel control block common area relation (RLCCBCOM). The audit also verifies data in several arrays: the long job occupancy map (PChlunit) and the channel control block foreground communication region (pcCCBFG). Two tables, one for ringing high-level service circuits (Hltabl1) and one for non-ringing high-level service circuits (Hltabl2) are checked at the end of the HLSC audit routine. Figure 1 shows these data structures.

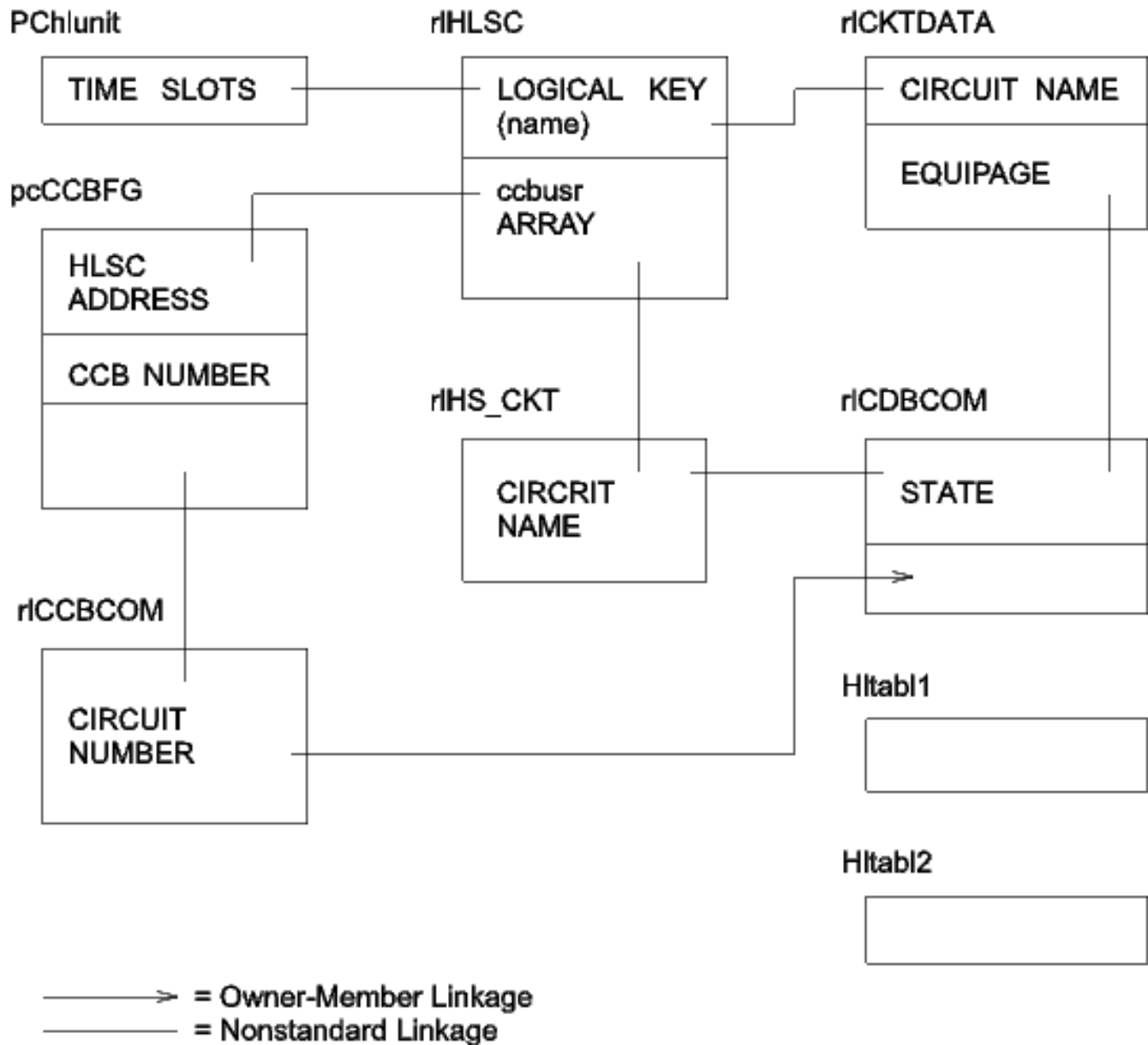


Figure 1 (HLSC-1) Data Structures

3. AUDIT CHECKS

The HLSC audit invokes the dynamic driver (DRDYN) to step through the tuples of RLHLSC and perform the following semantic checks. As the driver steps through rHLSC, it obtains the logical key of each tuple. The audit calculates the line unit to which the tuple (that is, the high-level service circuit represented by the tuple) belongs, and the internal high-level service circuit number.

- (1) Based on the line unit number and internal circuit number, the audit calculates the circuit name. With the circuit name, the audit checks the circuit data relation (rCKTDATA) and ensures that the line unit is equipped with the high-level service circuit (that is, the audit ensures that the circuit exists).

If the audit does not find the appropriate tuple in rCKTDATA, it makes a cross-reference check to rHS_CKT. If the tuple exists in rHS_CKT, the audit reports an error because a tuple of rCKTDATA has been lost. If the

tuple does not exist in rLHS_CKT and rICKTDATA, there should be no dynamic data in the rHLSC tuple.

The tuple in rICKTDATA provides information about the equipage state of the circuit. If the circuit is in the operational or growth state, dynamic data for the tuple should exist in rHLSC.

- (2) The audit checks the circuit name from rICKTDATA against the circuit name of the rHLSC tuple. The same circuit name is used to obtain the related tuple in rCDBCOM. If the circuit name in rHLSC and rICKTDATA do not agree, the audit reports an error and passes the name of the related rCDBCOM tuple to the recovery function. The tuples of rCDBCOM contain status information (in-service and out-of-service) for the service circuits; the circuit status determines the recovery actions the audit takes.
- (3) If the circuit names in rHLSC and rICKTDATA agree, the audit obtains the circuit status from rCDBCOM.

- (a) If the circuit is out of service, the related PChlunit array entries should all be busy. If they are not, the audit invokes the recovery function and reports the error.

If the entries in PChlunit are busy, the audit checks the rHLSC tuple and ensures that all the slots in the ccbusr array are on the out-of-service list. The audit considers slots not on the out-of-service list errors and invokes the recovery function.

- (b) If the high-level service circuit is in service, the audit checks each slot in the PChlunit array. If the entry is set, there must be a corresponding entry in the ccbusr array within rHLSC that is not null. The audit examines the pcCCBFG communication region entry for the tuple of rCCBCOM using the high-level service circuit based on the ccbusr array entry. The audit performs a point-to/point-back check from rHLSC to the channel control block foreground communication region for processes time-sharing the high-level service circuit.

If the entry is not set, the audit tests the slot in the ccbusr array for consistency.

- (4) After all rHLSC tuples have been audited, the audit interfaces with the Peripheral Control subsystem (PC) by calling a function to check the Hltabl1 and Hltabl2 tables. If the PC function detects an error, it returns a failure to the audit and corrects the data itself. The failure triggers a GRPINFO audit error.

4. ERROR HANDLING

Error recovery depends on whether the circuit is in or out of service. The audit makes the determination based on the information in rCDBCOM.

If the circuit is out of service, the audit makes the PChlunit array busy and adds each slot in the ccbusr array to the out-of-service list. The audit then quarantines the rCDBCOM tuple.

If the circuit is in service, the audit initializes PChlunit. It quarantines all related slots of the ccbusr array if they are set and if they pass a point-to/point-back check between rHLSC and the channel control block foreground communication region. If the tuples of the ccbusr array are not set, the audit initializes them.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The high level service circuit is out-of-service but the long job occupancy map is not marked busy.

Possible Error Effect:	The high level service circuit is in an inconsistent state and not totally out-of-service.
Corrective Action Taken:	The audit zeroes the long job occupancy map for the high level service circuit in error. All rICBCOM tuples associated with the high level service circuit are quarantined. The ccbusr array is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rHLSC tuple.
<i>Dump</i>	rHLSC tuple high level service circuit number. Long job occupancy map for the line unit.

Table 2 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The high level service circuit is marked traffic busy in the long job occupancy map, but the rICBCOM user array is not active indicating that an rICBCOM is not using the high level service circuit on a call.
Possible Error Effect:	A high level service circuit is incorrectly activated on a call.
Corrective Action Taken:	The audit zeroes the long job occupancy map for the high level service circuit in error. All rICBCOM tuples associated with the high level service circuit are quarantined. The ccbusr array is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rHLSC tuple.
<i>Dump =</i>	rHLSC tuple high level service circuit number. Long job occupancy map for the line unit.

Table 3 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	Garbage has been found in an rHLSC tuple that should be vacant (zeroed out).
Possible Error Effect:	Propagation of errors in PC allocation of HLSC resources.
Corrective Action Taken:	The tuple is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	Garbage in the name field (circuit name).
<i>Good Data =</i>	The circuit name found in rUNITDATA.
<i>Logical Index =</i>	The physical key of the rHLSC tuple.
<i>Dump</i>	rHLSC tuple.

Table 4 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	Garbage has been found in an rHLSC tuple that should be vacant (zeroed out).
Possible Error Effect:	Propagation of errors in PC allocation of HLSC resources.
Corrective Action Taken:	The tuple is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	Garbage in the name field (circuit name).
<i>Good Data =</i>	0.
<i>Logical Index =</i>	0.
<i>Dump</i>	rHLSC tuple.

Table 5 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The HLSC preferred ringing group information tables are inconsistent with

	respect to the HLSC long job occupancy tables. The PChlconfig function audits these tables for all HLSCs over all line units. The recovery of the preferred ringing group tables is done by the PChlconfig function. No additional information is needed for analysis.
Possible Error Effect:	The HLSC resource allocation for long/short jobs may be affected.
Corrective Action Taken:	The PChlconfig function corrects the inconsistency.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump</i>	NA.

Table 6 Error Code: HLSC_CCB

Error Code:	HLSC_CCB
Error Description:	The high level service circuit is marked traffic busy, but the rCCBCOM using this time portion of the high level service circuit fails point-to/point-back checks.
Possible Error Effect:	A high level service circuit is incorrectly activated on a call.
Corrective Action Taken:	The audit zeroes the long job occupancy map for the high level service circuit in error. All rCCBCOM tuples associated with the high level service circuit are quarantined. The ccbusr array is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rHLSC tuple.
<i>Dump =</i>	rHLSC tuple High level service circuit number. Long job occupancy map for the line unit.

Table 7 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The high level service circuit is marked traffic idle in the long job occupancy map, but the rCCBCOM user array is not idle indicating that a rCCBCOM is using the high level service circuit on a call.
Possible Error Effect:	A high level service circuit is incorrectly activated on a call.
Corrective Action Taken:	The audit zeroes the long job occupancy map for the high level service circuit in error. All rCCBCOM tuples associated with the high level service circuit are quarantined. The ccbusr array is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rHLSC tuple.
<i>Dump =</i>	rHLSC tuple High level service circuit number. Long job occupancy map for the line unit.

Table 8 Error Code: KEY

Error Code:	KEY
Error Description:	The rHLSC tuple circuit name is not the same as the rHS_CKT tuple circuit name.
Possible Error Effect:	Erroneous job allocation for the HLSC.
Corrective Action Taken:	The circuit name is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHLSC tuple.
<i>Bad Data =</i>	rHLSC circuit name.
<i>Good Data =</i>	rHS_CKT circuit name.
<i>Logical Index =</i>	DA key calculated for the rHLSC tuple.

<i>Dump</i>	rHLSC tuple.

Table 9 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The high level service circuit is out-of-service but there is an active call using the circuit.
Possible Error Effect:	An out-of-service circuit is in an inconsistent state and is being used by call processing when it shouldn't.
Corrective Action Taken:	The audit zeroes the long job occupancy map for the high level service circuit in error. All rICCB COM tuples associated with the high level service circuit are quarantined. The ccbusr array is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rHLSC tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rHLSC tuple.
<i>Dump</i> =	rHLSC tuple high level service circuit number. Long job occupancy map for the line unit.

Table 10 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	The rICKTDATA tuple for the HLSC shows the circuit unequipped. There should be no dynamic data for this circuit, however the audit found garbage in the rHLSC tuple.
Possible Error Effect:	Errors may be propagated to PC for HLSC job allocation.
Corrective Action Taken:	The tuple is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rHLSC tuple.
<i>Bad Data</i> =	Contents in the key field of the rHLSC tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	DA key calculated for the rHLSC tuple.
<i>Dump</i>	NA.

HTLSEQN

Software Release: 5E10 and later

1. AUTOQUOTE HOTEL/MOTEL (HTLSEQN) AUDIT

This audit verifies the data in the RLHTL_SEQN relation. It checks the key and seq_num fields.

The RLHTL_SEQN relation is built from the static relation RLFC_HOTEL. This audit accesses each tuple in RLFC_HOTEL. For every auto-quote hotel identifier (aqhid), it checks to make sure there exists a rHTL_SEQN tuple. The audit then checks to see if the key is correct; it should be equal to the aqhid+1. If it is not, then an error is reported and the key is corrected. Next the sequence number field (seq_num) of the rHTL_SEQN relation is checked to make sure that it is in the valid range (1-999). If it is invalid, an error is reported and the field is set to 1.

2. DATA STRUCTURES

Figure 1 illustrates the basic audit structure.

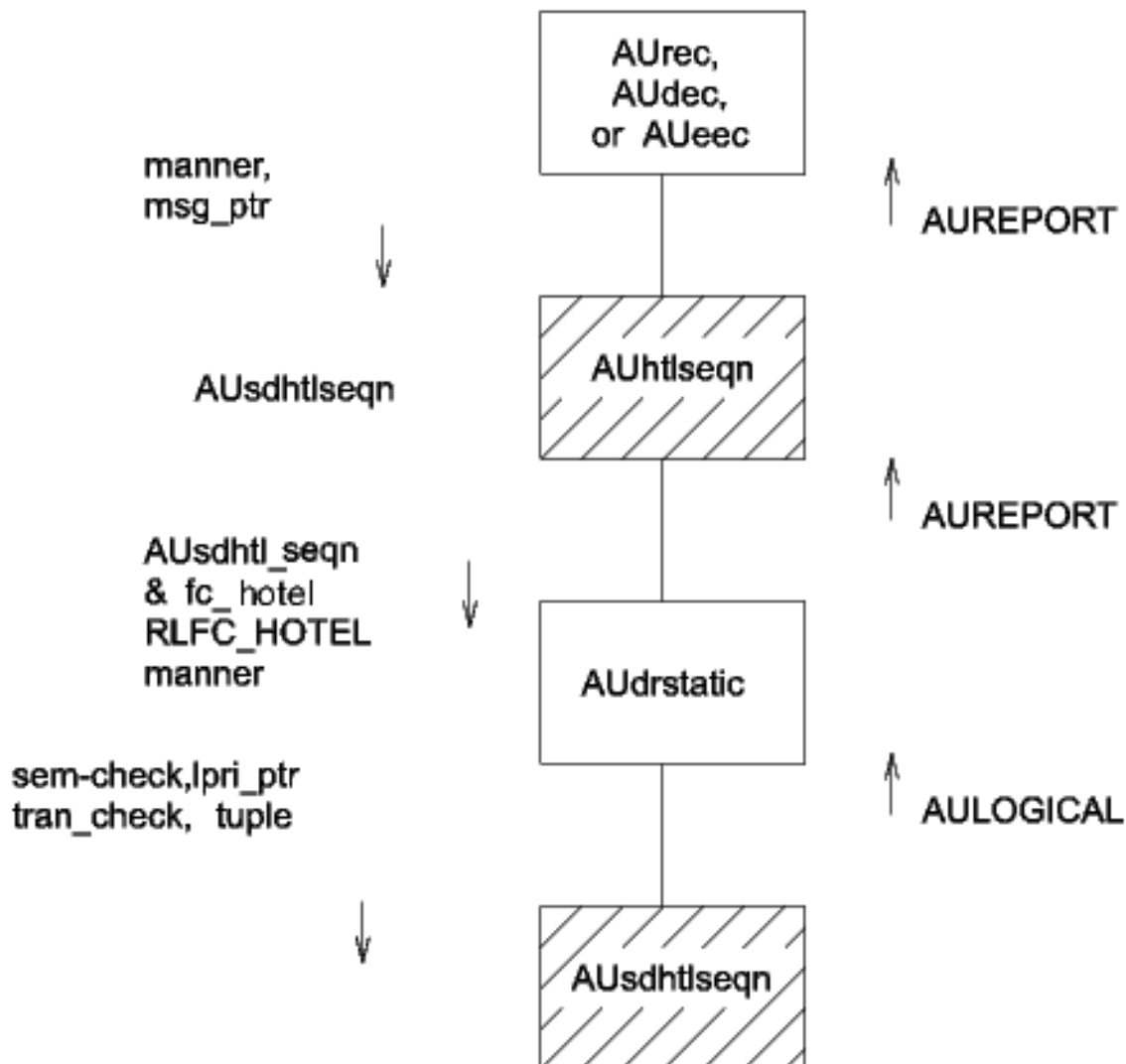


Figure 1 (HTLSEQN-1) HTLSEQN Audit Structure Chart

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The HTLSEQN audit calls the DRSTATIC driver. The driver calls AU\$dhtl_seqn() to perform the semantic checks.

When a lost tuple is found, a new tuple is created and inserted into the RLHTL_SEQN relation. When other errors are found, the error is reported and the error is corrected if possible. If not, the tuple is zeroed out and reinitialized.

The HTLSEQN audit uses the following error codes, and also all those used by the driver, AUdrstatic.

Table 1 Error Code: KEY

Error Code:	KEY
Error Description:	The hotelid field in rHTL_SEQN is not the correct value (aqhid+1).
Possible Error Effect:	None.
Corrective Action Taken:	The rHTL_SEQN tuple will be updated with the hotelid field set to the aqhid field + 1 of the FC_HOTEL relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHTL_SEQN.
<i>Bad Data =</i>	hotelid field of rHTLSEQN.
<i>Good Data =</i>	aqhid field + 1 of rFC_HOTEL.
<i>Logical Index =</i>	rHTL_SEQN.
<i>Dump =</i>	Contents of the rHTL_SEQN tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rHTL_SEQN does not exist for a port when it should.
Possible Error Effect:	This will cause lost resources.
Corrective Action Taken:	The rHTL_SEQN tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHTL_SEQN.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rHTL_SEQN (aqhid + 1).
<i>Dump =</i>	NA.

Table 3 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The seq_numb field in rHTL_SEQN is not within the range 1 - 999.
Possible Error Effect:	This will cause a bad sequence number to be used.
Corrective Action Taken:	The rHTL_SEQN tuple will be updated with the seq_numb field set to 1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHTL_SEQN.
<i>Bad Data =</i>	seq_numb field of rHTLSEQN.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	rHTL_SEQN.
<i>Dump =</i>	Contents of the rHTL_SEQN tuple.

ICLIDIDX

Software Release: 5E10 and later

1. INDIVIDUAL CALLING LINE IDENTIFICATION FOR CALLING NUMBER DELIVERY INDEX (ICLIDIDX) AUDIT

The RLICLIDIDX relation is used by the LASS feature ICLID to index into the RLCNDDATA and RLCNAMDATA (usage sensitive only) relations. The ICLIDIDX audit performs static to dynamic existence checks of the static RLICLIDIDX relation to the dynamic RLCNDDATA and RLCNAMDATA (usage sensitive only) relations. The audit is also scheduled during a selective initialization.

2. DATA STRUCTURES

The RLICLIDIDX relation is a static relation used by the ICLIDIDX audit to perform static to dynamic existence checks of the dynamic RLCNDDATA and RLCNAMDATA (usage sensitive only) data structure.

3. AUDIT CHECKS

The ICLIDIDX audit steps through the RLICLIDIDX relation and verifies that a corresponding dynamic rICNDDATA and rICNAMDATA (usage sensitive only) data tuples exists.

4. ERROR HANDLING

If a rLICLIDIDX tuple exists with no corresponding dynamic rICNDDATA tuple, the audit inserts and initializes the corresponding dynamic tuple. If a rLICLIDIDX tuple for usage sensitive only exists with no corresponding dynamic rICNAMDATA tuple, the audit inserts and initializes the corresponding dynamic tuple.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: KEY

Error Code:	KEY
Error Description:	The audit has detected an inconsistency between the static data in RLICLID_IDX and the dynamic data in RLCNDDATA. The inconsistency indicates that the static data ic_idx field does not match the key field of the corresponding dynamic rICNDDATA tuple.
Possible Error Effect:	An available rICNDDATA tuple may never be used for the ICLID feature.
Corrective Action Taken:	The audit will correct the key field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; itm_idx field of the related static rLICLID_IDX tuple.
<i>Bad Data =</i>	ic_idx of the rICNDDATA tuple (key field).
<i>Good Data =</i>	ic_idx of the related static rLICLID_IDX tuple.
<i>Logical Index =</i>	itm_type of the related static rLICLID_IDX tuple.
<i>Dump =</i>	The dynamic data rICNDDATA tuple; the static data rLICLID_IDX tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit has detected an inconsistency between the static data in RLICLID_IDX and the dynamic data in either RLCNDDATA or RLCNAMDATA. The inconsistency indicates that the static data has a tuple and yet no corresponding dynamic rICNDDATA or rICNAMDATA tuple exists.
Possible Error Effect:	Not enough rICNDDATA or rICNAMDATA tuples may be available for the ICLID feature information.

Corrective Action Taken:	The audit will insert a rICNDDATA or rICNAMDATA tuple and get it initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Relation ID for either RLCNDDATA or RLCNAMDATA.
<i>Good Data =</i>	ic_idx field of rLICLID_IDX (static data) tuple.
<i>Logical Index =</i>	The key of the rICNDDATA tuple that is missing.
<i>Dump =</i>	The static data rLICLID_IDX tuple.

Table 3 Error Code: MEMKEY

Error Code:	MEMKEY
Error Description:	The audit has detected an inconsistency between the static data in RLICLID_IDX and the dynamic data in RLCNAMDATA. The inconsistency indicates that the static data ic_idx field does not match the key field of the corresponding dynamic rICNAMDATA tuple.
Possible Error Effect:	An available rICNAMDATA tuple may never be used for the usage-sensitive ICLID feature.
Corrective Action Taken:	The audit will correct the key field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; itm_idx field of the related static rLICLID_IDX tuple.
<i>Bad Data =</i>	ic_idx of the rICNAMDATA tuple (key field).
<i>Good Data =</i>	ic_idx of the related static rLICLID_IDX tuple.
<i>Logical Index =</i>	itm_type of the related static rLICLID_IDX tuple.
<i>Dump =</i>	The dynamic data rICNAMDATA tuple; the static data rLICLID_IDX tuple.

ICPEST

Software Release: 5E10 and later

1. INTERPROCESSOR CENTRAL PROCESSOR EQUIPMENT STATUS TABLE (ICPEST) AUDIT

The metallic service units (MSUs) in the interface modules provide metallic access, GDX compensation, scanning, and power conversion. Another important function, testing, is handled by the MSUs by connecting test leads to the metallic test interconnect buses (MTIBs). Each MSU has access points for 14 MTIBs. The allocation of MTIBs depends on the software states stored in the central processor equipment status table relation (RLCPEST) and the switch maintenance equipment status table relation (RLSMEST) for proper operation. The equipment status table relations are checked by four audits: the switch maintenance equipment status table audit (SMEST), the administrative module equipment status table audit (CPEST), the circuit data audit (CKTDATA), and ICPEST. Of these, only ICPEST is an interprocessor audit. ICPEST ensures that the maintenance request administrator (MRA) can run diagnostics on the MTIBs by checking the maintenance states of both starting points (the preferred end and the other end) for diagnostics. Diagnostics may only be run from these two starting points. When the MRA diagnoses MTIBs, it starts from one switching module (the preferred end) and sequentially checks the MTIB connections through all the switching modules (SMs) to the last SM. If the preferred end is isolated, the MRA begins diagnosing MTIB connections from the module designated the other end.

The audit is part of the peripheral control audits subgroup. It runs in the communication module processor (CMP) and SM in the segmented mode.

2. DATA STRUCTURES

The audit checks the fields recording the status of the MTIBs and the process identifications in the two equipment status table relations, RLCPEST and RLSMEST, shown in Figure 1. The ICPEST audit also refers to the following data:

Glond = This global parameter contains the preferred SM to start MTIB diagnostics.

Glond = This global parameter contains the other SM to start MTIB diagnostics.

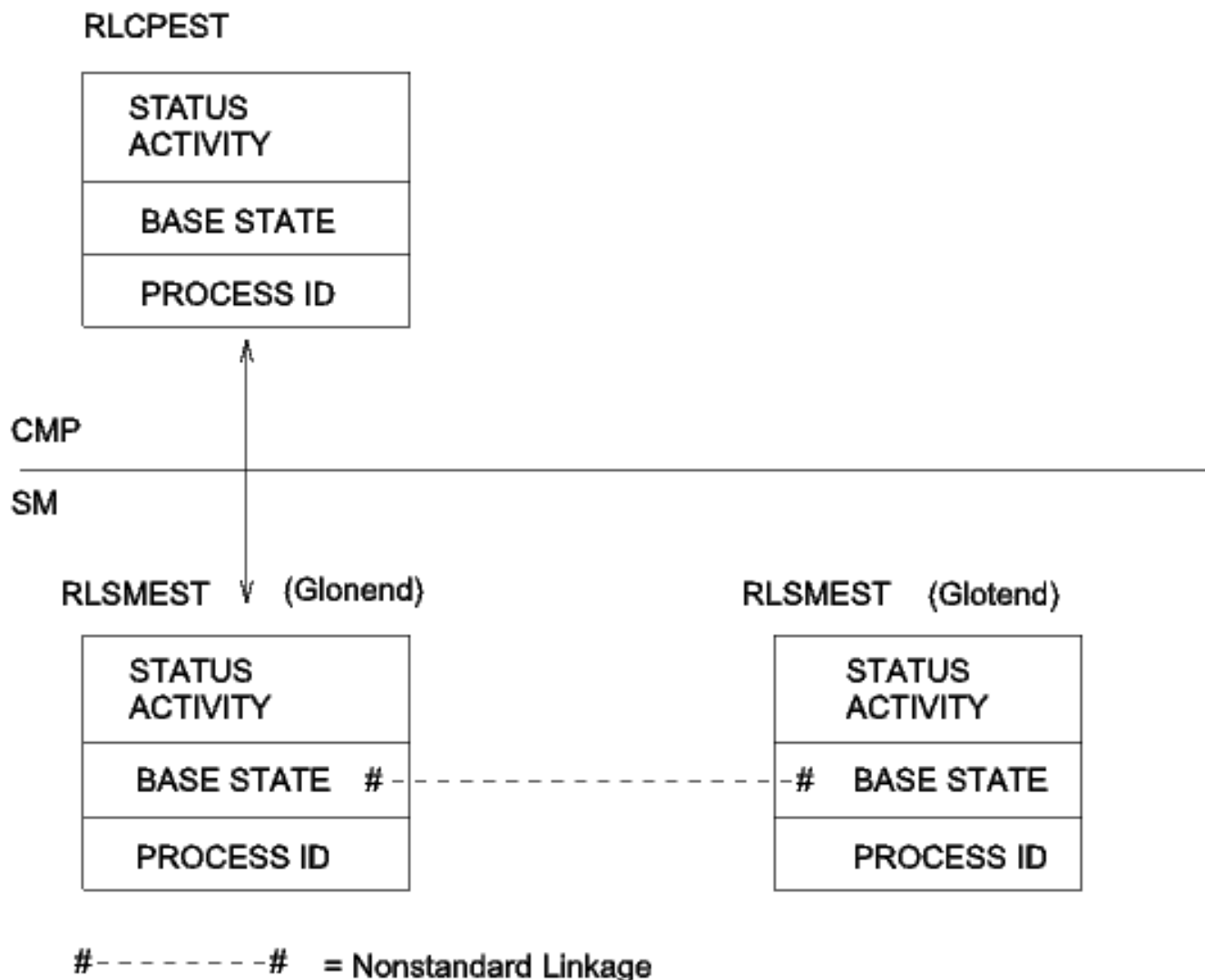


Figure 1 (ICPEST-1) Data Structures of the ICPEST Audit

3. AUDIT CHECKS

The CMP provides the master control of the audit. The master data, the RLSMEST relation, resides in the switching modules. In the CMP the following audit checks are performed:

If the audit has just been entered:

ZAP = Check if the audit has been entered due to a system interrupt.

If not:

BAS_EQPG = Check if the base state (bas_stat) field in the rLCPEST tuple is consistent with the data collected from the other (Glotend) SM.

BAS_STAT = Check if the base state (bas_stat) field is set correctly compared to data collected from the SMs.

BASEQUAL = Check if the base state (bas_stat) field in the rLCPEST tuple is consistent with the data collected

from the preferred (Glonend) SM.

IMPID	= Check if the process identification (tp_pid) field in the rICPEST tuple is consistent with the data collected from the preferred (Glonend) SM.
PID	= Check if the terminal process identification in the rICPEST tuple is null (OSnullpid) for an idle MTIB.
SMEST	= Check if the state activity (stat_act) field in the rICPEST tuple is consistent with data collected from both SMs.
STATUS	= Check if the state activity (stat_act) field in the rICPEST tuple is in a valid state.
TID	= Check if the process identification (tp_pid) field in the rICPEST tuple is consistent with the data collected from the other (Glotend) SM.

In the SM, the following audit checks are performed.

Collect MTIB maintenance status:

CKTDSMEST = Check if the rISMEST tuple is missing for this MTIB.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BASEQUAL

Error Code:	BASEQUAL
Error Description:	The base state field (bas_stat) in the rICPEST tuple in the CMP has been found to be different from the base state field (bas_stat) in the rISMEST tuple in the preferred SM (Glonend) for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rISMEST tuple. The data collected from the rISMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rICPEST tuple.
Possible Error Effect:	The MTIB associated with the rICPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rICPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rICDBCOM and rISMEST tuples associated with this MTIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICPEST tuple found in error.
<i>Bad Data =</i>	The base state (bas_stat) from the rICPEST tuple found in error.
<i>Good Data =</i>	The base state (bas_stat) from the rISMEST tuple in the preferred SM (Glonend).
<i>Logical Index =</i>	Logical key (unit_id) of the rICPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rICPEST tuple found in error.

<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLotend). SM.

Table 2 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The base state field (bas_stat) in the rICPEST tuple in the CMP has been found to be different from the base state field (bas_stat) in the rSMEST tuple in the other SM (Glotend) for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rICPEST tuple.
Possible Error Effect:	The MTIB associated with the rICPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rICPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rCDBCOM and rSMEST tuples associated with this MTIB. The equipment status tables in the AM and both end SM's are idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICPEST tuple found in error.
<i>Bad Data =</i>	The base state (bas_stat) from the rICPEST found in error.
<i>Good Data =</i>	The base state (bas_stat) from the rSMEST tuple in the other SM (Glotend).
<i>Logical Index =</i>	Logical key (unit_id) of the rICPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rICPEST tuple found in error.
<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLotend). SM.

Table 3 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The state activity field (stat_act) from the rICPEST tuple in the CMP and the state activity field (stat_act) from both rSMEST tuples in the preferred (Glonend) and the other (Glotend) SMs are idle. However, base state field (bas_stat) in the rICPEST tuple in the CMP has been found to be different from the base state field (bas_stat) in the rSMEST tuple in the preferred (Glonend) or other (Glotend) SM for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rICPEST tuple.
Possible Error Effect:	The MTIB associated with the rICPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rICPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a

	message will be sent to both SMs to recover the rICDBCOM and rSMEST tuples associated with this MTIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICPEST tuple found in error.
<i>Bad Data =</i>	The base state (bas_stat) from the rICPEST tuple found in error.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (unit_id) of the rICPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rICPEST tuple found in error.
<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLotend). SM.

Table 4 Error Code: CKTDSMEST

Error Code:	CKTDSMEST
Error Description:	The global parameter indicating the preferred SM (Glonend) or the other SM (Glotend) in the CMP indicate that MTIBs exist on this SM, however, the rSMEST dynamic tuple is not in the data base in the SM. This error indicates that either the global parameters are corrupt or the rSMEST tuple is erroneously missing from the database.
Possible Error Effect:	The MTIB associated with the rICPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The CKTDATA audit will be schedule to insert the rSMEST tuple into the database.
Manual Action Required:	If this error occurs over and over it could be an ODD inconsistency between the RLCKDTATA relation in the SM and the global parameters specifying the preferred SM (Glonend) and the other SM (Glotend) located in the CMP. If this SM contains an MSU: • Check the RLMUMTIB relation to find out if this SM is one of the end SMs. • If this SM is not an end SM, then the global parameters must be updated according to the RLMUMTIB relation. • If this is an end SM, then a tuple must be inserted into the RLCKTDATA relation for all this MTIB. If the SM does not contain an MSU, update the global parameters according to the RLMUMTIB relation.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple buffer.
<i>Bad Data =</i>	Circuit name of the MTIB found missing in the database.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 5 Error Code: IMPID

Error Code:	IMPID
Error Description:	The process identification field (tp_pid) from the rICPEST tuple in the CMP has been found to be different from the process identification field (tp_pid) from the rSMEST tuple in the preferred SM (Glonend) for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the

	rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rCPEST tuple.
Possible Error Effect:	The MTIB associated with the rCPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rCDBCOM and rSMEST tuples associated with this MTIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.
<i>Bad Data =</i>	The process identification field (tp_pid) from the rCPEST tuple found in error.
<i>Good Data =</i>	The process identification field (tp_pid) from the rSMEST tuple in the preferred SM (Glonend).
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rCPEST tuple found in error.
<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (Glonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (Glotend). SM.

Table 6 Error Code: PID

Error Code:	PID
Error Description:	The state activity field (stat_act) from the rCPEST tuple in the CMP and the state activity field (stat_act) from both rSMEST tuples in the preferred (Glonend) and the other (Glotend) SMs are idle. However, process identification field (tp_pid) in the rCPEST tuple in the CMP has been found to be different from the process identification field (tp_pid) in the rSMEST tuple in the preferred (Glonend) or other (Glotend) SM for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rCPEST tuple.
Possible Error Effect:	The MTIB associated with the rCPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rCDBCOM and rSMEST tuples associated with this MTIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.
<i>Bad Data =</i>	The process identification field (tp_pid) from the rCPEST tuple found in error.
<i>Good Data =</i>	Null process identification (OSnullpid).
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rCPEST tuple found in error.
<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (Glonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (Glotend). SM.

Table 7 Error Code: SMEST

Error Code:	SMEST
Error Description:	The state activity field (stat_act) from the rCPEST tuple in the CMP has been found to be idle (DBIDLE), but the data collected from the SMs indicate that state activity field (stat_act) in the rSMEST tuple in the preferred SM (Glonend) or other SM (Glotend) is busy (DBBUSY). For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rCPEST tuple.
Possible Error Effect:	The MTIB associated with the rCPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rCDBCOM and rSMEST tuples associated with this MTIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.
<i>Bad Data =</i>	The state activity field (stat_act) from the rCPEST tuple found in error.
<i>Good Data =</i>	The state activity field (stat_act) from the rSMEST tuple in the preferred SM (Glonend).
<i>Logical Index =</i>	Logical key (unit_id) of the rCPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rCPEST tuple found in error.
<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (Glonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (Glotend). SM.

Table 8 Error Code: STATUS

Error Code:	STATUS
Error Description:	The state activity field (stat_act) in the rCPEST tuple in the CMP has been found to be different from the base state field (bas_stat) in the rSMEST tuple in the preferred SM (Glonend) or the other SM (Glotend) for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rCPEST tuple.
Possible Error Effect:	The MTIB associated with the rCPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rCPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rCDBCOM and rSMEST tuples associated with this MTIB. P
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCPEST tuple found in error.
<i>Bad Data =</i>	The state activity field (stat_act) from the rCPEST tuple found in error.
<i>Good Data =</i>	The state activity field (stat_act) from the rSMEST tuple in the preferred SM (Glonend).
<i>Logical Index =</i>	Key of the rCPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rCPEST tuple found in error.

<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLotend). SM.

Table 9 Error Code: TID

Error Code:	TID
Error Description:	The process identification field (tp_pid) in the rICPEST tuple in the CMP has been found to be different from the process identification field (tp_pid) in the rSMEST tuple in the other SM (Glotend) for this MTIB. For both the preferred SM (Glonend) and the other SM (Glotend), a message is sent from the CMP to the SM to collect maintenance status for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple. The data collected from the rSMEST tuple is the base state (bas_stat), the state activity (stat_act), and the terminal process identification (tp_pid). This data is then sent to the CMP and compared to data stored in the rICPEST tuple.
Possible Error Effect:	The MTIB associated with the rICPEST tuple may not be removed or restored. Also, the MTIB will not be allocated to perform metallic testing.
Corrective Action Taken:	The rICPEST tuple will be recovered to an in-service state, the corresponding rIMPHDATA tuple in the CMP will be quarantined (quar = AUAUSET), and a message will be sent to both SMs to recover the rCDBCOM and rSMEST tuples associated with this MTIB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICPEST tuple found in error.
<i>Bad Data =</i>	The process identification field from the rICPEST tuple found in error.
<i>Good Data =</i>	The process identification field from the rSMEST tuple in the other SM (Glotend).
<i>Logical Index =</i>	Key of the rICPEST tuple found in error (MTIB circuit name).
<i>Dump 1 =</i>	rICPEST tuple found in error.
<i>Dump 2 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLonend). SM.
<i>Dump 3 =</i>	Status activity (stat_act), base state (bas_stat), and the terminal process identification (tp_pid) from the preferred SM (GLotend). SM.

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after taking a system interrupt (such as, write protect, out-of-range read, and so forth).
Possible Error Effect:	May cause further resets, audits, and or asserts.
Corrective Action Taken:	None, all data that would have caused the reset is reinitialized on the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The SM believed to the have a connection to the MTIB (rICPEST tuple under audit).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

ICR

Software Release: 5E10 and later

1. INTERCOM CALLING RESTRICTION (ICR) AUDIT

The purpose of ICR audit is to perform static-to-dynamic existence checks between RLICR and RLICRST, or RLICRLGST. Any missing tuples in the dynamic relations will be inserted by this audit.

2. DATA STRUCTURES

The RLICR relation is a static, indexed SM relation with group id as key. It is group partitioned, that is, for SMs containing the member of this SFG, one rLICR tuple is populated. This relation indicates the maximum number of simultaneous calls available for this SFG, AMA, and SFG busy lamp information, whether the SFG is a large or a small group and which SM the SFG is tracked.

The ICR audit will be located at AUICR module as part of BASE package. It runs in the segmented mode and operates in the OSDS-M environment.

If the rLICR tuple is located at the tracking SM, ICR audit will perform the existence check between the static relation RLICR and the dynamic relations RLICRST or RLICRLGST depending on the size of this SFG. If the maximum number of simultaneous calls is greater than 96 (large group), the corresponding rLICRLGST tuple must be existed. Otherwise, the rLICRST tuple should be populated at this SM. Any missing rLICRST or rLICRLGST tuple is inserted by this function.

If rLICRST or rLICRLGST tuple should be existed in the tracking SM and it is not, then an attempt will be made to insert the missing tuple via "DBfintup" and an error message is printed. An assert is fired if data base insert fails.

3. AUDIT CHECKS

ICR audit steps through the static relation RLICR one tuple at a time. For each tuple populated at the tracking SM, this audit verifies that the corresponding tuple is existing at either the relation RLICRST or RLICRLGST depending upon the size of this SFG group. If the corresponding tuple is not found, the audit inserts it and prints an error message on the ROP.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the dynamic RLICRST or RLICRLGST relation; the corresponding static tuple is in RLICR relation.
Possible Error Effect:	Application codes will ASSERT and calls originated for this ICR group will be rejected.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLICR tuple.
<i>Bad Data =</i>	0--Missing rLICRST, 1--Missing rLICRLGST.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of lost tuple.
<i>Dump =</i>	Dump of the rLICR tuple.

ICRLGST

Software Release: 5E10 and later

1. INTERCOM CALLING RESTRICTION LARGE STATUS (ICRLGST) AUDIT

The purpose of this audit is to perform dynamic-to-static existence checks between RLICRLGST and RLICR. Any extra RLICRLGST tuples are deleted. This audit also performs the key check and validates the RLICRLGST data.

2. DATA STRUCTURES

The RLICRLGST relation is a dynamic, indexed switching module (SM) relation with a group number as key. It contains the busy count maximum, busy/idle bit maps for call processing and keep alive status, current busy count, AMA billing flag and busy lamp status. If the SM is the tracking SM for this SFG and the maximum number of simultaneous calls is greater than 96, one RLICRLGST tuple should be populated at this SM.

The ICRLGST audit keep alive function performs the keep alive auditing on the busy/idle bits in the RLICRLGST relation. (Refer to Table 1.) If the busy_idle bit indicates the member is idle, this is a "don't care" situation for the keep alive and the audit proceeds to the next busy_idle bit.

If the busy_idle bit is busy, the audit examines the keep alive bit for that member. If the keep alive bit is set to 0, this is a safe state and the audit will set the keep alive bit to 1. If the keep alive bit is 1, this means that the keep alive has visited this member for longer than two keep alive audit cycles. The keep alive audit assumes that this member is stuck busy and will idle the busy_idle bit for this member and adjust the busy count. (Refer to Figure 1.)

Table 1 ICRLGST AUDIT KEEP ALIVE CALL FUNCTION

Keep Alive Bit	Call Processing Bit	Keep Alive Function Action
0	Busy	Set keep alive bit to 1
0	Idle	Don't care
1	Busy	Set keep alive bit to 0 + call processing bit to idle
1	Idle	Don't care

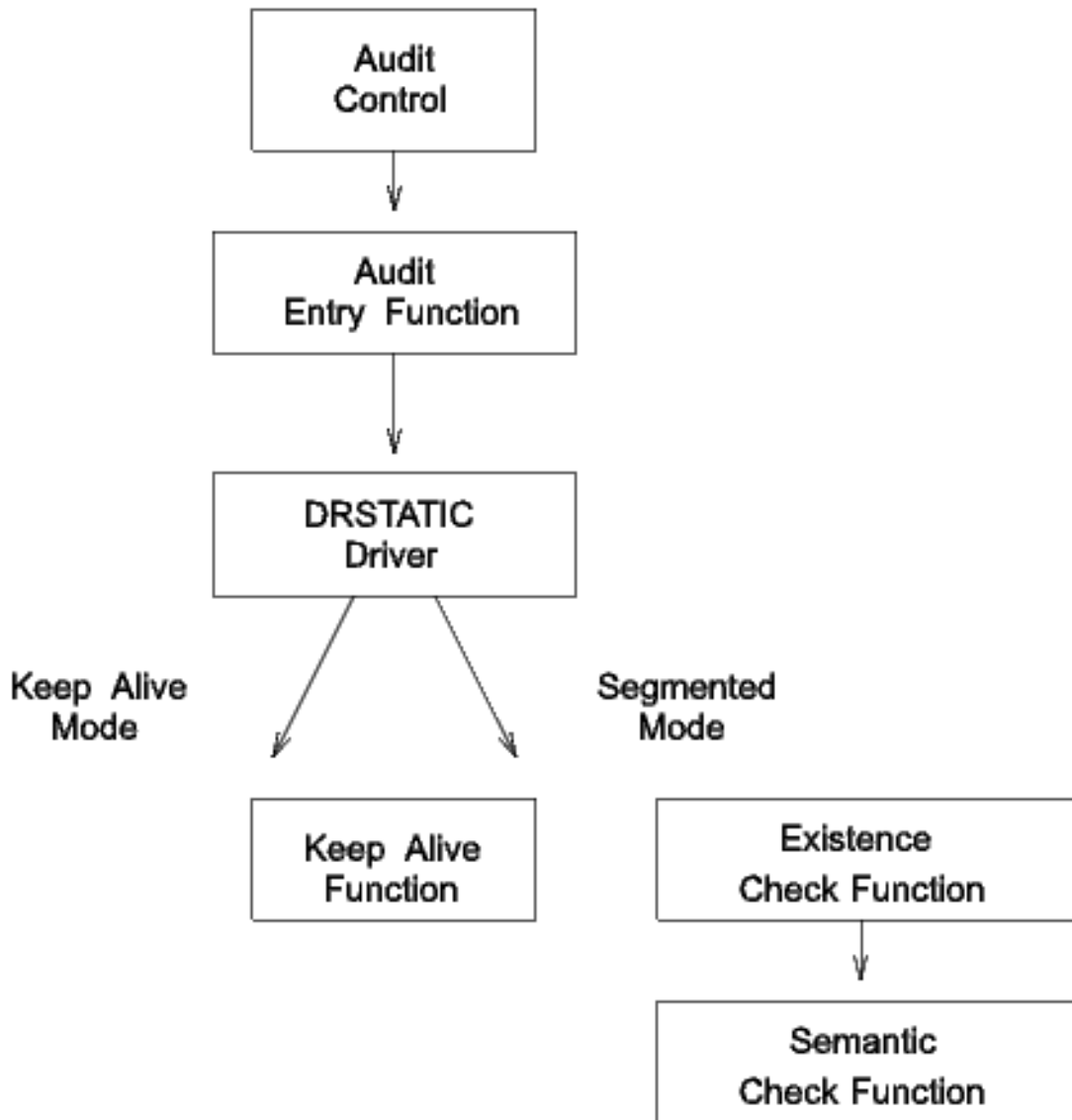


Figure 1 (ICRLGST-1) Existence Checks Processing

3. AUDIT CHECKS

The ICRLGST audit is located at AUicr module as part of the BASE package. The ICRLGST audit steps through the dynamic relation RLICRST and RLICRLGST respectively one tuple at a time. For each tuple, the following checks are performed:

- (1) Perform dynamic-to-static existence check. For each tuple, verify that the corresponding tuple is existing at the static relation rLICR for the tracking SM. If it is not, this tuple is deleted and an error message is printed on the ROP.
- (2) Verify the consistency between rLICR and rLICRLGST regarding size, AMA billing status and the type of ICR busy lamp. The ICR busy lamp could be associated with the analog or attendant console. If there is any inconsistency, the dynamic data is corrected to be consistent with the static data.
- (3) It verifies that current busy counts are the same as those indicated by busy/idle bit map. If there is any inconsistency, the current busy count is reset.

- (4) If this SFG has an analog or attendant console equipped with busy lamp, the busy lamp will be turned on/off according to the current busy count.

If any inconsistency is found by this semantic function, the dynamic data in the rLICRLGST tuple is corrected and the ICRLGST audit continues execution without returning FALSE to the driver, which causes the driver to take a time break.

4. ERROR HANDLING

If no tuple is found in RLICR, and RLICRLGST indicates that it exists, an attempt is made to delete the extra tuple and an error message is printed. Audit control also schedules the related audit to be run at the elevated mode after the ICRLGST audit is completed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The busy lamp type indicated by rLICRLGST dynamic tuple is inconsistent with that specified in the static rLICR tuple.
Possible Error Effect:	The attendant console may receive erroneous lamp treatment.
Corrective Action Taken:	Correct busy lamp type of rLICRLGST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.
<i>Bad Data =</i>	Busy lamp type at rLICRLGST.
<i>Good Data =</i>	Busy lamp type at rLICR.
<i>Logical Index =</i>	Key of rLICRLGST tuple.
<i>Dump =</i>	rLICRLGST tuple.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The busy count does not agree with that calculated from busy/idle status array.
Possible Error Effect:	Several error effects may happen: • ICR calls may be erroneously rejected. • The attendant console may receive erroneous lamp treatment. • Application code may waste time to search idle member.
Corrective Action Taken:	Correct busy count field to be consistent with busy/idle array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.
<i>Bad Data =</i>	Busy count before correction.
<i>Good Data =</i>	Actual busy count of busy/idle status.
<i>Logical Index =</i>	Key of rLICRLGST tuple.
<i>Dump =</i>	rLICRLGST tuple.

Table 4 Error Code: DDAMABIL

Error Code:	DDAMABIL
Error Description:	The AMA flag indicated by rLICRLGST dynamic tuple is inconsistent with that specified in the static rLICR tuple.
Possible Error Effect:	The AMA report could be erroneously generated or discarded.

Corrective Action Taken:	Correct AMA flag field of rLICRLGST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.
<i>Bad Data =</i>	AMA flag at rLICRLGST.
<i>Good Data =</i>	AMA flag at rLICR.
<i>Logical Index =</i>	Key of rLICRLGST tuple.
<i>Dump =</i>	rLICRLGST tuple.

Table 5 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The rLICRLGST tuple should not be populated at this SM, because the corresponding static rLICR tuple does not exist.
Possible Error Effect:	ICR calls originated from non-tracking SM may be erroneously accepted or rejected.
Corrective Action Taken:	The rLICRLGST tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rLICRLGST tuple.
<i>Dump =</i>	rLICRLGST tuple.

Table 6 Error Code: KEY

Error Code:	KEY
Error Description:	An invalid key value is found in the rLICRLGST tuple.
Possible Error Effect:	The intercom calls of this SFG will be rejected. Assert should be taken from application code.
Corrective Action Taken:	A key error will be reported and corrected by the ICRLGST audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.
<i>Bad Data =</i>	The error key.
<i>Good Data =</i>	Correct key value.
<i>Logical Index =</i>	Key of rLICRLGST tuple.
<i>Dump =</i>	rLICRLGST tuple.

Table 7 Error Code: RQSIZE

Error Code:	RQSIZE
Error Description:	The size indicated by rLICRLGST dynamic tuple is different from that specified in the static rLICR tuple.
Possible Error Effect:	The ICR calls may be erroneously rejected or free calls may be allowed.
Corrective Action Taken:	Correct the size field of rLICRLGST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.
<i>Bad Data =</i>	Size at rLICRLGST.
<i>Good Data =</i>	Size at rLICR.
<i>Logical Index =</i>	Key of rLICRLGST tuple.
<i>Dump =</i>	rLICRLGST tuple.

Table 8 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	The rLICRLGST tuple should not be populated at this SM. Either this SM is not tracking the SM or the size is not a large SFG.
Possible Error Effect:	If rLICRLGST is not populated at the tracking SM, application code will assert and request ICRLGST audit. No ICR mechanism will be implemented.
Corrective Action Taken:	The rLICRLGST tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRLGST tuple.

<i>Bad Data</i> =	The tracking SM of rLICR tuple. The size of rLICRLGST may be wrong. Check the rLICRLGST dump to verify that information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rLICRLGST tuple.
<i>Dump</i> =	rLICRLGST tuple.

ICRST

Software Release: 5E10 and later

1. INTERCOM CALLING RESTRICTION STATUS (ICRST) AUDIT

The purpose of this audit is to perform dynamic-to-static existence checks between RLICRST and RLICR. Any extra rLICRST tuples are deleted. This audit also performs the key check and validates the rLICRST data.

2. DATA STRUCTURES

The RLICRST relation is a dynamic, indexed SM relation with group number as key. It contains the busy count maximum, busy/idle bit maps for call processing and keep alive status, current busy count, AMA billing flag and busy lamp status. If the switching module (SM) is the tracking SM for this SFG and the maximum number of simultaneous calls is not greater than 96, one rLICRST tuple should be populated at this SM.

The ICRST audit keep alive function performs the keep alive auditing on the busy/idle bits in the RLICRST relation. (Refer to Table 1.) If the busy_idle bit indicates the member is idle, this is a "don't care" situation for the keep alive and the audit proceeds to the next busy_idle bit.

If the busy_idle bit is busy, the audit examines the keep alive bit for that member. If the keep alive bit is set to 0, this is a safe state and the audit will set the keep alive bit to 1. If the keep alive bit is 1, this means that the keep alive has visited this member for longer than two keep alive audit cycles. The keep alive audit assumes that this member is stuck busy and will idle the busy_idle bit for this member and adjust the busy count. Refer to Figure 1.

Table 1 ICRST AUDIT KEEP ALIVE FUNCTION

Keep Alive Bit	Call Processing Bit	Keep Alive Function Action
0	Busy	Set keep alive bit to 1
0	Idle	Don't care
1	Busy	Set keep alive bit to 0 + call processing bit to idle
1	Idle	Don't care

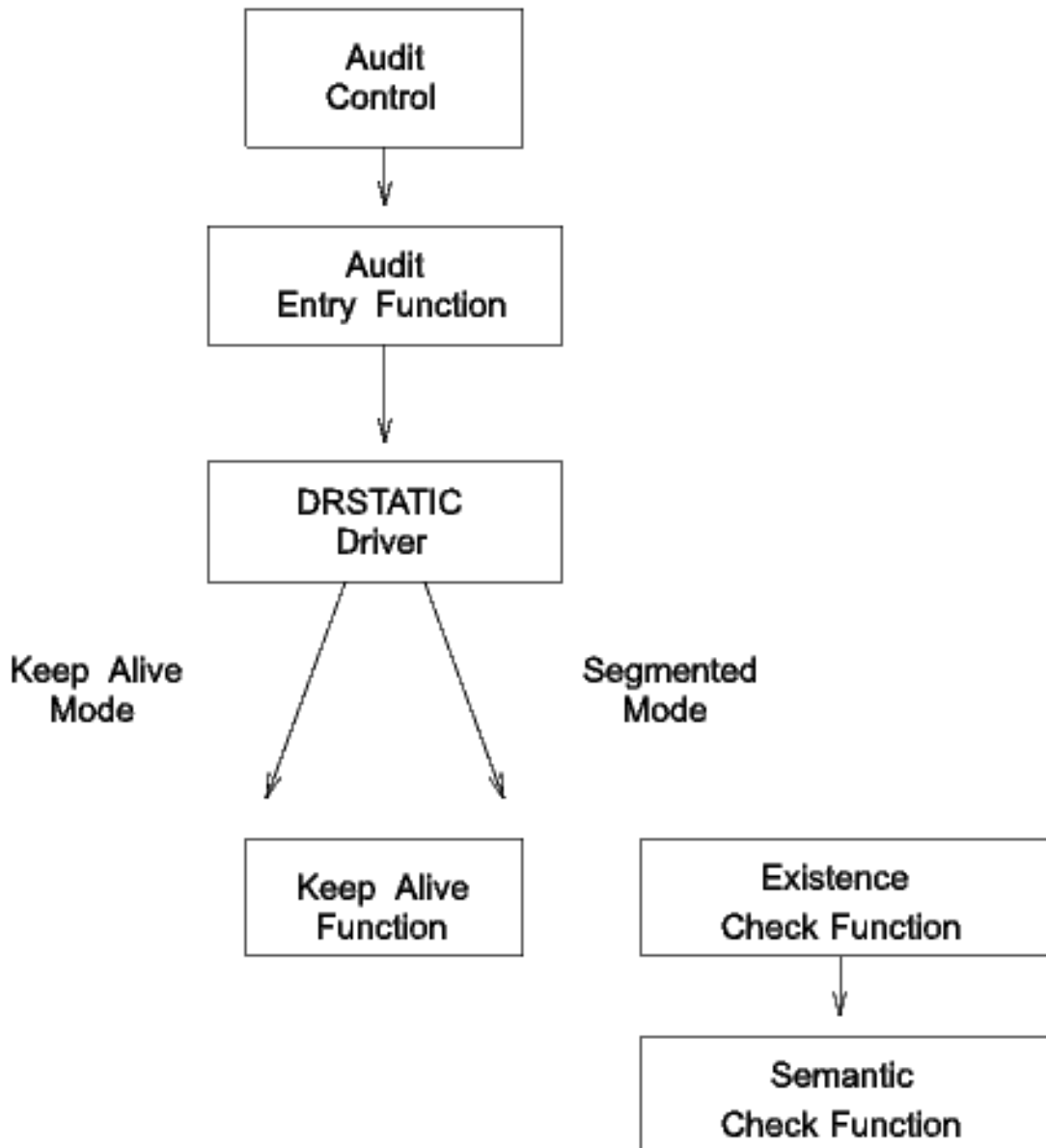


Figure 1 (ICRST-1) Existence Checks Processing

3. AUDIT CHECKS

The ICRST audit is located at AUicr module as part of the BASE package. It runs in the segmented mode and operates in the OSDS-M environment. The ICRST audit steps through the dynamic relation RLICRST and RLICRLGST respectively one tuple at a time. For each tuple, the following checks are performed:

- Perform dynamic-to-static existence check. For each tuple, verify that the corresponding tuple is existing at the static relation rlICR for the tracking SM. If it is not, this tuple is deleted and an error message is printed on the ROP.
- Verify the consistency between rlICR and rlICRST regarding size, AMA billing status and the type of ICR busy lamp. The ICR busy lamp could be associated with analog or attendant console. If there is any inconsistency, the dynamic data is corrected to be consistent with the static data.

- (c) It verifies that current busy counts are the same as those indicated by busy/idle bit map. If there is any inconsistency, the current busy count is reset.
- (d) If this SFG has an analog or attendant console equipped with busy lamp, the busy lamp will be turned on/off according to the current busy count.

If any inconsistency is found by this semantic function, the dynamic data in the rLICRST tuple is corrected and the ICRST audit continues execution without returning FALSE to the driver, which causes the driver to take a time break.

4. ERROR HANDLING

If no tuple is found in rLICR, and RLICRST indicates that it exists, an attempt is made to delete the extra tuple and an error message is printed. Audit control also schedules the related audit to be run at the elevated mode after the ICRST audit is completed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The busy lamp type indicated by rLICRST dynamic tuple is inconsistent with that specified in the static rLICR tuple.
Possible Error Effect:	The attendant console may receive erroneous lamp treatment.
Corrective Action Taken:	Correct busy lamp type of rLICRST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRST tuple.
<i>Bad Data =</i>	Busy lamp type at rLICRST.
<i>Good Data =</i>	Busy lamp type at rLICR.
<i>Logical Index =</i>	Key of rLICRST tuple.
<i>Dump =</i>	rLICRST tuple.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The busy count does not agree with that calculated from busy/idle status array.
Possible Error Effect:	Several error effects may happen: • ICR calls may be erroneously rejected. • The attendant console may receive erroneous lamp treatment. • Application code may waste time to search idle member.
Corrective Action Taken:	Correct busy count field to be consistent with busy/idle array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rLICRST tuple.
<i>Bad Data =</i>	Busy count before correction.
<i>Good Data =</i>	Actual busy count of busy/idle status.
<i>Logical Index =</i>	Key of rLICRST tuple.
<i>Dump =</i>	rLICRST tuple.

Table 4 Error Code: DDAMABIL

Error Code:	DDAMABIL

Error Description:	The AMA flag indicated by rICRST dynamic tuple is inconsistent with that specified in the static rICR tuple.
Possible Error Effect:	The AMA report could be erroneously generated or discarded.
Corrective Action Taken:	Correct AMA flag field of rICRST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rICRST tuple.
<i>Bad Data</i> =	AMA flag at rICRST.
<i>Good Data</i> =	AMA flag at rICR.
<i>Logical Index</i> =	Key of rICRST tuple.
<i>Dump</i> =	rICRST tuple.

Table 5 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The rICRST tuple should not be populated at this SM, because the corresponding static rICR tuple does not exist.
Possible Error Effect:	ICR calls originated from non-tracking SM may be erroneously accepted or rejected.
Corrective Action Taken:	The rICRST tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rICRST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rICRST tuple.
<i>Dump</i> =	rICRST tuple.

Table 6 Error Code: KEY

Error Code:	KEY
Error Description:	An invalid key value is found in the rICRST tuple.
Possible Error Effect:	The intercom calls of this SFG will be rejected. Assert should be taken from application code.
Corrective Action Taken:	A key error will be reported and corrected by the ICRST audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rICRST tuple.
<i>Bad Data</i> =	The error key.
<i>Good Data</i> =	Correct key value.
<i>Logical Index</i> =	Key of rICRST tuple.
<i>Dump</i> =	rICRST tuple.

Table 7 Error Code: RQSIZE

Error Code:	RQSIZE
Error Description:	The size indicated by rICRST dynamic tuple is different from that specified in the static rICR tuple.
Possible Error Effect:	The ICR calls may be erroneously rejected or free calls may be allowed.
Corrective Action Taken:	Correct the size field of rICRST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rICRST tuple.
<i>Bad Data</i> =	Size at rICRST.
<i>Good Data</i> =	Size at rICR.
<i>Logical Index</i> =	Key of rICRST tuple.
<i>Dump</i> =	rICRST tuple.

Table 8 Error Code: TUPLE

Error Code:	TUPLE
Error Description:	The rICRST tuple should not be populated at this SM, either this SM is not tracing SM or the size is not a large SFG.
Possible Error Effect:	If rICRST is not populated at the tracking SM, application code will assert and request ICRST audit. And free ICR calls will be given.

Corrective Action Taken:	The rIICRST tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rIICRST tuple.
<i>Bad Data</i> =	The tracking SM of rIICR tuple. The size of rIICRST may be wrong for this tuple. Check the rIICRST dump to verify that information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIICRST tuple.
<i>Dump</i> =	rIICRST tuple.

ICRTG

Software Release: 5E11 and later

1. INCOMING ROUTING DATA (ICRTG) AUDIT

ICRTG audit steps through the RLLAB7_SM relation. For each tuple in the RLLAB7_SM, the audit makes sure that the SM number in the CCicrtg[] is the same as the data in the RLLAB7_SM (CCS7 trunk routing data, which converts destination point code index and circuit identification code to the SM number which the trunk resides).

2. DATA STRUCTURES

For details on the fields contained in the RLLAB7_SM relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The dynamic flag CCicrmp does not equal static value Gleicr but no update is in progress (the value dumped in the error address field of the audit report is not SILOCKOUT).
Possible Error Effect:	CCicrtg[] may not be sent to the protocol handlers. CCS call processing could fail.
Corrective Action Taken:	The audit sets CCicrmp to the value of Gleicr (reported in the good data field of the audit error report).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Return from check for updates.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Gleicr.
<i>Logical Index =</i>	CCicrmp.
<i>Dump =</i>	NA.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	One of the following cases applies: (1) Gleicr equals DBYES but at least one of the following is true: - This is not a global SM (GI_pcrd != Gglobalsm). - The SM has no CCS PH3s or PH6s according to dynamic array SMph_type[] (which is built from relation RLDSLGDATA). That is, for all entries in the array, SMPH_TYPE.imgtype != SMPH3C and != SMPH6C.

	(2) Gleicr equals DBNO but both of the following are true: - This is a global SM (Glglobalsm). - At least one entry in SMph_type[] indicates that a CCS PH3 or PH6 is equipped (SMPH_TYPE.imgtype = SMPH3C or SMPH6C).
Possible Error Effect:	CCicrtg[] may not get sent to the protocol handlers (PHs). CCS call processing may fail.
Corrective Action Taken:	The PHDSLGP audit will run as a related audit and correct any errors in SMph_type[]. However, if the PHDSLGP audit does not find any errors and this error continues to be found, there may be a problem with the value of Gleicr, Glglobalsm or RLDSLGDATA. In this case, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual. Population rules for the above mentioned global data may also help technicians determine the exact problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Glglobalsm.
<i>Bad Data =</i>	PH number of the first CCS PH3, if any. Or this value will be SMMAXPH.
<i>Good Data =</i>	Gleicr.
<i>Logical Index =</i>	CCicrmp.
<i>Dump =</i>	NA.

Table 3 Error Code: MISSING

Error Code:	MISSING
Error Description:	Gleicr is set to DBYES and no update is in progress (the value dumped in the error address field of the audit report is not SILOCKOUT) but the Glicrsa[] optional data region has not been populated correctly according to system integrity.
Possible Error Effect:	CCicrtg[] may not have been sent to the protocol handlers. CCS call processing could fail.
Corrective Action Taken:	The audit initiates a rebuild of the CCicrtg[] table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Return from check for updates.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Gleicr.
<i>Logical Index =</i>	CCicrmp.
<i>Dump =</i>	NA.

Table 4 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	A CCicrtg[] entry is not populated in the Glicrsa[] optional data region. Note that the CCicrtg[] entry is accessed by using the static relation's (RLLAB7_SM on a non-RSM, RLLAB7SM on an RSM) keys (dpci and cic) as the keys to a hashing process which determines the array index.
Possible Error Effect:	CCS routing may fail sometimes because of the unpopulated CCicrtg[] entry.
Corrective Action Taken:	The audit attempts to insert the missing CCicrtg[] entry into the Glicrsa[] optional data region.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rLLAB7_SM.sm (in a non-RSM). rLLAB7SM.sm (in an RSM).
<i>Bad Data =</i>	rLLAB7_SM.pci (in a non-RSM). rLLAB7SM.pci (in an RSM).
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	rLAB7_SM.cic (in a non-RSM). rLAB7SM.cic (in an RSM).
<i>Dump</i> =	NA.

Table 5 Error Code: READ

Error Code:	READ
Error Description:	The audits attempts to read an rIDPC_STAT tuple whose key (rIDPC_STAT.dpci) matches rLAB7SM.dpci but a tuple in relation RLDPC_STAT is not found.
Possible Error Effect:	CCicrtg[] may not be sent to the PHs. Common channel signaling (CCS7) routing may fail when attempting to route to the destination point code (DPC) identified by rLAB7SM.dpci.
Corrective Action Taken:	Another audit inserts the missing tuple into relation RLDPC_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	rLAB7SM.sm.
<i>Good Data</i> =	rLAB7SM.cic.
<i>Logical Index</i> =	rLAB7SM.dpci.
<i>Dump</i> =	NA.

Table 6 Error Code: SEARCH_FAILED

Error Code:	SEARCH_FAILED
Error Description:	A CCicrtg[] entry could not be found in the Glicrsa[] optional data region and the table is fully populated. Note that the CCicrtg[] entry is accessed by using the static relation's (RLLAB7_SM on a non-RSM, RLLAB7SM on an RSM) keys (dpci and cic) as the keys to a hashing process which determines the array
Possible Error Effect:	CCS routing may fail sometimes because of the missing CCicrtg[] entry.
Corrective Action Taken:	None. This error is only for information. Something may be wrong with the Glicrsa[] optional data region layout or CCS code may have a bug. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rLAB7_SM.sm (on a non-RSM). rLAB7SM.sm (on an RSM).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	rLAB7_SM.pci (on a non-RSM). rLAB7SM.pci (on an RSM).
<i>Logical Index</i> =	rLAB7_SM.cic (on a non-RSM). rLAB7SM.cic (on an RSM).
<i>Dump</i> =	NA.

Table 7 Error Code: TBL_ERR

Error Code:	TBL_ERR
Error Description:	An entry in CCicrtg[] contains an sm field which does not match the sm field in the corresponding static rLAB7_SM (on a non-remote switching module) or rLAB7SM (on a remote switching module) tuple. Note that the CCicrtg[] entry is accessed by using the static relation's keys (dpci and cic) as the keys to a hashing process which determines the array index.
Possible Error Effect:	CCS routing may fail when attempting to route to the switching module provided in the CCicrtg[] entry.
Corrective Action Taken:	The audit triggers an update of the CCicrtg[] entry. In particular, CCICRTG.sm will be set to the value from rLAB7_SM.sm (on a non-RSM) or rLAB7SM (on an RSM).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rLAB7_SM.sm (on a non-RSM). rLAB7SM.sm (on an RSM).
<i>Bad Data</i> =	rICRTG.sm.
<i>Bad Data</i> =	rLAB7_SM.pci (on a non-RSM). rLAB7SM.pci (on an RSM).
<i>Logical Index</i> =	rLAB7_SM.cic (on a non-RSM). rLAB7SM.cic (on an RSM).
<i>Dump</i> =	NA.

Table 8 Error Code: UPDATE_FAILED

Error Code:	UPDATE FAILED
Error Description:	Due to recovery from other errors, the audit has made enough updates to the CCicrtg[] array that a table rebuild is needed.
Possible Error Effect:	Since access to the CCicrtg[] array is done using a hashing algorithm, if too many updates are made, additional insertions may result in collisions.
Corrective Action Taken:	The audit initiates a rebuild of the CCicrtg[] table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	A hashing key to the CCicrtg[] index: rLAB7_SM.dpci (in a non-RSM), rLAB7SM.dpci (in an RSM).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The other hashing key to the CCicrtg[] index: rLAB7_SM.cic (in a non-RSM), rLAB7SM.cic (in an RSM).
<i>Dump =</i>	NA.

Table 9 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect error, out of range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies depending on the process that took the interrupt and what the interrupt was.
Corrective Action Taken:	This report is for information only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CCicrmpm.
<i>Good Data =</i>	Gleicr.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

IDDS1**Software Release:** 5E10 and later**1. INTEGRATED DIGITAL SERVICE UNIT-DIGITAL SERVICE FACILITY (IDDS1) AUDIT**

The IDDS1 audit will perform the semantic checks of the integrated digital service unit (IDCU) digital service (DS1) facility array PCidds1[][].

2. DATA STRUCTURES

The data structure being audited by IDDS1 audit is the PCidds1[][] array. It is a doubly indexed array (PCidds1[idcu][ds1]) where the indexes are the IDCU number and the digital service facility DS1 number. The array fields and a brief description is as follows:

PCidds1[][].usage	= Contains the busy/idle status of the DS0 time slots.
PCidds1[][].rtsmask	= Contains the DS0 times slots that have been permanently or semi-permanently reserved (0 means reserved, 1 means idle).
PCidds1[][].rtpos	= Contains the DS1 number from the RT perspective (that is, 1-128).
PCidds1[][].maintst	= Contains the in-service/out-of-service state of the DS1 facility.

3. AUDIT CHECKS

The IDDS1 audit will perform the semantic checks of the PCidds1[][] array fields. The audit sequence is as follows:

ZAP	= Verify the audit entry mode.
ODDLOSTTUP	= Read the rIDS1_PT tuple for a given DS1 facility.
SMEST	= Read the rSMEST tuple for the DS1 facility.
TSOOSQUAL	= Audit the PCidds1[][].maintst field.
DEFRTPOS	= Audit the PCidds1[][].rtpos field for default value.
GR_RTMASK	= Audit the PCidds1[][].rtsmask field for the default value.
BADRTMASK	= Audit the PCidds1[][].rtsmask field against the rIDS1_PT tuple.
MAINT_STATE	= Audit PCidds1[][].maintst field is audited against the DS1's rISMSET tuple.
BAS_STATE	= Audit PCidds1[][].maintst field against the DS1's rSMEST tuple that is on protection.
RTPOS	= Audit PCidds1[][].rtpos field against the rIDS1_PT tuple.
KEY	= Read the rIDS0B tuple for a given rIDS0B.ds0b field.
IDLE_BUSY	= Audit the PCidds1[][].usage field for the correct idle setting.
BUSY_IDLE	= Audit the PCidds1[][].usage field for the correct busy setting.
STATE_VAR	= Audit the rIDS0B.use field for in-range value.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADRTMASK

Error Code:	BADRTMASK
Error Description:	The contents of the PCidds1[][].rtsmask field was found different than the value derived from the rIDS1_PT tuple.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	The contents of the PCidds1[][].rtsmask field will be set to the value derived from the rIDS1_PT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCidds1[][] array.
<i>Bad Data =</i>	The contents of the PCidds1[][].rtsmask field.
<i>Good Data =</i>	The contents of what the PCidds1[][].rtsmask field should be. It is derived from the rIDS1_PT tuple.
<i>Logical Index =</i>	The circuit name of the DS1 facility.
<i>Dump 1 =</i>	The contents of the PCidds1[][] array.
<i>Dump 2 =</i>	The contents of the rIDS1_PT tuple.

Table 2 Error Code: BAS_STATE

Error Code:	BAS_STATE
Error Description:	The PCidds1[][].maintst field was found to be out-of-service while the rISMEST tuple for the DS1 facility was out-of-service protected. The PCidds1[][].maintst field is expected to be active since the DS1 facility is protected.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	The audit will correct the PCidds1[][].maintst field to be active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCidds1[][] array.
<i>Bad Data =</i>	The contents of the PCidds1[][].maintst field.
<i>Good Data =</i>	The contents of what the PCidds1[][].maintst field should be, which is an active state.
<i>Logical Index =</i>	The circuit name of the DS1 facility.
<i>Dump 1 =</i>	The contents of the DS1 facility rISMEST tuple.
<i>Dump 2 =</i>	The contents of the PCidds1[][] array.

Table 3 Error Code: BUSY_IDLE

Error Code:	BUSY_IDLE
Error Description:	The bit in PCidds1[][].usage field was found to be busy but it should have been idle. This determination was made when the rIDS0B tuple for the DS0 time slot being audited was read and the use field in it was that of an idle DS0 time slot.
Possible Error Effect:	The call processing will not be able to use that particular DS0 time slot.
Corrective Action Taken:	The audit will idle out the corresponding bit in PCidds1[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCidds1[][] array.
<i>Bad Data =</i>	The value of the bit found idle which is a 0.
<i>Good Data =</i>	The value of the what the bit should be which is a 1.
<i>Logical Index =</i>	The DS0 time slot number found at error which is a number from 0-23.
<i>Dump 1 =</i>	The contents of the rIDS0B tuple.
<i>Dump 2 =</i>	The contents of the PCidds1[][] array.

Table 4 Error Code: DEFRTPOS

Error Code:	DEFRTPOS
Error Description:	The PCidds1[][].rtpos field was supposed to be set to the default value which is 0XFF but was found not to be so.
Possible Error Effect:	The call processing will try to use that DS1 facility but it is not usable.
Corrective Action Taken:	The audit will set the PCidds1[][].rtpos field to the default value of 0XFF.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidss1[][] array.
<i>Bad Data</i> =	The contents of the PCidss1[][].rtpos field.
<i>Good Data</i> =	The contents of what the PCidss1[][].rtpos field should be.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the PCidss1[][] array.

Table 5 Error Code: GR_RTMASK

Error Code:	GR_RTMASK
Error Description:	The audit was expecting the PCidss1[][].rtmask field to be set to the default value 0X00FFFFFF but was found not to be.
Possible Error Effect:	The call processing will try to use the DS1 facility but it is not usable.
Corrective Action Taken:	The audit will set the PCidss1[][].rtmask field to the default value of 0X00FFFFFF.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidss1[][] array.
<i>Bad Data</i> =	The contents of the PCidss1[][].rtpos field.
<i>Good Data</i> =	The contents of what the PCidss1[][].rtpos field should be.
<i>Logical Index</i> =	The name of the DS1 facility.
<i>Dump</i> =	The contents of the PCidss1[][] array will be dumped.

Table 6 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The bit in PCidss1[][].usage field was found to be idle but it should have been busy. This determination was made when the rIDS0B tuple for the DS0 time slot being audited was read and the use field in it was that of a stable call.
Possible Error Effect:	The call processing will reuse that particular DS0 time slot.
Corrective Action Taken:	The audit will busy out the corresponding bit in PCidss1[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidss1[][] array.
<i>Bad Data</i> =	The value of the bit found idle which is a 1.
<i>Good Data</i> =	The value of what the bit should be which is a 0.
<i>Logical Index</i> =	The DS0 time slot number found at error which is a number from 0-23.
<i>Dump 1</i> =	The contents of the rIDS0B tuple.
<i>Dump 2</i> =	The contents of the PCidss1[][] array.

Table 7 Error Code: KEY

Error Code:	KEY
Error Description:	The rIDS0B tuple was missing from the database.
Possible Error Effect:	The call processing will not be able to use that particular DS0 time slot.
Corrective Action Taken:	The audit will busy out the corresponding bit in PCidss1[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidss1[][] array.
<i>Bad Data</i> =	The rIDS0B.ds0b key field.
<i>Good Data</i> =	The time slot on the DS1 facility being audited.
<i>Logical Index</i> =	The rIDS0B.dsob key field.
<i>Dump</i> =	The contents of the PCidss1[][] array.

Table 8 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The PCidss1[][].maintst field was found to be different from the maintenance state of the rSMEST tuple.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	The PCidss1[][].maintst field will be corrected to match that of the rSMEST tuple maintenance state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidss1[][] array.

<i>Bad Data</i> =	The contents of the PCidds1[][].maintst field.
<i>Good Data</i> =	The contents of what the PCidds1[][].maintst field which is that of the rSMEST bas_stat field.
<i>Logical Index</i> =	The name of the DS1 facility.
<i>Dump 1</i> =	The dump of the DS1 facility rSMEST tuple.
<i>Dump 2</i> =	The contents of the PCidds1[][] array.

Table 9 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The rIDS1 audit tried to read the rIDS1_PT tuple for a given DS1 facility but there was no tuple in the database with that key.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should insert a tuple in the rIDS1_PT tuple for the given
Dump Description:	
<i>Error Address</i> =	The address of the rIDS1_PT buffer.
<i>Bad Data</i> =	The name of the DS1 facility.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the rIDS1_PT buffer will be dumped.

Table 10 Error Code: RTPOS

Error Code:	RTPOS
Error Description:	The PCidds1[][].rtpos field was found to be inconsistent with what the rIDS1_PT tuple contents.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	The audit will correct the PCidds1[][].rtpos field to be consistent with the rt_ds1 field in rIDS1_PT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidds1[][] array.
<i>Bad Data</i> =	The contents of the PCidds1[][].rtpos field.
<i>Good Data</i> =	The contents of what the PCidds1[][].rtpos field should be, which is derived from the rt_ds1 field in rIDS1_PT tuple.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the PCidds1[][] array.

Table 11 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for the DS1 facility was missing from the database.
Possible Error Effect:	The maintenance (removal and restoration) of the DS1 facility will not be possible.
Corrective Action Taken:	The audit will verify that all the fields in the PCidds1[][] entry being audited are set to the default value until the rSMEST tuple is inserted in the database by the related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rSMEST pointer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the rIDS1_PT buffer will be dumped.

Table 12 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The use field in rDSOB tuple was found to be an out of range value.
Possible Error Effect:	The call processing will not be able to use that particular DS0 time slot.
Corrective Action Taken:	The audit will idle out the corresponding bit in PCidds1[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidds1[][] array.

<i>Bad Data</i> =	The value of the use field in rIDS0B tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The DS0 time slot number found at error which is a number from 0-23.
<i>Dump 1</i> =	The contents of the rIDS0B tuple.
<i>Dump 2</i> =	The contents of the PCidss1[][] array.

Table 13 Error Code: TSOOSQUAL

Error Code:	TSOOSQUAL
Error Description:	The PCidss1[][].maintst field was found to be active, it should have been out-of-service.
Possible Error Effect:	The call processing will try to use that particular DS1 facility but it is not usable.
Corrective Action Taken:	The audit will set the PCidss1[][].maintst field to the out-of-service state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidss1[][] array.
<i>Bad Data</i> =	The value of the PCidss1[][].maintst field.
<i>Good Data</i> =	The value of what the PCidss1[][].maintst field should be.
<i>Logical Index</i> =	The circuit name of the DS1 facility.
<i>Dump</i> =	The contents of the PCidss1[][] array.

Table 14 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to provide recovery after taking a reset. When a reset occurs (usually triggered by an illegal or out of range memory access), the audit which was running at the time is aborted, and is then rescheduled to run in AUZAP mode to provide recovery of related data left inconsistent as a result of the reset.
Possible Error Effect:	The audit may continue to cause memory access violations, increasing the probability of drastic recovery action and preventing the audit from checking for further errors.
Corrective Action Taken:	The data which caused the reset is initialized by the next entry of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

IDLPRRT

Software Release: 5E10 and later

1. HUNTED IDLE PORT (IDLPRRT) AUDIT

During call processing, the RTA subsystem routes traffic based on the port status in the administrative module (AM). The RTA routing process determines the busy/idle status of each port by hunting in the AM. (Hunting is the process of selecting one of a set of available ports that satisfy a calling customer.) The busy/idle status of the port is made up of both the hunt key status and the hunted port status. Audits ensure that both are correct. The hunted idle port audit (IDLPRRT) verifies the hunted port status and ensures that ports that are idle in the switching module (SM) are listed as idle in the AM.

The IDLPRRT audit is an interprocessor audit that begins in the AM and audits one SM each time it runs. It will not audit the same SM twice in less than 15 minutes; that is, the minimum entry rate per SM is 15 minutes. In large office, the audit may require several cycles to audit all the SMs. To do this, the IDLPRRT audit runs in all three modes of the ABACABA audit cycle-seven times (and hence, seven SMs) per audit cycle. For example, the audit cycle when there is a heavy call load requires about 15 minutes in the AM. In an office with 80 SMs, three hours will be required before all the SMs are audited.

This audit is part of the routing and terminal allocation audits subgroup. It operates in both the operational kernel process (OKP) and operating system for distributed switching-switching module (OSDS-SM) environments in the post initialization and segmented modes.

The audit sequences through all the PORTLA tuples and loads each in service, hunted and idle port into a data message to be sent to the AM. The message loading is handled by the semantic function AUSidlprt. To accommodate the changes introduced by the CSD feature, the semantic function will check each hunted and idle port to see if it is a B-channel portname. If it is, the line unit is stripped from the portname. The ltype is checked to see if it is LU3. For LU3 and B-channel port is sent to the AM. Otherwise, the B-channel is converted to a D-channel port and then sent to the AM. In the AM, the function RTauptav is called to update the LNSTAT and LGLNSTAT relations.

This audit works only for trunk ports. The part of IDLPRRT responsible for idling of stuck MLHGs members has been replaced by the Keep Alive mechanism.

2. DATA STRUCTURES

The IDLPRRT audit operates on the port list relation (RLPRTLST) and the port linkage area relation (RLPORTLA) in the SM. The port list relation consists of the head cells for the lists of in-service and forced-in-service ports. The port linkage area relation contains information about the associations the port has with its processes and about the primary abbreviated status of the port.

The IDLPRRT audit is an interprocessor audit whose control is in the AM which calls the IDLPRRT audit collection function in the SM. The SM collection function collects the busy/idle status of the ports in rPORTLA and returns a message which contains the idle ports, to the AM to update the rLNSTAT.

IDLPRRT checks the busy/idle status in the RLLTMSTAT relation by going through the RLIS_LINE relation and checking if the lead_module is set to the current module and that the ltype is IS_MLHG. If these conditions are met, we can access an RLIS_TERM relation to check the cpe_type. If the cpe_type is set to IS_TYPATT, the port has an attendant associated with it.

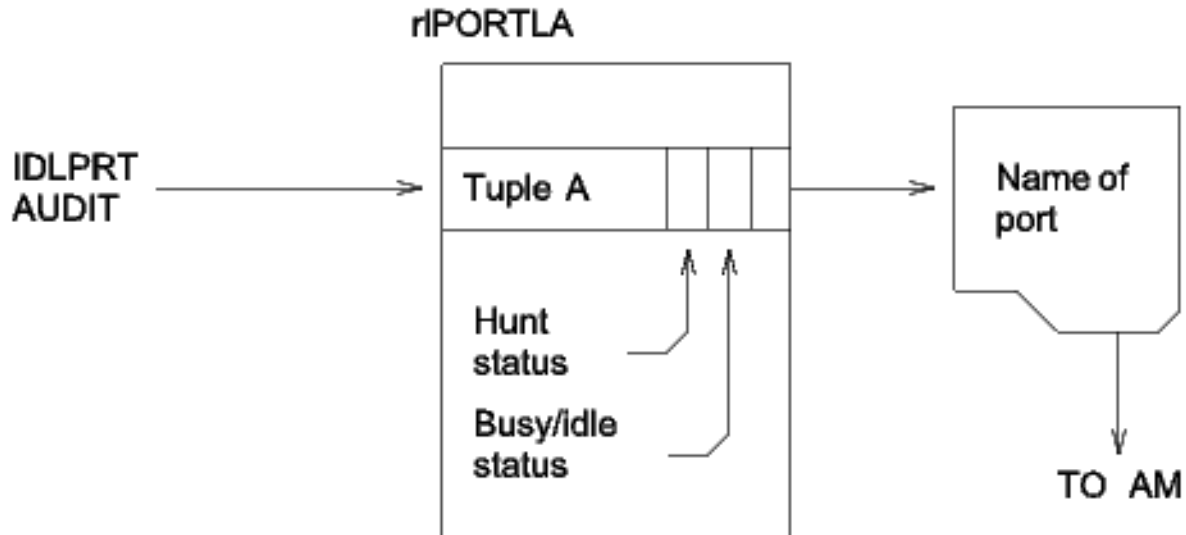
When dealing with the attendant, a DPB tuple must be accessed in order to find a TEDB linked to it. When a TEDB tuple is identified, a check is run to see if the sca bit is set to DBNUU. If it is, go through the RLLTMSTAT relation to check the busy/idle status by ANDing the status bits together.

If the cpe_type of the port is not IS_TYPATT, the audit goes through the rLTMSTAT relation to check the busy/idle

status by ORing the status bits together.

The audit then sends a message to the AM which contains the ports that are idle. The message is distributed to update the RLLNSTAT relation which might have a port marked busy when it should be idle.

Relations RLLNSTAT and RLLGLNSTAT will not be populated in the AM environment anymore.



SM

AM

Figure 1 (IDLPRT-1) Hunted Idle Port Audit

3. AUDIT CHECKS

The IDLPRT audit calls the traverse driver (DRTRAV) to traverse the tuples of the RLPORTLA relation.

This audit begins in an SM (refer to Figure 1) and searches the port linkage area relation (RLPORTLA) for hunted ports. When it finds a hunted port, it determines whether the port is busy or idle. If the hunted port is busy, the audit proceeds to the next port in the RLPORTLA relation. If the hunted port is idle, the audit notes this in a message (each message contains a total of 64 ports) to be sent to the AM. The AM compares the list of idle ports received from the SMs with the status of those ports that it contains.

The audit calls besides DRTRAN driver, the DRSTATIC driver. DRSTATIC is used for stepping through the rIIS_LINE tuples.

This audit may make the AM think that a port in the SM is idle when it is actually busy. This error occurs when the status of the port in the SM changes from idle to busy between the time it is audited and the time the AM receives information about its status. However, the error is insignificant because a call to a busy hunted port is automatically routed to the next idle port. The audit will recover the inconsistency the next time it runs.

The IDLPRT audit may find four errors during its execution:

- (1) The SM portion of the audit is called in the wrong mode (any mode other than segmented). If this occurs, the audit fires a returning assert that makes audit control initialize partition control for the audit.

- (2) Partition control has been corrupted. If this occurs, the audit fires an assert that purges the audit process.
- (3) The data message sent to the AM returns an invalid return code. If this occurs, the audit fires an assert that purges the audit process.
- (4) The DRTRAV driver finds a linkage error in a port linkage area (rIPORTLA) tuple. If this occurs, the IDLPRT audit aborts and schedules the PORTLA audit to run in the elevated segmented mode as a related audit.

If the IDLPRT audit detects an inconsistency between the AM and SM the port status is updated. If the audit encounters any errors with the static relation reads, an error is printed and the user is required to take manual action.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	There was a missing tuple found for rIIS_TERM.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	Fire an assert and return AUFALSE to the driver.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Logical address of the rIIS_TERM tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIIS_TERM tuple.
<i>Dump =</i>	NA.

Table 2 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered after taking a reset.
Possible Error Effect:	A machine reset has occurred.
Corrective Action Taken:	The audit has returned to Audit Control.
Manual Action Required:	No manual action is required.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Partition cursor.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

IDPIDB

Software Release: 5E10 and later

1. INTEGRATED DIGITAL SERVICE UNIT FACILITY PIDB (IDPIDB) AUDIT

The IDPIDB audit will perform the semantic checks of the integrated digital service unit (IDCU) peripheral interface database (PIDB).

2. DATA STRUCTURES

The data structure being audited by IDPIDB audit is the PCidpidb[][] array. It is a doubly indexed array (PCidpidb[idcu][pidb]) where the indexes are the IDCU number and the PIDB number. The array components (fields and a brief description) are as follows:

PCidpidb[][].usage	= Contains the busy/idle status of the PIDB time slots.
PCidpidb[][].rtsmask	= Contains the times slots that have permanently or semi-permanently been reserved (0 means reserved, 1 means idle).
PCidpidb[][].syspidb	= The system PIDB number.
PCidpidb[][].type	= The type of PIDB.
PCidpidb[][].maintst	= The maintenance status of the PIDB (active, out-of-service or blocked).

3. AUDIT CHECKS

The IDPIDB audit will perform the semantic checks of the PCidpidb[][] array structure. The audit sequence is as follows:

ZAP	= The audit entry mode is verified.
MAINT_STATE2	= The PCidpidb[][].maintst field is audited for default setting.
NON_NULL	= The PCidpidb[][].type field is audited for the default setting.
GR_SYSPIDB	= The PCidpidb[][].syspidb field is audited for the default setting.
GR_RTMASK	= The PCidpidb[][].rtsmask field is audited for the default setting.
ODDLOSTTUP	= The rISF14 tuple should exist for a given equipped IDCU.
ODDLOSTTUP	= The rIDPIDB_PT tuple should exist for given PIDB circuit name.
BADRTMASK	= The PCidpidb[][].rtsmask field is audited against the static rIDPIDB_PT tuple.
SMEST	= The rISMEST tuple for a given PIDB is read.
ACCBK	= A now authorized DUALPIDB is unblocked for path hunting.
NOTBUSY	= A now idle unauthorized DUALPIDB is removed from service.
MAINT_STATE	= The PCidpidb[][].maintst field is audited against the rISMEST tuple.
BADTYPE	= The PCidpidb[][].type field is audited against the rCKTDATA.pidbtype field.

SYSPIDB	= The PCidpidb[][].syspidb is audited against the static rCKTDATA.pidb field.
BADUSE	= An unauthorized DUALPIDB is path hunt blocked and removed from service if idle.
KEY	= The rIPTSB tuple for a given PIDB is read.
IDLE_BUSY	= The PCidpidb[][].usage is audited for an idle rIPTSB tuple.
BUSYIDLE	= The PCidpidb[][].usage is audited for an busy rIPTSB tuple.
STATE_VAR	= The rIPTSB.use field is audited.

4. ERROR HANDLING

The general audit error handling mechanism is to print the error code and make the PCidpidb[][] array consistent with the data that it is being compared against (such as, rSMEST tuple, rDPIDB_PT tuple and so forth). If a user intervention is required (such as, in the case of a missing static data) then a user intervention error code will be printed.

The audit produces the following error code(s).

Table 1 Error Code: ACCBLK

Error Code:	ACCBLK
Error Description:	The IDPIDB audit found that a PCDUALPIDB's path hunt was blocked even though it is within the valid pidb limit.
Possible Error Effect:	Call processing will not be able to use that particular PIDB facility.
Corrective Action Taken:	The PCidpidb[][].maintst will be set to match the value derived from the rSMEST tuples for the PIDBs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IDCU unit number.
<i>Bad Data</i> =	The count of active PCDUALPIDBs.
<i>Good Data</i> =	The pidb limit defined in rSF14 for this IDCU.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump 1</i> =	The contents of the rSF14 tuple.
<i>Dump 2</i> =	The contents of the PCidpidb[][] array.

Table 2 Error Code: BADRTMASK

Error Code:	BADRTMASK
Error Description:	The IDPIDB audit found a the PCidpidb[][].rtsmask field not to be set to what the static rDPIDB_PT tuple states it should be set to.
Possible Error Effect:	The call processing will not be able to use that particular PIDB facility.
Corrective Action Taken:	The PCidpidb[][].rtsmask will be set to match the value derived from the static rDPIDB_PT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].rtsmask field.
<i>Good Data</i> =	The contents of what the PCidpidb[][].rtsmask field should be.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The PCidpidb[][].type field was found to be different from what the static rCKTDATA.pidbtype field states it should be.
Possible Error Effect:	The call processing will not be able to use that PIDB facility.
Corrective Action Taken:	The audit will set the PCidpidb[][].pidbtype field to match the pidbtype

	rICKTDATA.pidbtype field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].type field.
<i>Good Data</i> =	The contents of what the PCidpidb[][].type field should be.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 4 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The IDPIDB audit found a PCDUALPIDB over the authorized limit defined by rISF14.
Possible Error Effect:	The call processing should not be able to use that particular PIDB facility.
Corrective Action Taken:	The PCDUALPIDB will be blocked from use by call processing.
Manual Action Required:	The office engineers should refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual to obtain authorization to use additional PIDBs.
Dump Description:	
<i>Error Address</i> =	The IDCU unit number.
<i>Bad Data</i> =	The count of active PCDUALPIDBs.
<i>Good Data</i> =	The pidb limit defined in rISF14 for this IDCU.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump 1</i> =	The contents of the rISF14 tuple.
<i>Dump 2</i> =	The contents of the PCidpidb[][] array.

Table 5 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The bit in PCidpidb[][].usage field was found to be busy but it should have been idle. This determination was made when the rIPTSB.use field was that of an idle rIPTSB.
Possible Error Effect:	The call processing will not be able to use that particular PIDB time slot.
Corrective Action Taken:	The audit will idle out the corresponding bit in PCidpidb[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].usage field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIPTSB.ptsb key.
<i>Dump 1</i> =	The contents of the rIPTSB tuple.
<i>Dump 2</i> =	The contents of the PCidpidb[][] array.

Table 6 Error Code: GR_RTMASK

Error Code:	GR_RTMASK
Error Description:	The IDPIDB audit found the PCidpidb[][].rtsmask field not set to the default value which is 0xFFFFFFFF.
Possible Error Effect:	The call processing will try to use that particular PIDB facility but it is not usable.
Corrective Action Taken:	The audit will set the PCidpidb[][].rtsmask field to the default value which is 0xFFFFFFFF.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].rtsmask field.
<i>Good Data</i> =	The contents of what the PCidpidb[][].rtsmask field should be which is 0X.FFFFFFFF.
<i>Logical Index</i> =	The circuit name of the IDCU DIB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 7 Error Code: GR_SYSPIDB

Error Code:	GR_SYSPIDB
Error Description:	The IDPIDB audit found the PCidpidb[][].syspidb field not set to the default

	value which is 0XFF.
Possible Error Effect:	The call processing will try to use that particular PIDB facility but it is not usable.
Corrective Action Taken:	The audit will set the field to the default value which is 0XFF.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCidpidb[][] array.
<i>Bad Data =</i>	The contents of the PCidpidb[][].syspidb field.
<i>Good Data =</i>	The contents of what the PCidpidb[][].syspidb field should be which is 0XFF.
<i>Logical Index =</i>	The circuit name of the IDCUCIDB facility.
<i>Dump =</i>	The contents of the PCidpidb[][] array.

Table 8 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The bit in PCidpidb[][].usage was found to be idle but it should have been busy. This determination was made when the rIPTSB tuple for the PIDB time slot being audited was read and the rIPTSB.use field was that of a stable call.
Possible Error Effect:	The call processing will reuse that particular PIDB time slot.
Corrective Action Taken:	The audit will busy out the corresponding bit in PCidpidb[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCidpidb[][] array.
<i>Bad Data =</i>	The contents of the PCidpidb[][].usage field.
<i>Good Data =</i>	The contents of what the bit should be which is a 0.
<i>Logical Index =</i>	The rIPTSB.ptsb key field.
<i>Dump 1 =</i>	The contents of the rIPTSB tuple.
<i>Dump 2 =</i>	The contents of the PCidpidb[][] array.

Table 9 Error Code: KEY

Error Code:	KEY
Error Description:	The rIPTSB tuple was missing for a given PIDB time slot.
Possible Error Effect:	The call processing will not be able to use that PIDB time slot.
Corrective Action Taken:	The audit will busy out the bit in PCidpidb[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB.ptsb key field.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The contents of the rIPTSB tuple.

Table 10 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The PCidpidb[][].maintst field was found to be different from what the rSMEST.bas_stat field is.
Possible Error Effect:	The call processing will not be able use that particular PIDB facility for call processing.
Corrective Action Taken:	The audit will set the PCidpidb[][].maintst field to match the rSMEST.bas_stat field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCidpidb[][] array.
<i>Bad Data =</i>	The contents of the PCidpidb[][].maintst field.
<i>Good Data =</i>	The contents of what the PCidpidb[][].maintst field should be.
<i>Logical Index =</i>	The circuit name of the DS1 facility.
<i>Dump =</i>	The contents of the PCidpidb[][] array.

Table 11 Error Code: MAINT_STATE2

Error Code:	MAINT_STATE2
Error Description:	The IDPIDB audit found the PCidpidb[][].maintst field not to be set to the default value which is SMOOS.
Possible Error Effect:	The call processing will try to use that particular PIDB facility but it is not usable.
Corrective Action Taken:	The audit will set the PCidpidb[][].maintst field to the SMOOS value.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].maintst field.
<i>Good Data</i> =	The contents of what the PCidpidb[][].maintst field should be.
<i>Logical Index</i> =	The circuit name of the IDCU PIDB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 12 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The IDPIDB audit found the PCidpidb[][].type field not set to the default value which is PCNULLPIDB.
Possible Error Effect:	The call processing will try to use that particular PIDB facility but it is not usable.
Corrective Action Taken:	The audit will set the PCidpidb[][].type field to the SMOOS value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].type field.
<i>Good Data</i> =	The contents of what the PCidpidb[][].type field should be.
<i>Logical Index</i> =	The circuit name of the IDCU DIDB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 13 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The IDPIDB audit found a PCDUALPIDB over the authorized limit defined by rISF14. Now that this PIDB is traffic idle, it will be removed from service.
Possible Error Effect:	Call processing should not be able to use that particular PIDB facility.
Corrective Action Taken:	The PCDUALPIDB will be blocked from use by call processing and removed from service.
Manual Action Required:	The office engineers should contact their next level of technical assistance to obtain authorization to use additional PIDBs.
Dump Description:	
<i>Error Address</i> =	The IDCU unit number.
<i>Bad Data</i> =	The count of active PCDUALPIDBs.
<i>Good Data</i> =	The pidb limit defined in rISF14 for this IDCU.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump 1</i> =	The contents of the rISF14 tuple.
<i>Dump 2</i> =	The contents of the PCidpidb[][] array.

Table 14 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The IDPIDB audit found a missing rIDPIDB_PT tuple for a given PIDB circuit name. The rCKTDATA.pidbtype field states that it is a directly connected PIDB, therefore, there should exist an rIDPIDB_PT tuple in the database.
Possible Error Effect:	The call processing will not be able to use that particular PIDB facility.
Corrective Action Taken:	None.
Manual Action Required:	The user should insert an rIDPIDB_PT in the database.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	The contents of the PCidpidb[][] array.
<i>Dump 2</i> =	The CKTDATA tuple.

Table 15 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The IDPIDB audit found a missing rISF14 or rPIDBLIMIT tuple for an equipped IDCU.
Possible Error Effect:	Call processing will default to using four PCDUALPIDBs for path hunting.
Corrective Action Taken:	None.

Manual Action Required:	The office engineers should refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual to create the missing tuple for the IDCU unit specified in the Error Address field with the appropriate pidb limit.
Dump Description:	
<i>Error Address</i> =	The idcu unit number.
<i>Bad Data</i> =	The circuit name for the associated PCIDUALPIDB.
<i>Good Data</i> =	The PIDB number.
<i>Logical Index</i> =	0.
<i>Dump</i> =	None.

Table 16 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for a given PIDB facility is missing from the database.
Possible Error Effect:	The maintenance (removal and restoration) of the PIDB facility will not be possible.
Corrective Action Taken:	The audit will verify that all the fields in the PCidpidb[][] entry being audited are set to the default value until the rSMEST tuple is inserted in the database by the related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 17 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rIPTSB.use field was found to be out of range.
Possible Error Effect:	The call processing will not be able to use that particular PIDB time slot.
Corrective Action Taken:	The audit will idle out the corresponding bit in PCidpidb[][].usage field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the rIPTSB.use field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The rIPTSB.ptsb key field.
<i>Dump 1</i> =	The contents of the rIPTSB tuple.
<i>Dump 2</i> =	The contents of the PCidpidb[][] array.

Table 18 Error Code: SYSPIDB

Error Code:	SYSPIDB
Error Description:	The PCidpidb[][].syspidb field was found to be different from what the static rCKTDATA.pidb field states it should be.
Possible Error Effect:	The call processing will not be able to use that PIDB facility.
Corrective Action Taken:	The audit will set the PCidpidb[][].syspidb field to match the rCKTDATA.pidb field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCidpidb[][] array.
<i>Bad Data</i> =	The contents of the PCidpidb[][].syspidb field.
<i>Good Data</i> =	The contents of what the PCidpidb[][].syspidb field should be.
<i>Logical Index</i> =	The circuit name of the PIDB facility.
<i>Dump</i> =	The contents of the PCidpidb[][] array.

Table 19 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to provide recovery after taking a reset. When a reset occurs (usually triggered by an illegal or out of range memory access), the audit which was running at the time is aborted, and is then rescheduled to run in AUZAP mode to provide recovery of related data left inconsistent as a result of

	the reset.
Possible Error Effect:	The audit may continue to cause memory access violations, increasing the probability of drastic recovery action and preventing the audit from checking for further errors.
Corrective Action Taken:	The data which caused the reset is initialized by the next entry of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

IFD-A**Software Release:** 5E10 - 5E11**1. INTERNAL FRAME DESCRIPTOR (IFD) AUDIT**

IFDs are used in packet switching to keep track of frame ownership. Each IFD owns one or more buffer descriptors (BDs) which are used to control ownership of the buffers which contain the packet information. There are two types of IFDs:

- (1) IFDs linked to logical link control blocks (LLCBs) or quad-linked packet switched LLCBs (QLLCBs).
- (2) IFDs linked to arbitrary structures based on the IFD.client and IFD.clientidx fields.

These two types are mutually exclusive. That is, a given audit environment will include at most one of the two types. Error reports specific to one of the types have been noted as such below. Errors for which the associated checks differ between the types have also been noted.

Finally, the BD linkage field in an IFD may be implemented as either an offset or as a pointer. Since this does not affect the checks done, the implementation is not mentioned below.

2. DATA STRUCTURES

IFDs using the *llcb* field to track ownership may be owned by elements of PSllcb[] or QLllcb[]. IFDs whose ownership is based on the *client* and *clientidx* fields may be owned by elements of PSllcb[] or PVvccb[]. Either type may be owned by the free list headcell (PSrifdhd).

3. AUDIT CHECKS

The following list gives a basic indication of the order of the checks made by this audit. If the audit is being (re)entered:

ZAP = Check for system interrupt handling.

If the audit is completing:

STATUS = Check for integrity monitor invocation handling.

If the audit is checking an IFD:

BADOWNER = Check of IFD owner type.

BDLINK = Validity check of the IFD's BD linkage field.

IFD_LINK = Linkage check between an IFD and its owner.

OUTOFRANGE = Range check IFD ownership fields.

PT2PT = Linkage check between IFDs.

If the audit is recovering an IFD:

HD_CELL = Validity check of PSrifdhd.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	This check only applies to IFDs using fields <i>client</i> and <i>clientidx</i> for IFD to owner linkages. IFD.client (owner type) is not valid.
Possible Error Effect:	IFD handling code may assert. Otherwise, the IFD is probably lost and will contribute to the possibility of IFD resource failures.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data</i> =	IFD.client (the invalid IFD owner type).
<i>Good Data</i> =	IFD.clientidx.
<i>Logical Index</i> =	IFD number (index into PSifd[]).
<i>Dump</i> =	The IFD.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	An internal frame descriptor's (IFD's) buffer descriptor (BD) linkage field is invalid. That is, either the IFD is idle (linked to PSrfdhd[]) and the BD field is not PSBDNULL or the IFD is not idle and the BD field does not contain a valid value.
Possible Error Effect:	If the IFD is idle, it may be preventing valid users from accessing the BD. If this happened often enough, the processor could experience BD resource failures. Otherwise, IFD user code may assert or otherwise fail when handling the IFD.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data</i> =	IFD to IFD linkage fields. The top 16 bits contain IFD.next_ifd. The lower 16 bits contain IFD.back_ifd.
<i>Good Data</i> =	IFD owner information (IFD.llcb or the top 16 bits contain IFD.client and the lower 16 bits contain IFD.clientidx).
<i>Logical Index</i> =	IFD number (index into PSifd[]).
<i>Dump</i> =	NA.

Table 3 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The IFD free list headcell (PSrfdhd) is corrupt. That is, one or more of the following is true: - PSrfdhd.first or PSrfdhd.last are out of range ($\geq$ PSRIFDS). (< PSRIFDS) but PSifd[PSrfdhd.first].back_ifd != PSIFDNULL or (depending on IFD type) llcb != PSLLCBNULL or <i>clientidx</i> !=

	PSICLNULL. ($< \text{PSRIFDS}$) but $\text{PSifd}[\text{PSrfdhd.first}].\text{next\_ifd} \neq \text{PSIFDNULL}$ or (depending on IFD type) $\text{llcb} \neq \text{PSLLCBNULL}$ or $\text{clientidx} \neq \text{PSICLNULL}$.
Possible Error Effect:	Allocation of IFDs may not be possible. This would cause all frames to be dropped.
Corrective Action Taken:	The audit rebuilds the IFD free list by clearing the headcell and linking all IFDs whose (depending on IFD type) llcb field = PSLLCBNULL or client field = PSINOCIENT .
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Minimum valid IFD index.
<i>Bad Data</i> =	Number of IFDs (PSRIFDS).
<i>Good Data</i> =	Null IFD index (PSIFDNULL).
<i>Logical Index</i> =	The IFD under audit when the error was detected.
<i>Dump</i> =	The IFD free list headcell (PSrfdhd).

Table 4 Error Code: IFD_LINK

Error Code:	IFD_LINK
Error Description:	There is an IFD to IFD linkage failure. At least one of the following is true: - IFD.next_ifd is in range ($< \text{PSRIFDS}$) but $\text{PSifd}[\text{IFD.next\_ifd}].\text{back\_ifd} \neq \text{IFD index}$. - IFD.$\text{back\_ifd}$ is in range ($< \text{PSRIFDS}$) but $\text{PSifd}[\text{IFD.back\_ifd}].\text{next\_ifd} \neq \text{IFD index}$. - The owner information between two linked IFDs does not match. Either: - IFD.$\text{llcb} \neq [\text{next or back}] \text{IFD.llcb}$ or - IFD.client or $\text{clientidx} \neq [\text{next or back}]$
Possible Error Effect:	The list of IFDs is broken. If the current list is the idle list, the processor might experience IFD resource failures. Otherwise, IFD user code may assert or otherwise fail while trying to process the frame associated with this IFD.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data</i> =	IFD owner linkage fields. The top 16 bits contain the owner's f_ifd field. The lower 16 bits contain the owner's i_ifd field.
<i>Good Data</i> =	IFD owner information (IFD.llcb or the top 16 bits contain IFD.client and the lower 16 bits contain IFD.clientidx).
<i>Logical Index</i> =	IFD number (index into $\text{PSifd}[]$).
<i>Dump 1</i> =	The IFD.
<i>Dump 2</i> =	If the PSIFD.next_ifd is $< \text{PSRIFDS}$, this will be the IFD indexed by that value. Otherwise the dump will not appear. Note that the error address printed in the dump message will be 0.
<i>Dump 3</i> =	If the PSIFD.back_ifd is $< \text{PSRIFDS}$, this will be the IFD indexed by that value. Otherwise the dump will not appear. Note that the error address printed in the dump message will be 1.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The IFD owner information is out of range. That is, one of the following is true: • IFD.llcb is not equal to PSLLCBNULL but is greater than or equal to both PSRLLCBS and QLNLLCBS. • IFD.client is a valid type but the associated IFD.clientidx value is out of range (the valid values depend on the <i>client</i> type).
Possible Error Effect:	The IFD is not available for use. If enough IFDs became unavailable, the processor might experience IFD resource failures.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data</i> =	IFD to IFD linkage fields: The top 16 bits contain IFD.next_ifd. The lower 16 bits contain IFD.back_ifd.
<i>Good Data</i> =	IFD owner information: IFD.llcb or The top 16 bits contain IFD.client. The lower 16 bits contain IFD.clientidx.
<i>Logical Index</i> =	IFD number (index into PSifd[]).
<i>Dump</i> =	NA.

Table 6 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	One or more of the following is true about an IFD: • IFD.next_ifd = PSIFDNULL but the IFD is not the last IFD on the list according to the owning structure's <i>l_ifd</i> field. • IFD.back_ifd = PSIFDNULL but the IFD is not the first IFD on the list according to the owning structure's <i>f_ifd</i> field. • One or both of the IFD's IFD linkage fields (PSIFD.next_ifd and PSIFD.back_ifd) is out of range ($\neq$ PSIFDNULL and $\geq$ PSRIFDS).
Possible Error Effect:	The IFD is not available for use. If enough IFDs became unavailable, the processor might experience IFD resource failures.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data</i> =	IFD to IFD linkage fields. The top 16 bits contain IFD.next_ifd. The lower 16 bits contain IFD.back_ifd.
<i>Good Data</i> =	IFD owner information (IFD.llcb or the top 16 bits contain IFD.client and the lower 16 bits contain IFD.clientidx).
<i>Logical Index</i> =	IFD number (index into PSifd[]).
<i>Dump 1</i> =	The IFD.
<i>Dump 2</i> =	If the PSIFD.next_ifd is $<$ PSRIFDS, this will be the IFD indexed by that value. Otherwise the dump will not appear. Note that the error address printed in the dump message will be 0.
<i>Dump 3</i> =	If the PSIFD.back_ifd is $<$ PSRIFDS, this will be the IFD indexed by that value. Otherwise the dump will not appear. Note that the error address printed in the dump message will be 1.

Table 7 Error Code: STATUS

Error Code:	STATUS
Error Description:	This check only applies to IFDs using fields <i>client</i> and <i>clientidx</i> for IFD to owner linkages. Integrity monitor has invoked the IFD audit because of a packet switching internal frame descriptor (PSIFD) resource problem.
Possible Error Effect:	Frames can not be passed if IFDs are not available.
Corrective Action Taken:	The audit determines which PSRLCB is the heaviest PSIFD user and idles all PSIFDs belonging to the PSRLCB in question (PSIFD.client = PSICRLCB and PSIFD.clientidx = PSRLCB index).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The number of IFDs being used by the maximum user.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the maximum user (index into PSrlcb[]).
<i>Dump</i> =	The maximum user.

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	If the IFD index is in range (< PSRIFDS), the audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The number of IFDs (PSRIFDS).
<i>Logical Index</i> =	IFD number (index into PSifd[]). This value may be out of range ($\geq$ PSRIFDS).
<i>Dump</i> =	The IFD if the IFD index is in range. Otherwise, NA.

IFD-B

Software Release: 5E12 and later

1. INTERNAL FRAME DESCRIPTOR (IFD) AUDIT

IFDs are used in packet switching to keep track of frame ownership. Each IFD owns one or more buffer descriptors (BDs) which are used to control ownership of the buffers which contain the packet information.

IFDs may be linked to arbitrary structures based on the IFD.client and IFD.clientidx fields.

Finally, the BD linkage field in an IFD may be implemented as either an offset or as a pointer. Since this does not affect the checks done, the implementation is not mentioned below.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The IFD free list headcell (PSrfdhd) contains the index of an IFD (PSrfdhd.first or PSrfdhd.last) which is not idle (IFD.client != PSICLNULL or IFD.clientidx != PSICLNULLIDX).
Possible Error Effect:	IFD handling code may assert. Packet traffic may be lost if IFDs cannot be allocated. This may in turn cause processor resource overload reports.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of the corrupted IFD list.
<i>Bad Data =</i>	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Good Data =</i>	The number of IFDs found in the list.
<i>Logical Index =</i>	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Dump 1 =</i>	IFD linkage information for this client.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit was unable to retrieve IFD linkage information for the IFD free list headcell (PSrfdhd).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit rebuilds the IFD free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The top 16 bits contain the count of IFDs found on the list. The lower 16 bits contain the index to the failed IFD list.

<i>Bad Data</i> =	An internal value indicating the reason the linkage retrieval failed.
<i>Good Data</i> =	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Logical Index</i> =	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Dump</i> =	NA.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	An IFD either owns a BD when it should not or does not own a BD when it should.
Possible Error Effect:	Either one or more BDs are being held when they should not be, possibly contributing to resource overload, or, data associated with the frame may be lost.
Corrective Action Taken:	The audit idles the IFD and, depending on the IFD owner type, may take additional recovery on that owner.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The IFD's BD field (IFD.bd or IFD.bdptr).
<i>Bad Data</i> =	The top 16 bits contain the IFD.next_ifd field value. The lower 16 bits contain the IFD.back_ifd field value.
<i>Good Data</i> =	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Logical Index</i> =	IFD number (index into PSifd[]).
<i>Dump 1</i> =	The IFD.
<i>Dump 2</i> =	The IFD indexed by the current IFD.next_ifd field (if available).
<i>Dump 3</i> =	The IFD indexed by the current IFD.back_ifd field (if available).
<i>Dump 4</i> =	The IFD owner (if available).

Table 4 Error Code: HEAD

Error Code:	HEAD
Error Description:	Corruption has been detected in the owner linkage of an IFD list.
Possible Error Effect:	IFD resources may become lost.
Corrective Action Taken:	The audit recovers IFD list linkages and, depending on the IFD owner type, may take additional recovery on that owner.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the corrupted IFD list.
<i>Bad Data</i> =	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Good Data</i> =	The number of IFDs found in the list.
<i>Logical Index</i> =	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Dump 1</i> =	IFD linkage information for this client.
<i>Dump 2</i> =	There may be an optional second dump containing debugging information.

Table 5 Error Code: IFD_LINK

Error Code:	IFD_LINK
Error Description:	There is an IFD to IFD linkage failure. At least one of the following is true: • IFD.next_ifd is in range (< PSRIFDS) but PSifd[IFD.next_ifd].back_ifd != IFD index. • IFD.back_ifd is in range (< PSRIFDS) but PSifd[IFD.back_ifd].next_ifd != IFD index. • The owner information between two linked IFDs does not match (IFD.client or clientidx != [next or back] IFD.client or clientidx).
Possible Error Effect:	The list of IFDs is broken. If the current list is the idle list, the processor might

	experience IFD resource failures. Otherwise, IFD user code may assert or otherwise fail while trying to process the frame associated with this IFD.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data =</i>	IFD owner linkage fields. The top 16 bits contain the owner's <i>f_ifd</i> field. The lower 16 bits contain the owner's <i>l_ifd</i> field.
<i>Good Data =</i>	The top 16 bits contain IFD.client. The lower 16 bits contain IFD.clientidx.
<i>Logical Index =</i>	IFD number (index into PSifd[]).
<i>Dump 1 =</i>	The IFD.
<i>Dump 2 =</i>	The IFD indexed by the current IFD.next_ifd field (if available).
<i>Dump 3 =</i>	The IFD indexed by the current IFD.back_ifd field (if available).
<i>Dump 4 =</i>	The IFD owner (if available).

Table 6 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	An IFD list which should be empty is not empty.
Possible Error Effect:	IFD resources may become lost.
Corrective Action Taken:	The audit idles the IFD and, depending on the IFD owner type, may take additional recovery on that owner.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index of the corrupted IFD list.
<i>Bad Data =</i>	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Good Data =</i>	The number of IFDs found in the list.
<i>Logical Index =</i>	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Dump 1 =</i>	IFD linkage information for this client.
<i>Dump 2 =</i>	There may be an optional second dump containing debugging information.

Table 7 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit was unable to retrieve IFD list information for the owner of this IFD (represented by IFD.client and IFD.clientidx).
Possible Error Effect:	The IFD is not available for use. If enough IFDs became unavailable, the processor might experience IFD resource failures.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data =</i>	An internal value indicating the reason the linkage retrieval failed.
<i>Good Data =</i>	The top 16 bits contain IFD.client. The lower 16 bits contain IFD.clientidx.
<i>Logical Index =</i>	IFD number (index into PSifd[]).
<i>Dump =</i>	The IFD.

Table 8 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	One or more of the following is true about an IFD: • IFD.next_ifd = PSIFDNULL but the IFD is not the last IFD on the list according to the owning structure's <i>l_ifd</i> field. • IFD.back_ifd = PSIFDNULL but the IFD is not the first IFD on the list according to the owning structure's <i>f_ifd</i> field.

	One or both of the IFD's IFD linkage fields (IFD.next_ifd and IFD.back_ifd) is out of range (!= PSIFDNULL and ≥ PSRIFDS).
Possible Error Effect:	The IFD is not available for use. If enough IFDs became unavailable, the processor might experience IFD resource failures.
Corrective Action Taken:	The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The IFD BD linkage field (IFD.bd or IFD.bdptr).
<i>Bad Data =</i>	IFD to IFD linkage fields. The top 16 bits contain IFD.next_ifd. The lower 16 bits contain IFD.back_ifd.
<i>Good Data =</i>	The top 16 bits contain IFD.client. The lower 16 bits contain IFD.clientidx.
<i>Logical Index =</i>	IFD number (index into PSifd[]).
<i>Dump 1 =</i>	The IFD.
<i>Dump 2 =</i>	The IFD indexed by the current IFD.next_ifd field (if available).
<i>Dump 3 =</i>	The IFD indexed by the current IFD.back_ifd field (if available).
<i>Dump 4 =</i>	The IFD owner (if available).

Table 9 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve IFD list information for the IFD free list headcell (PSrifdhd).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit rebuilds the IFD free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The number of IFDs found on the IFD list.
<i>Bad Data =</i>	An internal value indicating the reason the linkage retrieval failed.
<i>Good Data =</i>	The maximum number of IFDs allowed on an IFD list.
<i>Logical Index =</i>	The top 16 bits contain the IFD.client field value. The lower 16 bits contain the IFD.clientidx field value.
<i>Dump =</i>	There may be an optional dump containing debugging information.

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	If the index indicating the last IFD that was checked is in range (< PSRIFDS), The audit unlinks the IFD from any list it is in, sets all fields to their default values and links it to the IFD idle list. Any BDs that had been associated with this IFD will be recovered by the related BD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The number of IFDs (PSRIFDS).
<i>Logical Index =</i>	IFD number (index into PSifd[]). This value may be out of range (≥ PSRIFDS).
<i>Dump =</i>	The IFD if the IFD index is in range. Otherwise, NA.

IMIIA

Software Release: 5E10 and later

1. INTERRUPT INHIBIT ADMINISTRATOR (IMIIA) AUDIT

The interrupt inhibit administrator (IIA) is a program that controls the state of the inhibit bits for all the hardware in the switching modules (SMs). The interrupt inhibit administrator is called by fault recovery, the maintenance request administrator (MRA), or by manual request using an INH:INTR,[bit], ALW:INTR,[bit], RST:[circuit], or a RMV:[circuit] input message. Refer to the Input Messages manual (235-600-700). The state of the inhibit bits is either inhibit (INH) or allow (ALW). Specifically, the IIA controls 38 bits on the following eight registers: the reset mask register, the hardware reset source mask register, the software reset source mask register, the data link interface (DLI) error mask register, the three interrupt controller registers, and the write protect control register.

The basic states of hardware in the duplex SM environment are active, standby, out-of-service, and unavailable. The switching module interrupt inhibit administrator audit (IMIIA) checks both sides of the switching module (SM) if the SM's state is active/standby. If the SM's state is either active/out-of-service or active/unavailable, the audit checks only the active side, except for the mate controller ready-time out bit (MMTRT). In this case, an interrupt may be allowed on an out-of-service circuit. The audit does not alter the hardware configuration; it alters the hardware interrupt mask registers to reflect the hardware state.

The IMIIA audit also performs interrupt status checks on the remote clock (RCLK) circuit. The clock is a duplex circuit. Regardless of the number of active processors, the audit checks both clock sides.

This audit is part of the switch maintenance audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

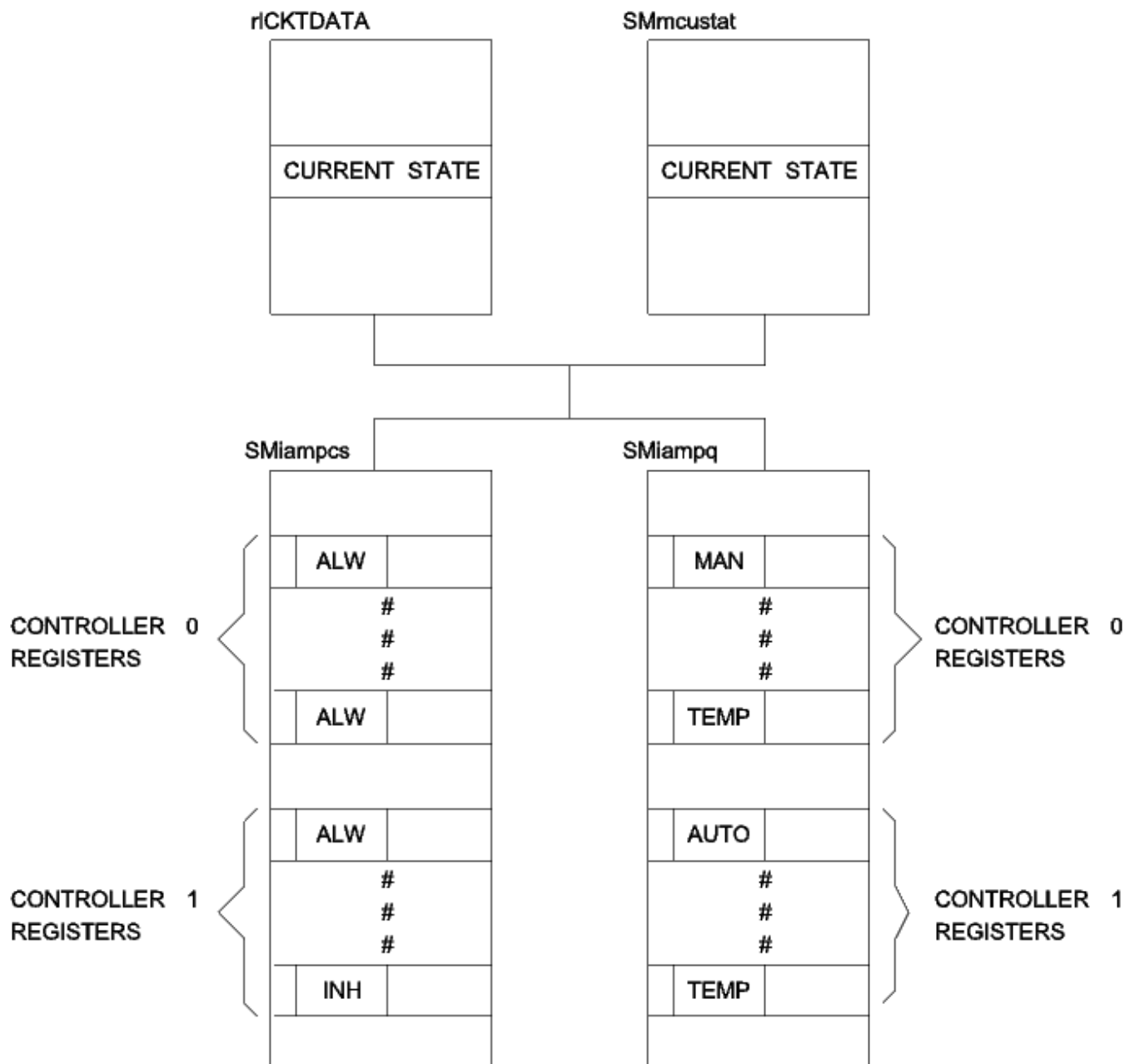


Figure 1 (IMIIA-1) IIA Arrays

2. DATA STRUCTURES

Figure 1 shows the two arrays in global data checked by the IMIIA audit: the inhibit administrator module processor current state array (SMiampcs) and the inhibit administrator module processor qualifiers array (SMiampq). The SMiampcs array is 16 words long. Words zero through seven reflect the state of the eight registers for controller zero; words eight through 15 contain the states of the eight registers for controller one. This array changes with the current state of the hardware. The qualifiers array (SMiampq), which describes how each inhibit bit was set, is 76 words long. Words zero through 37 are for the 38 bits in the eight registers for controller zero, and words 38 through 75 are for the corresponding bits for controller one. The audit checks one write-protected relation: the circuit data relation (RLCKTDATA). This relation may be modified only by using recent change procedures.

The audit also checks one variable array, the switch maintenance current state array (SMmccustat).

3. AUDIT CHECKS

The IMIIA audit checks the consistency between the software image of the interrupt masks and the hardware state of the interrupt masks. Initially, the audit allows parity checking on the time slot interchanger (TSI). Then the audit performs the following checks on the SM, DLI, and control interface (CI) interrupt masks.

- (a) The audit first determines which side is currently executing (0 or 1) to know which half of the software image array should be used.
- (b) The audit determines if the mate side should be audited. If the state is out-of-service or unavailable (OOS or UNAV), no audit is performed on the mate.
- (c) Then the audit performs the following checks for each of the 38 bits on the side to be audited.
 - (1) The audit reads the appropriate hardware register to get the actual state of the bit (ALW or INH).
 - (2) The audit reads the SMiampcs array to obtain the software image for the same bit.
 - (3) The audit checks the SMiampq array and ensures that it contains one of three acceptable values: TEMP, AUTO, or MAN. These qualifiers describe how the inhibit bit was set: either temporarily (TEMP) while diagnostics are running, automatically (AUTO) during fault recovery, or manually (MAN) as requested by maintenance personnel. Any other value will cause an audit error report, and the value is changed to AUTO.
 - (4) The audit calls a function to calculate the normal state for the bit. The function called depends on the bit involved. For CI bits, the audit checks rICKTDATA to determine if the corresponding CI is operational. The normal state for an operational CI is allow. For all other cases, the normal state is inhibit. For SM and DLI bits, the audit checks the SMmccustat array to determine if the circuit is active or standby. The normal state for an active or standby SM or DLI is allow. For all other cases, the normal state is inhibit.
 - (5) If no errors are detected for the bit, the audit checks the TEMP/INHIBIT state. This state should only be detected by the audit if the unit is being actively diagnosed. If this state is detected, the audit reports it and then continues. If this state is not detected, the audit checks if the hardware value of the mask matches the software image of the mask. When the audit runs in a remote switch module (RSM) environment, it performs another check on DLI bits. The audit verifies that the FI hardware value is consistent with the DLI software image. If there is an inconsistency, the audit reports it and corrects it.

On each call to the remote clock check routine, the audit determines whether the hardware interrupt status for the clock and its child circuit (RCXC, RCOXC, RCOSC, or RCREP) is consistent with the switch maintenance equipment status table (rISMEST) interrupt information.

4. ERROR HANDLING

If an inconsistency exists between the normal state and the software image, the audit changes the hardware interrupt mask register to agree with the normal state. One exception is the MMTRT bit. If the software image for MMTRT is inhibit, the audit reports the error and corrects it. This bit should never be masked.

The audit then updates the appropriate bits for the circuit in the current state array (SMiampcs) to that state. The audit proceeds to the qualifiers array where it updates the qualifier to AUTO. Once the registers and arrays are updated, the software and actual hardware configuration are consistent, and the audit proceeds to the next bit.

Remote clock errors are corrected by calling the interrupt administration peripheral refresh routine, which updates the hardware status of the RCLK PICB and child circuit control communication buffer (CCB) with the interrupt status information in rSMEST provided that the parent remote clock is not out of service.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CR_DATA

Error Code:	CR_DATA
Error Description:	This type of error indicates that some or all of the current data related to MCTSI critical regions (SMcrnt and the structure SMcrdata) is not in the expected state. SMcrnt should be equal to zero at this point which indicates that no critical regions are in effect at this time. Also, SMcrdata.who should be set to SMCRWNULL (0) and SMcrdata.level should be set to SMCRNULL (0). SMcrdata.data_1 holds additional information passed into SMcrbegin() by the caller.
Possible Error Effect:	Various critical region asserts may be seen indicating such problems as a user asking for a critical region when one is already invoked (SMCRCOUNT) or a user asking to end a critical region when they don't have one (SMCRBEND). Until the corruption is corrected, a user may be unable to obtain a critical region. Also, much of the SM hardware error detection may be disabled.
Corrective Action Taken:	The critical region data and the associated hardware are refreshed to the expected state which indicates that no critical region is in effect.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the global structure SMcrdata.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The bad data dumped out with the audit error report should be interpreted as follows (each letter represents 4 bits of data): vvwwxxxxxyyyzzzz where: vv = SMcrdata.who. ww = SMcrdata.level. xxxx = SMcrdata.intlev. yyyy = SMcrnt. zzzz = 2 bytes of padding (ignore).

Table 2 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	This type of error indicates that the software current state image, which is kept in SMiampcs[] does not agree with the hardware mask bit state. Further comparison of each state to the normal state showed that the hardware state was correct and that the software current state image was incorrect.
Possible Error Effect:	Unexpected interrupts.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image

	number plus the image offset, is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT). The normal state depends upon the state of the circuit corresponding to the bit number. The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMiampcs[], indexed by the image number plus the image offset.
<i>Bad Data</i> =	Current bit state.
<i>Good Data</i> =	Normal bit state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 3 Error Code: MP_HDW_IA

Error Code:	MP_HDW_IA
Error Description:	This type of error indicates that the hardware mask bit state does not agree with the software current state image, which is kept in SMiampcs[]. Further comparison of each state to the normal state showed that the software current state image was correct and that the hardware mask bit state was in error.
Possible Error Effect:	Unexpected interrupts.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image number plus the image offset, is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT). The normal state depends upon the state of the circuit corresponding to the bit number. The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Hardware mask bit state.
<i>Good Data</i> =	Normal bit state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 4 Error Code: MM_ALWOOS

Error Code:	MM_ALWOOS
Error Description:	This type of error indicates that the current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset, is allow (SMALLOW) while the state of the circuit corresponding to the bit number is out of service (OOS). This combination of alw/oos is illegal because interrupts should never be allowed on a nonoperational circuit. This error code corresponds only to the module controller mask bits.
Possible Error Effect:	Unexpected interrupts from an out-of-service, unequipped, or a prior-to-out-of-service circuit.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset, is updated to the normal state of inhibit (SMINHIBIT). The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset, is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMiampcs[], indexed by the image number plus the register offset.
<i>Bad Data</i> =	Current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset.
<i>Good Data</i> =	Normal state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 5 Error Code: M_NOTINH

Error Code:	M_NOTINH
Error Description:	This type of error indicates that a mask bit that is always inhibited is in the allow (SMALLOW) or an unknown state.
Possible Error Effect:	Unexpected interrupts from an out-of-service, unequipped, or growth circuit.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset, is updated to the normal state of inhibit (SMINHIBIT). The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset, is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMiampcs[], indexed by the image number plus the register offset.
<i>Bad Data</i> =	Current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset.
<i>Good Data</i> =	Normal state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 6 Error Code: M_TMPINH

Error Code:	M_TMPINH
Error Description:	This type of error indicates that the current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset, is temporary (SMIATEMP) and the current state, which is kept in SMiampcs[] and indexed by the image number plus the image offset, is inhibit (SMINHIBIT). The combination of temp/inh is illegal because it should never be seen by anything but a running diagnostic phase, which never takes a real time break. This error code pertains only to the module controller mask bits.
Possible Error Effect:	Interrupts are blocked unnecessarily.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image number plus the image offset, is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT). The normal state depends upon the state of the circuit corresponding to the bit number. The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMiampcs[], indexed by the image number plus the image offset.
<i>Bad Data</i> =	Current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset.
<i>Good Data</i> =	Normal state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 7 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	This type of error indicates that the current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset, is of an unknown type (not SMIA_AUTO, SMIA_MAN, or SMIATEMP). This error code pertains strictly to module controller mask bits.
Possible Error Effect:	Could cause asserts and rejection of all IIA requests to inhibit or allow interrupts on this circuit.
Corrective Action Taken:	The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset is updated to auto (SMIA_AUTO).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMiampq[], indexed by the bit number plus the bit offset.

<i>Bad Data</i> =	Current state qualifier in SMiampq[], indexed by the bit number plus the bit offset.
<i>Good Data</i> =	New qualifier (SMIA_AUTO).
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 8 Error Code: PROC_IA

Error Code:	PROC_IA
Error Description:	This type of error indicates that a mask bit that is always allowed is in the inhibit (SMINHIBIT) state. Currently this error code only pertains to the module controller mate ready timeout mask.
Possible Error Effect:	Upon switching to the mate module controller system could roll on time out errors.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset, is updated to the normal state of allow (SMALLOW). The current state qualifier, which is kept in SMiampq[] and indexed by the bit number plus the bit offset, is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMiampcs[], indexed by the image number plus the register offset.
<i>Bad Data</i> =	Current state, which is kept in SMiampcs[] and indexed by the image number plus the register offset.
<i>Good Data</i> =	Normal state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 9 Error Code: P_ALWOOS

Error Code:	P_ALWOOS
Error Description:	This type of error indicates that the Equipment Status Table current state is allow (SMALLOW), while the state of the circuit is either out of service (OOS) or prior to out of service (PROOS). This combination is illegal because interrupts should never be allowed on an OOS or PROOS circuit. This error code pertains only to the peripheral audit function.
Possible Error Effect:	Unwanted interrupts during a diagnostic, rolling interrupts, and degrading machine performance.
Corrective Action Taken:	The Equipment Status Table current state is updated to the normal state of inhibit (SMINHIBIT). The Equipment Status Table current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware and the software images are refreshed to match the current state of the Equipment Status Table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the Equipment Status Table tuple.
<i>Bad Data</i> =	Current state.
<i>Good Data</i> =	Normal state.
<i>Logical Index</i> =	Circuit name in the tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: P_HDW_IA

Error Code:	P_HDW_IA
Error Description:	This type of error indicates that the hardware mask bit state of the RLI does not agree with the software current state image of the RLI, which is kept in SMiampcs[].
Possible Error Effect:	Unexpected interrupts.
Corrective Action Taken:	The current state, which is kept in SMiampcs[] and indexed by the image number plus the image offset, is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT). The normal state depends upon the state of the circuit corresponding to the bit number. The current state qualifier, which is

	kept in SMiampq[] and indexed by the bit number plus the bit offset is updated to auto (SMIA_AUTO). The hardware is refreshed to match the current state of the RLI in SMiampcs[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	RLI mask bit state.
<i>Good Data</i> =	Normal bit state.
<i>Logical Index</i> =	Bit number plus the bit offset.
<i>Dump</i> =	NA.

Table 11 Error Code: P_TMPINH2

Error Code:	P_TMPINH2
Error Description:	This type of error indicates that the Equipment Status Table current state qualifier is temporary (SMIATEMP) and the current state is inhibit (SMINHIBIT). This combination is illegal because it should only be seen while running a diagnostic phase, which never takes a real time break with temporary inhibits in place. This condition probably means a diagnostic phase was purged before it had a chance to clean up from temp/inh to temp/alw. This error code only pertains to the peripheral audit function.
Possible Error Effect:	Interrupts are blocked unnecessarily.
Corrective Action Taken:	The Equipment Status Table current state is updated to the normal state of allow (SMALLOW). The Equipment Status Table current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware and the software images are refreshed to match the current state of the Equipment Status Table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the Equipment Status Table tuple.
<i>Bad Data</i> =	Current state qualifier.
<i>Good Data</i> =	New qualifier.
<i>Logical Index</i> =	Circuit name in the tuple.
<i>Dump</i> =	NA.

Table 12 Error Code: P_UNKQUAL

Error Code:	P_UNKQUAL
Error Description:	This type of error indicates that the Equipment Status Table current state qualifier is of an unknown type (not SMIA_AUTO, SMIA_MAN, or SMIATEMP). This error code pertains strictly to the peripheral audit function.
Possible Error Effect:	Cause asserts, and reject all IIA requests for the circuit.
Corrective Action Taken:	The Equipment Status Table current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The Equipment Status Table current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware and the software images are refreshed to match the current state of the Equipment Status Table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the Equipment Status Table tuple.
<i>Bad Data</i> =	Current state qualifier.
<i>Good Data</i> =	New qualifier.
<i>Logical Index</i> =	Circuit name in the tuple.
<i>Dump</i> =	NA.

Table 13 Error Code: P_UNKSTAT2

Error Code:	P_UNKSTAT2
Error Description:	This type of error indicates that the Equipment Status Table current state is of an unknown type (not SMALLOW, or SMINHIBIT). This error code only pertains to the peripheral audit function.
Possible Error Effect:	Cause asserts, and reject IIA calls for this circuit.

Corrective Action Taken:	The Equipment Status Table current state is updated to the normal state of allow (SMALLOW). The Equipment Status Table current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware and the software images are refreshed to match the current state of the Equipment Status Table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the Equipment Status Table tuple.
<i>Bad Data</i> =	Current state.
<i>Good Data</i> =	Normal state.
<i>Logical Index</i> =	Circuit name in the tuple.
<i>Dump</i> =	NA.

17. IN - J

INPCB-A

Software Release: 5E12 only

1. ETHERNET INTERNET PROTOCOL CONTROL BLOCK (INPCB) AUDIT

Internet protocol is used to transmit and receive data through an ethernet network. The INPCBs are used to represent per-connection information such as foreign IP addresses and port numbers. They are either idle and are linked to the idle list or linked and are linked to a SOCKET and possibly to a transmission control protocol control block (TCPCB). The audit verifies linkages among CMinpcb[], CMtcpcb[], CMsocket[] and CMidleinpcb and verifies data within every entry in CMinpcb[].

2. DATA STRUCTURES

The following data are used by the INPCB audit:

CMinpcb[] = Internet protocol control block.

CMtcpcb[] = TCP control block.

CMsocket[] = Socket protocol control block.

CMidleinpcb = Internet control block idle list.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The next field in the current CMinpcb[] entry is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	busy_list.inp_next.
<i>Bad Data =</i>	busy_list.inp_prev.
<i>Good Data =</i>	CMinpcb[index].inp_next.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 2 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The CMinpcb[] entry is the first member in a linked list of busy entries. One of the following conditions has occurred: • CMinpcb[index].inp_next is not within range, • CMinpcb[index].inp_next equals the index into CMinpcb[],

	• CMinpcb[CMinpcb[index].inp_next].inp_prev does not equal the index into CMinpcb[] or • CMinpcb[CMinpcb[index].inp_next].inp_head_ptr does not equal inp_head_ptr of the owner.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMinpcb[CMinpcb[index].inp_next].inp_prev.
<i>Good Data =</i>	CMinpcb[CMinpcb[index].inp_next].inp_head_ptr.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 3 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The CMinpcb[] entry is the last member in a linked list of busy entries. One of the following conditions has occurred: • CMinpcb[index].inp_prev is not within range, • CMinpcb[index].inp_prev equals the index into CMinpcb[], • CMinpcb[CMinpcb[index].inp_prev].inp_next does not equal the index into CMinpcb[] or • CMinpcb[CMinpcb[index].inp_prev].inp_head_ptr does not equal inp_head_ptr of the owner.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMinpcb[CMinpcb[index].inp_prev].inp_head_ptr.
<i>Good Data =</i>	CMinpcb[CMinpcb[index].inp_prev].inp_next.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 4 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The audit finds the MEPIPE is OOS but its entry in CMinpcb[] is marked that it is linked when it should not be linked.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMinpcb[index].inp_socket.
<i>Bad Data =</i>	CMinpcb[CMinpcb[index].inp_socket].so_pcb.
<i>Good Data =</i>	CMinpcb[index].inp_next.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump 1 =</i>	Contents of CMinpcb[index].

<i>Dump 2 =</i>	Contents of CMscddb.state.

Table 5 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	There is no CMincpb[] entry linked to this busy list owner but its <i>inp_head_ptr</i> field is not the null value (CM_NULL). The entry in CMatpmbuff[] can be linked to one of three busy lists. The value in <i>Good Data</i> indicates to which busy list the error refers. 0 = CMintcp, 1 = CMinudp and 2 = CMinrip.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit sets the <i>inp_head_ptr</i> field of the owner to CM_NULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	<busy>[side].inp_head_ptr.
<i>Good Data =</i>	1, 2 or 3.
<i>Logical Index =</i>	Side of busy list which has the error.
<i>Dump =</i>	Contents of CMinpcb[index].

Table 6 Error Code: BADPROTOCOL

Error Code:	BADPROTOCOL
Error Description:	For protocols other than CMIPTCP, the <i>inp_pccb</i> field should be its null value but it is not.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMinpcb[index].inp_ppcb.
<i>Bad Data =</i>	CMsocket[CMinpcb[index].inp_socket].so_protocol.
<i>Good Data =</i>	CMinpcb[index].inp_socket.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 7 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	An entry in CMinpcb[] should be linked to an entry in CMsocket[] if the <i>CMscdb.state</i> is not CMCHSTOFF. One of the following has occurred: • CMinpcb[index].inp_qindx does not equal CMNIL_NDX, • CMinpcb[index].inp_socket is out of range, • CMinpcb[index].inp_next is out of range, • CMinpcb[index].inp_prev is out of range or • CMsocket[CMinpcb[index].inp_socket].so_pcb does not equal the index into CMinpcb[].

Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_socket.
<i>Bad Data</i> =	CMinpcb[index].inp_next.
<i>Good Data</i> =	CMsocket[CMinpcb[index].inp_socket].so_pcb.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 8 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The state of an entry in CMinpcb[] is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMinpcb[index].inp_cbstate.
<i>Good Data</i> =	CMINCBMAX
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 9 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	Either one or both indices in the owner of the current CMinpcb[] which have the head and tail of the linked CMinpcb[] entries is out of range. The entry in CMatpmbuff[] can be linked to one of three busy lists. The value in <i>Good Data</i> indicates to which busy list the error refers. 0 = CMintcp. 1 = CMinudp. 2 = CMinrip.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit repairs the linked list by linking around the bad entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	<busy>[side].inp_next.
<i>Bad Data</i> =	<busy>.inp_prev.
<i>Good Data</i> =	1, 2 or 3.
<i>Logical Index</i> =	Side of busy list which has the error.
<i>Dump</i> =	Contents of CMinpcb[<busy>].

Table 10 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The current CMinpcb[] entry is linked to a busy list. One of the following has occurred: CMinpcb[<owner>].inp_next].inp_head_ptr does not equal address of <owner>.

	• CMinpcb[<owner>.inp_next].inp_prev does not equal CMNIL_NDX, • CMinpcb[<owner>.inp_prev].inp_head_ptr does not equal address of <owner> or • CMinpcb[<owner>.inp_prev].inp_next does not equal CMNIL_NDX. The entry in CMatpmbuff[] can be linked to one of three busy lists. The value in <i>Good Data</i> indicates to which busy list the error refers. 0 = CMintcp, 1 = CMinudp and 2 = CMinrip.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit repairs the linked list by linking around the bad entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[<busy>[side].inp_next].inp_prev.
<i>Bad Data</i> =	CMinpcb[<busy>[side].inp_next].inp_next.
<i>Good Data</i> =	1, 2 or 3.
<i>Logical Index</i> =	Side of busy list which has the error.
<i>Dump 1</i> =	Contents of <busy>[side].
<i>Dump 2</i> =	<busy>[side].inp_next.
<i>Dump 3</i> =	<busy>[side].inp_prev.

Table 11 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The CMinpcb[] is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	CMinpcb[index].inp_cbstate.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 12 Error Code: HEAD

Error Code:	HEAD
Error Description:	The current CMinpcb[] entry should be linked to a busy list but is not linked.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_ppcb
<i>Bad Data</i> =	CMsocket[CMinpcb[index].inp_socket].so_protocol.
<i>Good Data</i> =	CMinpcb[index].inp_socket.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 13 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An INPCB in-use idle stack entry ($\text{index} \leq \text{idle\_stack\_toppt}$) is corrupt. One or more of the following is true ("idle_inpcb" refers to the contents of the idle stack entry <i>inpcb[index]</i>): - idle_inpcb $\geq$ max - The INPCB indexed the contents of idle stack entry <i>inpcb[idle_inpcb]</i> and does not point back to the idle list ($\text{INPCB.inp\_qindx} \neq \text{idle\_inpcb}$).
Possible Error Effect:	INPCB may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry ($\text{index} = \text{idle\_stack\_toppt}$), the audit clears the idle stack entry (<i>inpcb[index]</i> = CMNIL_NDX) and decrements the count of idle INPCBs (<i>toppt</i> is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets <i>inpcb[index]</i> = <i>sopcb[toppt]</i>, relinks ($\text{INPCB.inp\_qindx} = \text{index}$), clears the old entry (<i>so_pcb[CMidleinpcb.toppt]</i> = CMNIL_NDX), and decrements the number of idle INPCBs (decrements <i>toppt</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Indicates INPCB idle list (CMidleinpcb).
<i>Bad Data</i> =	Value from idle stack entry (INPCB index).
<i>Good Data</i> =	value of "max".
<i>Logical Index</i> =	Index into idle stack.
<i>Dump</i> =	NA.

Table 14 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMinpcb[] idle stack is corrupt. One or more of the following is true: - The idle stack <i>toppt</i> value is out-of-range ($> \text{max}$ or $< \text{min}$). - The idle stack entry <i>inpcb[toppt]</i> = CMNIL_NDX. - The idle stack <i>toppt</i> is $< \text{max}$ and the idle stack entry <i>inpcb[toppt+1]</i> $\neq$ CMNIL_NDX.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit clears the CMinpcb[] idle stack (<i>toppt</i> = 0) and rebuilds the idle list (links all entries whose <i>inp_cbstate</i> is CMINCBFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMidleinpcb.inpcb[CMidleinpcb.toppt]
<i>Bad Data</i> =	CMidleinpcb.inpcb[CMidleinpcb.toppt+1].
<i>Good Data</i> =	CM_NUMINPCB.
<i>Logical Index</i> =	CMidleinpcb.toppt
<i>Dump</i> =	NA.

Table 15 Error Code: INDEX

Error Code:	INDEX
Error Description:	Based on the protocol in CMsocket[], the entry in CMinpcb[] should be linked to a CMtcpcb[] entry. One of the following conditions has occurred: The <i>inp_ppcb</i> is out of range or

	The <i>t_inpcb</i> in the <i>CMinpcb[]</i> entry does not point back to the <i>CMinpcb[]</i> entry.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (<i>CMidleinpcb</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	<i>CMinpcb[index].inp_socket</i> .
<i>Bad Data</i> =	<i>CMtpcb[CMinpcb[index].inp_ppcb].t_inpcb</i> .
<i>Good Data</i> =	<i>CMinpcb[index].inp_ppcb</i> .
<i>Logical Index</i> =	Index into <i>CMinpcb[]</i> .
<i>Dump</i> =	Contents of <i>CMinpcb[index]</i> .

Table 16 Error Code: INDRANGE_BAD

Error Code:	INDRANGE_BAD
Error Description:	The previous field in the current <i>CMinpcb[]</i> entry is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (<i>CMidleinpcb</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	<i>busy_list.inp_next</i> .
<i>Bad Data</i> =	<i>busy_list.inp_prev</i> .
<i>Good Data</i> =	<i>CMinpcb[index].inp_prev</i> .
<i>Logical Index</i> =	Index into <i>CMinpcb[]</i> .
<i>Dump</i> =	Contents of <i>CMinpcb[index]</i> .

Table 17 Error Code: INV

Error Code:	INV
Error Description:	The <i>CMinpcb[index].inp_ppcb</i> is null but <i>CMsocket[CMinpcb[index].inp_socket].so_protocol</i> is not a valid protocol.
Possible Error Effect:	For RTCD, RTCD messages may be lost. For VCDX, OSDS messages may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (<i>CMidleinpcb</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	<i>CMinpcb[index].inp_ppcb</i> .
<i>Bad Data</i> =	<i>CMsocket[CMinpcb[index].inp_socket].so_protocol</i> .
<i>Good Data</i> =	<i>CMinpcb[index].inp_socket</i> .
<i>Logical Index</i> =	Index into <i>CMinpcb[]</i> .
<i>Dump</i> =	Contents of <i>CMinpcb[index]</i> .

Table 18 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The state of a <i>CMinpcb[]</i> is one not recognized by the audit.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (<i>CMidleinpcb</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	<i>CMinpcb[index].inp_cbstate</i> .
<i>Good Data</i> =	NA..
<i>Logical Index</i> =	Index into <i>CMinpcb[]</i> .
<i>Dump</i> =	Contents of <i>CMinpcb[index]</i> .

Table 19 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused INPCB idle stack entry is not null (<i>index < toppt</i> and <i>inpcb[index] != CMNIL_NDX</i>).
Possible Error Effect:	INPCBs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the INPCB idle stack entry (<i>inpcb[index] = CMNIL_NDX</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Indicate INPCB idle list (<i>CMidleinpcb</i>).
<i>Bad Data =</i>	Contents of the idle stack entry (<i>inpcb[index]</i>).
<i>Good Data =</i>	CMNIL_NDX.
<i>Logical Index =</i>	Index into the idle stack.
<i>Dump =</i>	NA.

Table 20 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	One of the fields in an idle <i>CMinpcb[]</i> entry is not its default value.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (<i>CMidleinpcb</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	<i>CMidleinpcb.inpcb[CMMinpcb[index].inp_qindx]</i> .
<i>Good Data =</i>	<i>CMinpcb[index].inp_cbstate</i> .
<i>Logical Index =</i>	Index into <i>CMinpcb[]</i> .
<i>Dump =</i>	Contents of <i>CMinpcb[index]</i> .

Table 21 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle stack entry, if any, that was being audited when the stimulus occurred. • Recovers the <i>CMinpcb[]</i> entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	<i>CMidleinpcb.inpcb[CM_NUMSOPCB - AUcur_audstat.position]</i> .
<i>Good Data =</i>	<i>CM_NUMINPCB - AUcur_audstat.position</i> .
<i>Logical Index =</i>	<i>AUcur_audstat.position</i>
<i>Dump =</i>	Contents of <i>CMinpcb[AUcur_audstat.position]</i> .

INPCB-B

Software Release: 5E13 and later

1. ETHERNET INTERNET PROTOCOL CONTROL BLOCK (INPCB) AUDIT

Internet protocol is used to transmit and receive data through an ethernet network. The INPCBs are used to represent per-connection information such as foreign IP addresses and port numbers. They are either idle and are linked to the idle list or linked and are linked to a SOCKET and possibly to a transmission control protocol control block (TCPCB). The audit verifies linkages among CMinpcb[], CMtcpcb[], CMsocket[] and CMidleinpcb and verifies data within every entry in CMinpcb[].

2. DATA STRUCTURES

The following data are used by the INPCB audit:

CMinpcb[] = Internet protocol control block.
 CMtcpcb[] = TCP control block.
 CMsocket[] = Socket control block.
 CMidleinpcb = Internet protocol control block idle list.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The next field in the current CMinpcb[] entry is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	busy_list.inp_next.
<i>Bad Data =</i>	busy_list.inp_prev.
<i>Good Data =</i>	CMinpcb[index].inp_next.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 2 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The CMinpcb[] entry is in a linked list of busy entries. One of the following conditions has occurred: • CMinpcb[CMinpcb[index].inp_next].inp_prev does not equal the index into CMinpcb[] or

	CMinpcb[CMinpcb[index].inp_next].inp_head_ptr does not equal inp_head_ptr of the owner.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMinpcb[CMinpcb[index].inp_next].inp_prev.
<i>Good Data =</i>	CMinpcb[CMinpcb[index].inp_next].inp_head_ptr.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 3 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The CMinpcb[] entry is in a linked list of busy entries. One of the following conditions has occurred: - CMinpcb[CMinpcb[index].inp_prev].inp_next does not equal the index into CMinpcb[] or - CMinpcb[CMinpcb[index].inp_prev].inp_head_ptr does not equal inp_head_ptr of the owner.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMinpcb[CMinpcb[index].inp_prev].inp_head_ptr.
<i>Good Data =</i>	CMinpcb[CMinpcb[index].inp_prev].inp_next.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 4 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The audit finds the MEPIPE is OOS but an entry in CMinpcb[] is marked that it is linked when it should not be linked.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMinpcb[index].inp_socket.
<i>Bad Data =</i>	CMsocket[CMinpcb[index].inp_socket].so_pcb.
<i>Good Data =</i>	CMinpcb[index].inp_next.
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump 1 =</i>	Contents of CMinpcb[index].
<i>Dump 2 =</i>	Contents of CMscddb.state.

Table 5 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	There is no CMinpcb[] entry linked to this busy list owner but CMinpcb[].inp_head_ptr is not the null value (CM_NULL).

	The entry in CMinpcb[] can be linked to one of three busy lists. The value in <i>Good Data</i> indicates to which busy list the error refers. 0 = CMintcp, 1 = CMinudp or 2 = CMinrip.
Possible Error Effect:	Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit sets the <i>inp_head_ptr</i> field of the owner to CM_NULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	<busy>[side].inp_head_ptr.
<i>Good Data</i> =	1, 2 or 3.
<i>Logical Index</i> =	Side of busy list which has the error.
<i>Dump</i> =	Contents of CMinpcb[index].

Table 6 Error Code: BADPROTOCOL

Error Code:	BADPROTOCOL
Error Description:	For protocols other than CMIPTCP, the <i>inp_pcb</i> field should be its null value but it is not.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_ppcb.
<i>Bad Data</i> =	CMsocket[CMinpcb[index].inp_socket].so_protocol.
<i>Good Data</i> =	CMinpcb[index].inp_socket.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 7 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	An entry in CMinpcb[] should be linked to an entry in CMsocket[] if the CMscdb.state is not CMCHSTOFF. One of the following has occurred: • CMinpcb[index].inp_qindx does not equal CMNIL_NDX, • CMinpcb[index].inp_socket is out of range, • CMsocket[CMinpcb[index].inp_socket].so_pcb does not equal the index into CMinpcb[].
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_socket.
<i>Bad Data</i> =	CMinpcb[index].inp_next.
<i>Good Data</i> =	CMsocket[CMinpcb[index].inp_socket].so_pcb.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 8 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The state of an entry in CMinpcb[] is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMinpcb[index].inp_cbstate.
<i>Good Data =</i>	CMINCBMAX
<i>Logical Index =</i>	Index into CMinpcb[].
<i>Dump =</i>	Contents of CMinpcb[index].

Table 9 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	Either one or both indices in the owner of the current CMinpcb[] which have the head and tail of the linked CMinpcb[] entries is out of range. The entry in CMinpcb[] can be linked to one of three busy lists. The value in <i>Good Data</i> indicates to which busy list the error refers. 0 = CMintcp, 1 = CMinudp or 2 = CMinrip.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit repairs the linked list by linking around the bad entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	<busy>[side].inp_next.
<i>Bad Data =</i>	<busy>.inp_prev.
<i>Good Data =</i>	1, 2 or 3.
<i>Logical Index =</i>	Side of busy list which has the error.
<i>Dump =</i>	Contents of CMinpcb[<busy>].

Table 10 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The current CMinpcb[] entry is linked to a busy list. One of the following has occurred: • CMinpcb[<owner>].inp_next.inp_head_ptr does not equal address of <owner> , • CMinpcb[<owner>].inp_next.inp_prev does not equal CMNIL_NDX, • CMinpcb[<owner>].inp_prev.inp_head_ptr does not equal address of <owner> or • CMinpcb[<owner>].inp_prev.inp_next does not equal CMNIL_NDX. The entry in CMinpcb[] can be linked to one of three busy lists. The value in <i>Good Data</i> indicates to which busy list the error refers. 0 = CMintcp,

	1 = CMinudp or 2 = CMinrip.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit repairs the linked list by linking around the bad entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[<busy>[side].inp_next].inp_prev.
<i>Bad Data</i> =	CMinpcb[<busy>[side].inp_next].inp_next.
<i>Good Data</i> =	1, 2 or 3.
<i>Logical Index</i> =	Side of busy list which has the error.
<i>Dump 1</i> =	Contents of <busy>[side].
<i>Dump 2</i> =	<busy>[side].inp_next.
<i>Dump 3</i> =	<busy>[side].inp_prev.

Table 11 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The CMinpcb[] is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	CMinpcb[index].inp_cbstate.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 12 Error Code: HEAD

Error Code:	HEAD
Error Description:	The current CMinpcb[] entry should be linked to a busy list but is not linked.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_ppcb
<i>Bad Data</i> =	CMsocket[CMinpcb[index].inp_socket].so_protocol.
<i>Good Data</i> =	CMinpcb[index].inp_socket.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 13 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A INPCB in-use idle stack entry (index ≤ CMidleinpcb.toppt) is corrupt. One or more of the following is true: • CMidleinpcb.inpcb[index] ≥ CM_NUMINPCB • CMidleinpcb.inpcb[index] ≥ 0 • CMinpcb[CMidleinpcb.inpcb[index]].inp_qindx != index into CMidleinpcb.inpcb[].

Possible Error Effect:	INPCB may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry (index = CMidleinpcb.toppt), the audit clears the idle stack entry (CMidleinpcb[index] = CMNIL_NDX) and decrements the count of idle INPCB (CMidleinpcb.toppt is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets CMidleinpcb.inpcb[index] = CMidleinpcb.inpcb[CMidleinpcb.toppt], relinks the INPCB (CMinpcb[CMidleinpcb.inpcb[CMidleinpcb.toppt]].inp_qindx = index), clears the old entry (CMidleinpcb.inpcb[CMidleinpcb.toppt] = CMNIL_NDX) and decrements the number of idle TCPBs (decrements CMidleinpcb.toppt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUINPTYPE.
<i>Bad Data</i> =	Index of next available idle CMinpcb[].
<i>Good Data</i> =	CM_NUMINPCB.
<i>Logical Index</i> =	Index into CMidleinpcb.tcpb[].
<i>Dump</i> =	NA.

Table 14 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMinpcb[] idle stack is corrupt. One or more of the following is true: - The idle stack CMidleinpcb.toppt value is out-of-range (> max or < min). - The idle stack entry CMidleinpcb.inpcb[CMidleinpcb.toppt] = CMNIL_NDX. - The idle stack CMidleinpcb.toppt is < max and the idle stack entry CMidleinpcb.inpcb[CMidleinpcb.toppt+1] != CMNIL_NDX.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit sets CMidleinpcb.toppt = 0 and rebuilds the idle list (links all entries whose CMinpcb[].inp_cbstate is CMINCBFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMidleinpcb.inpcb[CMidleinpcb.toppt]
<i>Bad Data</i> =	CMidleinpcb.inpcb[CMidleinpcb.toppt+1].
<i>Good Data</i> =	CM_NUMINPCB.
<i>Logical Index</i> =	CMidleinpcb.toppt
<i>Dump</i> =	NA.

Table 15 Error Code: INDEX

Error Code:	INDEX
Error Description:	Based on the protocol in CMsocket[], the entry in CMinpcb[index] should be linked to a CMtcpb[CMinpcb[index].inp_ppcb] entry. One of the following conditions has occurred: - The CMinpcb[].inp_ppcb is out of range or - The CMtcpb[CMinpcb[index].inp_ppcb].t_inpcb does not point back to the CMinpcb[index] entry.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_socket.
<i>Bad Data</i> =	CMtpcb[CMinpcb[index].inp_ppcb].t_inpcb.
<i>Good Data</i> =	CMinpcb[index].inp_ppcb.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 16 Error Code: INDRANGE_BAD

Error Code:	INDRANGE_BAD
Error Description:	The previous field in the current CMinpcb[] entry is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	busy_list.inp_next.
<i>Bad Data</i> =	busy_list.inp_prev.
<i>Good Data</i> =	CMinpcb[index].inp_prev.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 17 Error Code: INV

Error Code:	INV
Error Description:	The CMinpcb[index].inp_ppcb is null but CMsocket[CMinpcb[index].inp_socket].so_protocol is not a valid protocol.
Possible Error Effect:	For RTCD, RTCD messages may be lost. For VCDX, OSDS messages may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMinpcb[index].inp_ppcb.
<i>Bad Data</i> =	CMsocket[CMinpcb[index].inp_socket].so_protocol.
<i>Good Data</i> =	CMinpcb[index].inp_socket.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 18 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The state of a CMinpcb[] is one not recognized by the audit.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMinpcb[index].inp_cbstate.
<i>Good Data</i> =	NA..
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 19 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused INPCB idle stack entry (index > CMidleinpcb.toppt) is not null (CMidleinpcb.inpcb[index] != CMNIL_NDX and index < CMidleinpcb.toppt)
Possible Error Effect:	INPCBs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the INPCB idle stack entry (CMidleinpcb.inpcb[index] = CMNIL_NDX).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	AUINPTYPE.
<i>Bad Data</i> =	Contents of the idle stack entry (CMidleinpcb.inpcb[index]).
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into the idle stack.
<i>Dump</i> =	NA.

Table 20 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	One of the fields in an idle CMinpcb[] entry is not its default value.
Possible Error Effect:	For VCDX, ethernet link to workstation is restarted. Calls needing workstation may fail. For RTCD, MELNK is restarted. A RTCD record may be lost.
Corrective Action Taken:	The audit has all fields set to default values, unlinks it from any list on which it may be and puts the entry on the idle stack (CMidleinpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUINPTYPE.
<i>Bad Data</i> =	CMidleinpcb.inpcb[CMinpcb[index].inp_qindx].
<i>Good Data</i> =	CMinpcb[index].inp_cbstate.
<i>Logical Index</i> =	Index into CMinpcb[].
<i>Dump</i> =	Contents of CMinpcb[index].

Table 21 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle list entry, if any, that was being audited when the stimulus occurred. • Recovers the CMinpcb[] entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMidleinpcb.inpcb[CM_NUMINPCB - AUcur_audstat.position].
<i>Good Data</i> =	CM_NUMINPCB - AUcur_audstat.position.
<i>Logical Index</i> =	AUcur_audstat.position
<i>Dump</i> =	Contents of CMinpcb[AUcur_audstat.position].

INSPDATA

Software Release: 5E11 and later

1. INSPECT DATA (INSPDATA) AUDIT

ISDN inspect, a network-provided service, allows National ISDN phase 3 users to be presented (using displays) with information about feature activators, idle call appearances, and non-idle call appearances. The relation RLINSPDATA holds the daily ISDN inspect delivered counts per lport.

2. DATA STRUCTURES

The RLINSPDATA dynamic relation is the dynamic representation of the number of Inspect usage-sensitive terminals or lports. Its key (lport) is obtained from the static relation RLIS_TERM and the only other attribute is the count of the number of ISDN inspect usages for the lport.

Read the static RLIS_TERM tuple with the same key. If RLIS_TERM.ama_insp is DBYES, return AUEXISTS; else return AUEXTRA so this RLINSPDATA tuple is deleted.

3. AUDIT CHECKS

The INSPDATA audit uses the DRINDEX audit driver. There are no error checks specific to this audit.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

IPCSPCI

Software Release: 5E14 and later

1. ISUP POINT CODES (IPCSPCI) AUDIT

This audit steps through the RLIPCS_PCI (ISUP point code status relation) and checks the data in it against the RLPCI_RTE (point code route relation).

2. DATA STRUCTURES

For details on the fields contained in the RLIPCS_PCI and the RLPCI_RTE relations refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit attempts to read a rIPCI_RTE tuple but fails.
Possible Error Effect:	Integrated services digital network user part (ISUP) calls may be affected.
Corrective Action Taken:	The related audit PCIRTE is scheduled and attempts to insert the missing rIPCI_RTE tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIIPCS_PCI tuple.
<i>Bad Data =</i>	Dynamic point code index (rIPCI_RTE.pci).
<i>Good Data =</i>	Static point code index (rIIPCS_PCI.pci).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	Contents of rIIPCS_PCI tuple.

Table 2 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The audit detects either rIPCI_RTE.isup_opc does not match rIIPCS_PCI.isup_opc or rIPCI_RTE.isup_dpc does not match rIIPCS_PCI.isup_dpc.
Possible Error Effect:	ISUP calls may be affected.
Corrective Action Taken:	The audit sets rIPCI_RTE.isup_opc equal to rIIPCS_PCI.isup_opc and sets rIPCI_RTE.isup_dpc equal to rIIPCS_PCI.isup_dpc and notifies application code about the correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIPCI_RTE.isup_opc.
<i>Bad Data =</i>	rIIPCS_PCI.isup_opc.
<i>Good Data =</i>	rIPCI_RTE.isup_dpc.
<i>Logical Index =</i>	rIIPCS_PCI.isup_dpc.
<i>Dump 1 =</i>	Contents of rIIPCS_PCI tuple.
<i>Dump 2 =</i>	Contents of rIPCI_RTE tuple.

ISANBUS

Software Release: 5E10 and later

1. INTEGRATED SERVICES ACCESS NETWORK BUS (ISANBUS) AUDIT

The integrated service line unit (ISLU) Z-interface (ISLU-Z) converts digital signals to analog. The ISLU-Z uses ringing generators (RGs) and metallic access network (MAN) circuits. Several audits check the data structures related to the ISLU-Z circuits: the ringing generator occupancy table audit (RGOCCUP), the ringing generator task data block audit (RGTDDB), the ringing generator bus audit (RGBUS), the channel control block user audit (CCBUSR), the ISLU-Z access network bus audit (ISANBUS), and the spare line card audit (SPRCRD). The ISANBUS audit checks the peripheral control arrays used for allocating the use of the ISLU metallic access network.

The ISANBUS audit is part of the peripheral control audits subgroup. It is part of the Integrated services line unit (ISLU) application loadable package (ISLUAPP).

Refer to the "Audit Checks" section below for more information about the use and purpose of the ISANBUS audit.

2. DATA STRUCTURES

The ISANBUS audit uses the following data structures. The peripheral control line group bus array (PClgbus) allocates and deallocates time-shared users for the ISLU's metallic access network line group bus. The peripheral control test bus array (PCtstbus) allocates time-shared users for the ISLU's metallic access network test bus. The peripheral control bus maps array (PCbusmaps) contains occupancy bitmaps for the ISLU line group and test group buses. The peripheral control bus idle array (PCbusidle) consists of 256 bits that correspond inversely to the bits in PCbusmaps. The audit also checks states in the switch maintenance equipment status relation (RLSMEST), the circuit data block common area relation (RLCDBCOM), the circuit queue block relation (RLCQB), and the channel control block common area relation (RLCCBCOM).

3. AUDIT CHECKS

The ISANBUS audit verifies the manner and mode of its input data, then calls its semantic check function to make the following checks. For each line group and test bus on an integrated services line unit (ISLU):

For each line group bus:

LOSTCDB = Read rICDBCOM for bus.

READSMEST = Read rISMEST for bus.

If circuit is active:

NIDLE = Check maintenance bit in PC for one.

NOTIDLE = Check maintenance bit in PC for zero.

If circuit is out-of-service:

CDBCOM = Read rICDBCOM for bus.

IDLE_LINK = Check rICDBCOM owner.

NOCCB = Read rICCB COM for bus.

OWNER = Check rICDBCOM owner.

STATE = Check maintenance bit.

If queue bit is not set:

CQB_LINK = Queue bit not set, but linked to an rICQB.

If queue bit is set:

CQBCDB = Queue bit set, but not linked to an rICQB.

If user bit is set:

ASSOC_CDB = Can't access the rICBCOM.

LNK_FE = Compare bus array with PCbusmaps array.

For each test bus:

OWNLINK = Read rICDBCOM for bus.

SMEST = Read rSMEST for bus.

If circuit is active:

NOTBUSY = Check maintenance bit in PC for zero.

STATE_IDL = Check maintenance bit in PC for one.

If circuit is out-of-service:

CDBITS = Read rICBCOM for bus.

COUNT_CDB = Read rICDBCOM for bus.

MAINT_STATE = Check maintenance bit.

STATE_VAR = Check rICDBCOM owner.

WRONGLIST = Check rICDBCOM owner.

If the queue bit is not set:

CQBHEADC = Queue bit not set, but linked to an rICQB.

If the queue bit is set:

CQBCCB = Queue bit set, but not linked to an rICQB.

If the user bit is set:

DSCC_CDB	= Can't access the rCCBCOM.
IDL_STATE	= PCbusmaps indicates busy; PCbusidle indicates idle.
INV	= Verify that the first bit in PCbusidle is one.
STATE_LNK	= Compare bus array with PCbusmaps array.
STATE_SET	= PCbusmaps indicates idle; PCbusidle indicates busy.

For each entry in PCbusmaps[]:

BUSY3MAP	= The PCbusmaps entry is not idle, but no entries exist in its bitmap.
BUSYIDLE	= The PCbusmaps entry is not idle, but no entries exist in its ccbusr.
GRBG	= Can't retrieve rCDBCOM whose key was found in PCbusmaps.
INV_STATE	= A busy entry in PCbusmaps[].ccbusr exists, but the hardware path is free to the circuit.
IS_NOTUSED	= The PCbusmaps entry is idle, but a non-idle entry exists in its bitmap.
LPSTATE	= rCCBCOM lpstate is invalid for this PCbusmaps entry.
NOACCBK	= Can't retrieve rCDBCOM whose key was found in PCbusmaps.
TSIDLE_BUSY	= The PCbusmaps entry is idle, but a non-idle entry exists in its ccbusr.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	The audit cannot access the rCCBCOM associated with a PCbusmaps entry.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Set the PCbusmaps ccbusr field to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PCbusmaps ccbusr field.
<i>Good Data</i> =	Return code from trying to access rCCBCOM tuple.
<i>Logical Index</i> =	LGBUS tuple usr field.
<i>Dump</i> =	NA.

Table 2 Error Code: BUSY3MAP

Error Code:	BUSY3MAP
Error Description:	A PCbusmaps[] entry is not idle (PCbusmaps[].backlink is not equal to 0xffff) but there are no entries in its associated bitmap array (PCbusmaps[].bitmap[]).
Possible Error Effect:	Resources used for ISLU-Z ringing may be lost.
Corrective Action Taken:	All resources for this entry are cleared. The PCbusidle bit is cleared, PCbusmaps[].backlink is cleared, PCbusmaps[].ccbusr[] and PCbusmaps[].bitmap[] arrays are cleared.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of entry in PCbusmaps.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Array index of PCbusmaps.
<i>Dump</i> =	PCbusmaps entry.

Table 3 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	A PCbusmaps[] entry is not idle (PCbusmaps[].backlink is not equal to 0xffff) but there are no entries in its associated ccbusr array (PCbusmaps[].ccbusr[]).
Possible Error Effect:	Resources used for ISLU-Z ringing may be lost.
Corrective Action Taken:	All resources for this entry are cleared. The PCbusidle bit is cleared, PCbusmaps[].backlink is cleared, PCbusmaps[].ccbusr[] and PCbusmaps[].bitmap[] arrays are cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in PCbusmaps.
<i>Bad Data</i> =	PCNULLCCB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Array index of PCbusmaps.
<i>Dump</i> =	PCbusmaps entry.

Table 4 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The audit cannot access an associated rICBCOM tuple.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rICBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Return code from trying to access the rICBCOM tuple.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Associated rICBCOM tuple.
<i>Dump 2</i> =	Associated rICBCOM tuple.

Table 5 Error Code: CDBCOM

Error Code:	CDBITS
Error Description:	Cannot access the rICBCOM tuple associated with this TSTBUS.
Possible Error Effect:	A TSTBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rICBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Return code from trying to access rICBCOM.
<i>Logical Index</i> =	rICBCOM tuple name field, circuit name of TSTBUS being checked.
<i>Dump 1</i> =	Associated rICBCOM tuple.
<i>Dump 2</i> =	rSMEST tuple for the TSTBUS being checked.

Table 6 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	The audit cannot access an associated rICBCOM tuple.
Possible Error Effect:	A TSTBUS may not be usable.
Corrective Action Taken:	Schedule the CKTDATA and CCBCOM audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Return code from trying to access the rICBCOM tuple.
<i>Logical Index</i> =	NA.

<i>Dump 1</i> =	Associated rICDBCOM tuple.
<i>Dump 2</i> =	Wrong CCBCOM accessed.

Table 7 Error Code: CQBCCB

Error Code:	CQBCCB
Error Description:	The TSTBUS queue field indicates queuing but the associated rICDBCOM indicates no queuing.
Possible Error Effect:	An TSTBUS may not be usable.
Corrective Action Taken:	Set the TSTBUS queue bit to indicate no queuing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	TSTBUS queue bit. (Not actually bad data.)
<i>Good Data</i> =	rICDBCOM queuing data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Associated rICDBCOM tuple.

Table 8 Error Code: CQB CDB

Error Code:	CQB CDB
Error Description:	The LGBUS queue field indicates queuing but the associated rICDBCOM indicates no queuing.
Possible Error Effect:	A LGBUS may not be usable.
Corrective Action Taken:	Set the LGBUS queue bit to indicate no queuing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LGBUS queue bit. (Not actually bad data.)
<i>Good Data</i> =	rICDBCOM queuing data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Associated rICDBCOM tuple.

Table 9 Error Code: CQBHEADC

Error Code:	CQBHEADC
Error Description:	The TSTBUS queue field indicates no queuing but the associated rICDBCOM indicates other than no queuing.
Possible Error Effect:	An TSTBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rICDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	TSTBUS queue bit. (Not actually bad data.)
<i>Good Data</i> =	rICDBCOM queuing data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Associated rICDBCOM tuple.

Table 10 Error Code: CQB_LINK

Error Code:	CQB_LINK
Error Description:	The LGBUS queue field indicates no queuing but the associated rICDBCOM indicates other than no queuing.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rICDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LGBUS queue bit. (Not actually bad data.)
<i>Good Data</i> =	rICDBCOM queuing data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Associated rICDBCOM tuple.

Table 11 Error Code: CQB_LINK

Error Code:	CQB_LINK

Error Description:	The LGBUS queue field indicates queuing but the associated rICDBCOM indicates no queuing.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Set the LGBUS queue bit to indicate no queuing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LGBUS queue bit. (Not actually bad data.)
<i>Good Data</i> =	rICDBCOM queuing data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Associated rICDBCOM tuple.

Table 12 Error Code: DSCC_CDB

Error Code:	DSCC_CDB
Error Description:	The audit cannot access the rICCB COM associated with a busmaps entry.
Possible Error Effect:	A TSTBUS may not be usable.
Corrective Action Taken:	Set the busmaps ccbusr field to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Busmaps ccbusr field.
<i>Good Data</i> =	Return code from trying to access rICCB COM tuple.
<i>Logical Index</i> =	TSTBUS tuple usr field.
<i>Dump</i> =	NA.

Table 13 Error Code: GRBG

Error Code:	GRBG
Error Description:	An RLCCB COM key exists in the ccbusr array of PCbusmaps[] (PCbusmaps[].ccbusr[]), but no valid rICCB COM exists with that key.
Possible Error Effect:	Resources used for ISLU-Z ringing may be lost.
Corrective Action Taken:	All resources for this entry are cleared. The PCbusidle bit is cleared, PCbusmaps[].backlink is cleared, PCbusmaps[].ccbusr[] and PCbusmaps[].bitmap[] arrays are cleared. Associated PCtstbus and PClgbus entries are also cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in PCbusmaps.
<i>Bad Data</i> =	Invalid entry in PCbusmaps[].ccbusr[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Array index of PCbusmaps.
<i>Dump</i> =	PCbusmaps entry.

Table 14 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The LGBUS tuple reports out-of-service in rISMEST and is linked to an idle list.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Report the error and schedule the CKTDATA audit to link back to correct list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Idle list linked to.
<i>Good Data</i> =	LGBUS rISMEST base state.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	rISMEST tuple for the LGBUS being checked.
<i>Dump 2</i> =	Associated rICDB COM tuple.

Table 15 Error Code: IDL_STATE

Error Code:	IDL_STATE
Error Description:	The audit has detected an inconsistency between a bit in busidle and the corresponding entry in busmaps.
Possible Error Effect:	The tuple in busmaps may try to be accessed incorrectly.

Corrective Action Taken:	Set the bit in busidle to 1, indicating busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value of bit in busidle.
<i>Good Data =</i>	Value of backlink field in busmaps.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 16 Error Code: INV

Error Code:	INV
Error Description:	The first bit in the PCbusidle array is not set to 1.
Possible Error Effect:	The first tuple in busmaps may try to be accessed incorrectly.
Corrective Action Taken:	Set the bit to 1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value of first bit in PCbusidle.
<i>Good Data =</i>	Value the first bit in PCbusidle should be.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 17 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	Less than success has been returned from PC check timeslot path function.
Possible Error Effect:	A loss of RG resources could develop.
Corrective Action Taken:	Quarantine the rCCBCOM tuple associated to the entry in busmaps.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Element in busmaps array being referenced. (Not actually bad data.)
<i>Good Data =</i>	Second index into busmaps array.
<i>Logical Index =</i>	Service group.
<i>Dump =</i>	rCCBCOM tuple to be quarantined.

Table 18 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An entry in PCbusmaps has no backlink to CCBCOM but one or more of its bitmap entries are non-zero.
Possible Error Effect:	Loss of ISLU-Z ringing resources.
Corrective Action Taken:	All bitmap entries for this PCbusmap are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCbusmaps entry.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Zero.
<i>Logical Index =</i>	Index into PCbusmaps.
<i>Dump =</i>	PCbusmaps entry.

Table 19 Error Code: LNK_FE

Error Code:	LNK_FE
Error Description:	The audit has detected an inconsistency between the LGBUS tuple being checked and the associated backlink field in the associated busmap array.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Set the LGBUS usr field to zero. Set the corresponding busmaps backlink field to 0xffff. Idle the corresponding entry in the PCbusidle array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of busmaps backlink field (Not an address).
<i>Bad Data =</i>	Number of the line unit the LGBUS belongs to. (Not actually bad data.)
<i>Good Data =</i>	Number of the LGBUS.
<i>Logical Index =</i>	Service group of LGBUS.

<i>Dump</i> =	NA.

Table 20 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	Cannot access the rCCBCOM associated with this LGBUS.
Possible Error Effect:	An LGBUS may not be able to used.
Corrective Action Taken:	Related CKTDATA audit should make the cdbcom tuple available
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of cdbcom tuple, provided via DA primitive.
<i>Bad Data</i> =	Name field of cdbcom returned from DA.
<i>Good Data</i> =	unit_id field of related rSMEST tuple.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	rSMEST tuple for the LGBUS being checked.
<i>Dump 2</i> =	Associated rCDBCOM tuple.

Table 21 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The lpstate of a rCCBCOM tuple associated with an entry in busmaps is incorrect.
Possible Error Effect:	A loss of RG resources could develop.
Corrective Action Taken:	Set the busmaps ccbusr field to null, clear the appropriate PCbusidle bit, reset the busmaps backlink field to 0xffff, and zero out the busmaps bitmap fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	rCCBCOM lpstate.
<i>Good Data</i> =	PCRING.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rCCBCOM tuple.

Table 22 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The TSTBUS is out of service in rSMEST and the TSTBUS maintenance bit is not set.
Possible Error Effect:	An TSTBUS may be used when it should not be.
Corrective Action Taken:	Set the maintenance bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	TSTBUS maintenance bit.
<i>Good Data</i> =	rSMEST TSTBUS tuple basestate.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rSMEST tuple for the TSTBUS being checked.

Table 23 Error Code: NIDLE

Error Code:	NIDLE
Error Description:	The LGBUS tuple is active in rSMEST, the LGBUS maintenance bit is set to indicate maintenance activity and yet the rCDBCOM tuple associated with the LGBUS is not linked to a valid maintenance process.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rCDBCOM tuple and zero the LGBUS maintenance bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LGBUS maintenance bit.
<i>Good Data</i> =	What the rCDBCOM is linked to.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Associated rCDBCOM tuple.
<i>Dump 2</i> =	rSMEST tuple for the LGBUS being checked.

Table 24 Error Code: NOACCBK

Error Code:	NOACCBK
Error Description:	While auditing PCbusmap[] entries, was not able to access the related rCCBCOM tuple.
Possible Error Effect:	A PCbusmaps[] entry may be improperly linked.
Corrective Action Taken:	Set the PCbusmaps entry to an idle state and idle the usr field in the related PCtstbus[] or PClgbus[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCBCOM tuple returned from DA primitive.
<i>Bad Data =</i>	ccbnum field of CCBCOM tuple returned from DA primitive.
<i>Good Data =</i>	ccbusr field of PCbusmaps[] entry being checked.
<i>Logical Index =</i>	PCbusmaps entry being checked.
<i>Dump =</i>	CCBCOM tuple retrieved by DA primitive.

Table 25 Error Code: NOCCB

Error Code:	NOCCB
Error Description:	Cannot access the CCBCOM tuple for a PClgbus[][] entry that is OOS in SMEST and not on an OOS list. The PClgbus[][] entry should be owned by a maintenance process with an associated CCBCOM tuple.
Possible Error Effect:	A PClgbus[][] entry may be improperly linked.
Corrective Action Taken:	Quarantine the related CDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCBCOM tuple returned from DA function.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	Return from DA function.
<i>Logical Index =</i>	Name field of related CDBCOM tuple.
<i>Dump 1 =</i>	Related CDBCOM tuple.
<i>Dump 2 =</i>	Related SMEST tuple.

Table 26 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The PCtstbus tuple is active in rSMEST, the PCtstbus maintenance bit indicates no maintenance, and yet the rCDBCOM associated with this PCtstbus is not on the idle list.
Possible Error Effect:	An PCtstbus may not be usable.
Corrective Action Taken:	Quarantine the associated rCDBCOM tuple and schedule the rCDBCOM audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Place where the associated TSTBUS is linked to.
<i>Good Data =</i>	Value of the idle list the PCtstbus should be linked to.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	rSMEST tuple for the PCtstbus being checked.
<i>Dump 2 =</i>	Associated rCDBCOM tuple.

Table 27 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The LGBUS tuple is active in rSMEST, the LGBUS maintenance bit indicates no maintenance, and yet the rCDBCOM tuple associated with this LGBUS is not on the idle list.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rCDBCOM tuple and schedule the CDBCOM audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Place where the associated LGBUS is linked to.
<i>Good Data =</i>	Value of the idle list the LGBUS should be linked to.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	rSMEST tuple for the LGBUS being checked.

<i>Dump 2 =</i>	Associated rICBCOM tuple.

Table 28 Error Code: OWNER

Error Code:	OWNER
Error Description:	The owner of the LGBUS is not a maintenance process when it should be.
Possible Error Effect:	An LGBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rICBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rSMEST basestate for the LGBUS being checked.
<i>Bad Data =</i>	Associated rICBCOM lpstate.
<i>Good Data =</i>	What the lpstate should be.
<i>Logical Index =</i>	rICBCOM→cdccm.st ownk.
<i>Dump 1 =</i>	Associated rICBCOM tuple.
<i>Dump 2 =</i>	Associated rICBCOM tuple.
<i>Dump 3 =</i>	rSMEST tuple for the LGBUS being checked.

Table 29 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Cannot access the rICBCOM associated with this PCtstbus[][].
Possible Error Effect:	A TSTBUS may not be usable.
Corrective Action Taken:	Related CKTDATA audit should make the rICBCOM tuple available.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CDBCOM tuple, provided via DA primitive.
<i>Bad Data =</i>	Name field of CDBCOM returned from DA.
<i>Good Data =</i>	unit_id field of related rSMEST tuple.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	rSMEST tuple for the LGBUS being checked.
<i>Dump 2 =</i>	Associated rICBCOM tuple.

Table 30 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The audit cannot read rSMEST data associated with the PClgbus being checked.
Possible Error Effect:	A PClgbus may not be usable.
Corrective Action Taken:	Report the error and schedule the CKTDATA and SMEST audits to determine if an rSMEST entry should exist for the PClgbus being checked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Line unit that the PClgbus belongs. (Not actually bad data.)
<i>Good Data =</i>	PClgbus number.
<i>Logical Index =</i>	Circuit name for the PClgbus being checked.
<i>Dump =</i>	NA.

Table 31 Error Code: SMEST

Error Code:	SMEST
Error Description:	The audit cannot read rSMEST data associated with the PCtstbus being checked.
Possible Error Effect:	A PCtstbus may not be usable.
Corrective Action Taken:	Report the error and schedule the CKTDATA and SMEST audits to determine if a rSMEST entry should exist for the PCtstbus being checked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from trying to access rSMEST data for the PCtstbus being checked.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name for the PCtstbus being checked.
<i>Dump =</i>	NA.

Table 32 Error Code: STATE

Error Code:	STATE
Error Description:	The PClgbus is reported to be out-of-service in rSMEST and the PClgbus maintenance bit is not set.
Possible Error Effect:	An PClgbus may be used when it should not be.
Corrective Action Taken:	Set the maintenance bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PClgbus maintenance bit.
<i>Good Data =</i>	rSMEST PClgbus tuple basestate.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	rSMEST tuple for the PClgbus being checked.

Table 33 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The PCtstbus tuple is active in rSMEST, the PCtstbus maintenance bit is set to indicate maintenance activity, and yet the rCDBCOM tuple associated with the PCtstbus is not linked to a valid maintenance process.
Possible Error Effect:	An PCtstbus may not be usable.
Corrective Action Taken:	Quarantine the associated rCDBCOM tuple and zero the PCtstbus maintenance bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	PCtstbus maintenance bit.
<i>Good Data =</i>	What the rCDBCOM tuple is linked to.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Associated rCDBCOM tuple.
<i>Dump 2 =</i>	rSMEST tuple for the PCtstbus being checked.

Table 34 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	The audit has detected an inconsistency between the TSTBUS tuple being checked and the associated backlink field in the associated busmap array.
Possible Error Effect:	A TSTBUS may not be usable.
Corrective Action Taken:	Set the TSTBUS usr field to zero. Set the corresponding busmaps backlink field to 0xffff. Idle the corresponding entry in the PCbusidle array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of busmaps backlink field (Not an address).
<i>Bad Data =</i>	Number of the line unit the TSTBUS belongs to. (Not actually bad data.)
<i>Good Data =</i>	Number of the TSTBUS.
<i>Logical Index =</i>	Service group of TSTBUS.
<i>Dump =</i>	NA.

Table 35 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The audit has detected an inconsistency between a bit in PCbusidle and the corresponding entry in busmaps.
Possible Error Effect:	The tuple in busmaps may not try to be accessed.
Corrective Action Taken:	Set the bit in PCbusidle to 0, indicating idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCbusmaps[] entry with the error.
<i>Bad Data =</i>	Value of bit in PCbusidle.
<i>Good Data =</i>	Value of backlink field in busmaps.
<i>Logical Index =</i>	Entry in PCbusmaps[] being checked.
<i>Dump 1 =</i>	Related PCbusidle[] entry.
<i>Dump 2 =</i>	PCbusmaps[] entry.

Table 36 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The owner of the TSTBUS is not a maintenance process when it should be.
Possible Error Effect:	A TSTBUS may not be usable.
Corrective Action Taken:	Quarantine the associated rICBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rISMEST basestate for the TSTBUS being checked. (Not an address.)
<i>Bad Data =</i>	Associated rICBCOM LPSTATE.
<i>Good Data =</i>	What the LPSTATE should be.
<i>Logical Index =</i>	rICDBCOM→cdccm.st_ownk.
<i>Dump 1 =</i>	Associated rICDBCOM tuple.
<i>Dump 2 =</i>	Associated rICBCOM tuple.
<i>Dump 3 =</i>	rISMEST tuple for the TSTBUS being checked.

Table 37 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY
Error Description:	A PCbusmaps[] entry is idle (PCbusmaps[].backlink is equal to 0xffff) but an element in its ccbusr array is not idle (PCbusmaps[].ccbusr[]).
Possible Error Effect:	Resources used for ISLU-Z ringing may be lost.
Corrective Action Taken:	The ccbusr array is cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of entry in PCbusmaps.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Array index of PCbusmaps.
<i>Dump =</i>	PCbusmaps entry.

Table 38 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The TSTBUS tuple is out of service in rISMEST and linked to an idle list.
Possible Error Effect:	An PCtstbus may not be usable.
Corrective Action Taken:	Report the error and schedule the CKTDATA audit to link back to correct list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Idle list linked to.
<i>Good Data =</i>	PCtstbus rISMEST base state.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	rISMEST tuple for the PCtstbus being checked.
<i>Dump 2 =</i>	Associated rICDBCOM tuple.

Table 39 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit has been invoked after taking a system reset.
Possible Error Effect:	Causes asserts to fire and/or system resets.
Corrective Action Taken:	The audit has been returned to audit control and will be rescheduled for recovery activities.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

ISATUCD

Software Release: 5E10 and later

1. ATTENDANT TERMINAL UNIFORM CALL DISTRIBUTION (ISATUCD)

The purpose of this audit is to perform dynamic-to-static existence checks between rISATUCD and rLICX_ICI tuples. Any surplus rISATUCD tuples are deleted. This audit also performs the key type check for these dynamic tuples.

The dynamic relation RLISATUCD is an indexed relation, keyed by the integrated services digital network (ISDN) attendant terminal index. The relation RLISATUCD contains the ISDN attendant uniform call distribution (UCD) hunt variables and out-of-service (OOS) status for the ISAT group. This relation must be present for any group which is an ISAT hunt type.

2. DATA STRUCTURES

The RLISATUCD relation contains the ISDN attendant UCD hunt variable and OOS bit status.

The ISATUCD audit is part of the AURtaque module. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The ISATUCD audit performs existence checks between the static relation RLICX_ICI accessed by information from RLRT_MHG, and the dynamic relation RLISATUCD.

If no tuple is found in RLICX_ICI and RLISATUCD indicates that it exists and if recent change is not active on RLICX_ICI, then an attempt is made to delete the extra tuple and an error message is printed.

4. ERROR HANDLING

When the audit encounters an error, an error message is printed on the read-only printer (ROP) and an attempt is made to recover from an error.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rISATUCD tuple found for an ISDN attendant multi-line hunt group.
Possible Error Effect:	Unable to preempt the lines in this group and possible ASSERT failures.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rISATUCD tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rISATUCD tuple.
<i>Dump =</i>	Dump of the rISATUCD tuple.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	An rISATUCD tuple has bad data in the key (isat_idx) field for an ISDN attendant multi-line hunt group.
Possible Error Effect:	Unable to preempt the lines in this group and possible ASSERT failures.

Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rllSATUCD tuple.
<i>Bad Data</i> =	Contents of the isat_idx field in the rllSATUCD tuple.
<i>Good Data</i> =	Correct of the rllSATUCD tuple.
<i>Logical Index</i> =	Key of the rllSATUCD tuple.
<i>Dump</i> =	Dump of the rllSATUCD tuple.

ISDCCFG

Software Release: 5E10 and later

1. ISLU COMMON DATA PATH CONFIGURATION (ISDCCFG) AUDIT

The integrated services line unit (ISLU) common data path configuration table is dynamic data that contains the D-channel type, the common data configuration, and the circuit names of the primary and secondary common data for each of the eight possible peripheral units. This global dynamic array is associated with the ISLU and remote ISLU (RISLU) peripheral units.

The ISDCCFG audit is part of the peripheral control audits subgroup.

2. DATA STRUCTURES

The main data structures used by the ISDCCFG audit are the ISLU common data path configuration array (PCiscdconfig[]), the ISLU peripheral interface data bus relation (RLISLUPIDB), a static relation, the switch maintenance equipment status relation (RLSMEST), a dynamic relation, and the peripheral control line unit service group name (PClusgname[] array. These data structures are shown in Figure 1 .

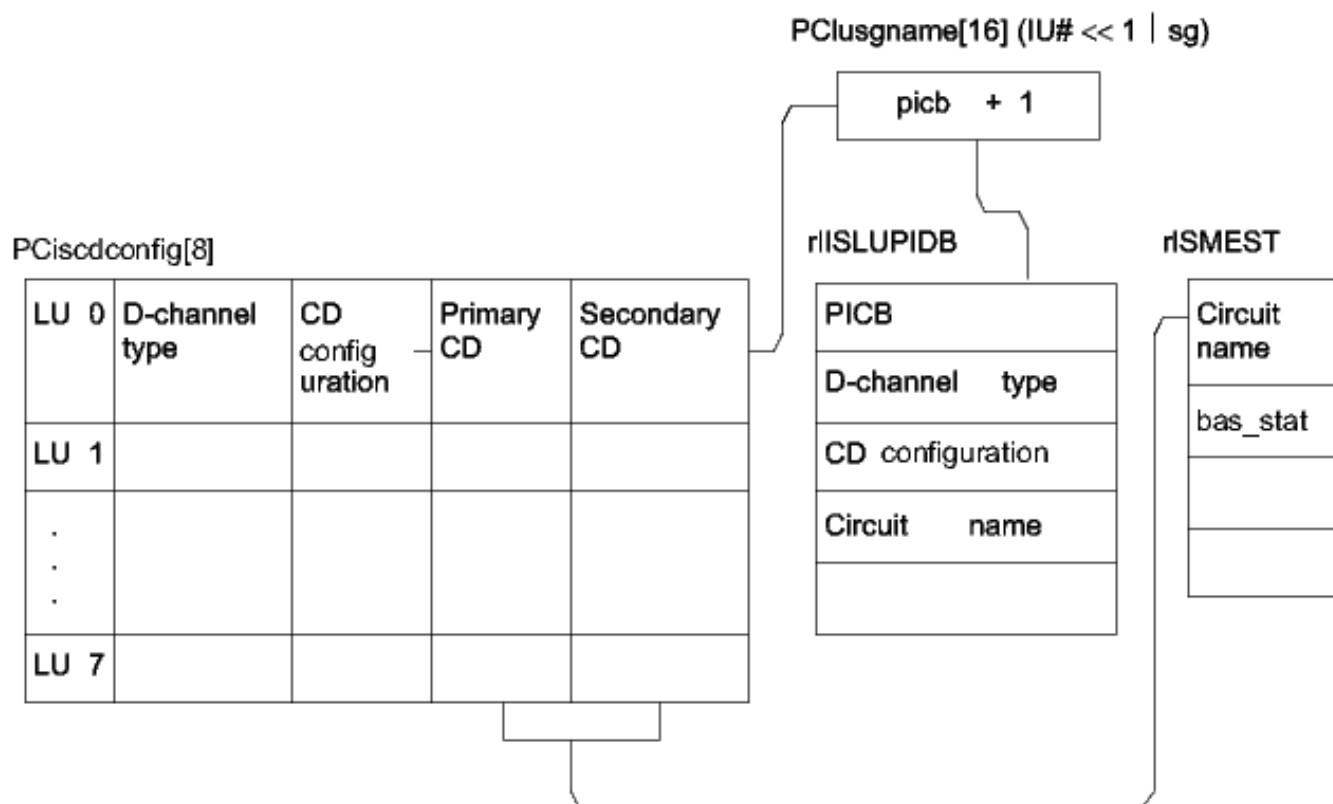


Figure 1 (ISDCCFG-1) Data Structures

3. AUDIT CHECKS

The ISDCCFG audit steps through PCiscdconfig[] for all peripheral units and performs the following checks:

- Ensures that the D-channel type and common data configuration are the same as the values in the static RLISLUPIDB relation.

- (b) Checks the circuit names of primary and secondary common data against rISMEST. The key of rISMEST (cktname) is formed as follows:



If the common data configuration field is in the active/standby mode, the circuit name in the primary common data should match an rISMEST tuple with a bas_stat of active. The circuit name in the secondary common data should match the rISMEST tuple with a bas_stat of standby. Statuses other than active or standby in rISMEST should have null in the primary or secondary common data fields.

If the common data configuration field is in the active/active mode, the circuit name in the primary common data should match an rISMEST tuple with a bas_stat of active major. The circuit name in the secondary common data should match the rISMEST tuple with a bas_stat of active minor. Statuses other than active major or active minor in rISMEST should have null in the primary or secondary common data fields.

- (c) Ensures that the common data bits in the ISLU/RISLU universal control interface (UCI) are consistent with the data in the primary and secondary common data fields.

4. ERROR HANDLING

If the audit detects an inconsistency in the D-channel type field and common data configuration fields, the audit will report the error and change the common data configuration to agree with data in rISLUPIDB. The audit changes inconsistencies in the primary and secondary common data to agree with rISMEST.

If inconsistencies are found between common data bits in the ISLU/RISLU UIC and PCiscdconfig[] array, errors are reported and the common data bits are updated to reflect the status of PCiscdconfig[].

The CKTDATA, ISPIDBTS, ISLIDBTS, and PTSB audits are scheduled as related audits.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The ISLU common controller restoral bit in PCiscdconfig[] is not correct.
Possible Error Effect:	None.
Corrective Action Taken:	The ISLU common controller restoral bit will be restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCiscdconfig[].
<i>Bad Data =</i>	Content of ISLU common controller restoral flag.
<i>Good Data =</i>	Corrected ISLU common controller restoral value.
<i>Logical Index =</i>	Line unit number.
<i>Dump =</i>	Both common controller rISMEST tuples.

Table 2 Error Code: CCNON

Error Code:	CCNON
Error Description:	The common controller duplex fail array SMccdupfail[] is not correct.
Possible Error Effect:	None.

Corrective Action Taken:	The common controller duplex fail array will be restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMccdupfail[].
<i>Bad Data =</i>	Content of common controller duplex fail flag.
<i>Good Data =</i>	Corrected common controller duplex fail flag.
<i>Logical Index =</i>	Line unit number.
<i>Dump =</i>	Both common controller rISMEST tuples.

Table 3 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The rCDBCOM tuple associated with the ISLU common data Circuit or ISLU common control cannot be accessed due to a failure from dynamic access (DA) primitives or the key of the rCDBCOM tuple has been mutilated.
Possible Error Effect:	This may cause application code to assert when trying to access the rCDBCOM tuples associated with the ISLU common data and ISLU common control circuits during a removal or restoral of these circuits.
Corrective Action Taken:	The CKTDATA and CDBCOM audits will be scheduled as a related audit to recover the rCDBCOM tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rCDBCOM tuple.
<i>Bad Data =</i>	cktname.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 4 Error Code: CDBITS

Error Code:	CDBITS
Error Description:	The common data bits in the general processor control register or peripheral service state register is not correct.
Possible Error Effect:	None.
Corrective Action Taken:	The common data error in the UCI register will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Line unit number.
<i>Bad Data =</i>	Contents of registers.
<i>Good Data =</i>	Register contents should be.
<i>Logical Index =</i>	swno.
<i>Dump =</i>	NA.

Table 5 Error Code: CDCFG

Error Code:	CDCFG
Error Description:	The common data configuration in PCiscdconfig[] is not correct.
Possible Error Effect:	None.
Corrective Action Taken:	The common data configuration is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCiscdconfig[].
<i>Bad Data =</i>	Contents of the common data configuration in PCiscdconfig[].
<i>Good Data =</i>	Corrected common data configuration.
<i>Logical Index =</i>	Line unit number.
<i>Dump =</i>	rISLUPIDB tuple.

Table 6 Error Code: CDPRIM

Error Code:	CDPRIM
Error Description:	The common data primary configuration is not correct.
Possible Error Effect:	None.
Corrective Action Taken:	The common data primary configuration will be corrected.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of PCiscdconfig[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	swno.
<i>Dump</i> =	Both common data rISMEST tuples and PCiscdconfig[].

Table 7 Error Code: CDCEC

Error Code:	CDCEC
Error Description:	The common data secondary configuration is not correct.
Possible Error Effect:	If the common data packs are in the load shared configuration no paths will be set-up on the secondary side.
Corrective Action Taken:	The common data secondary configuration will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCiscdconfig[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	swno.
<i>Dump</i> =	Both common data rISMEST tuples and PCiscdconfig[].

Table 8 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The D-channel type in PCiscdconfig[] is not correct.
Possible Error Effect:	None.
Corrective Action Taken:	The D-channel type is restored to its correct type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCiscdconfig[].
<i>Bad Data</i> =	Contents of the D channel type in PCiscdconfig[].
<i>Good Data</i> =	Corrected D channel type.
<i>Logical Index</i> =	Line unit number.
<i>Dump</i> =	rISLUPIDB tuple.

Table 9 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The operational scanning bit (HWMSCAN) in the processor control register (HWPCTRL) of the unified control interface (UCI) is not enabled. This bit should only be inhibited during a common controller duplex failure.
Possible Error Effect:	Operational scanning may be turned off for this common data causing line card origination to be seen by the switch. Line card terminations can still be made.
Corrective Action Taken:	The operational scanning bit (HWMSCAN) in the processor control register (HWPCTRL) of the UCI will be enabled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the UIC register being read.
<i>Bad Data</i> =	Content of the operational scanning bit.
<i>Good Data</i> =	The operational scanning bit enabled value (HWMSCAN).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 10 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The common datas are in an inconsistent state (for example: cd1 - OOS and cd2 - STBY).
Possible Error Effect:	No call processing can take place.
Corrective Action Taken:	The CKTDATA audit will be scheduled as a related audit to recover the rISMEST tuples for the common datas to the proper state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCiscdconfig[].
<i>Bad Data</i> =	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	swno.
<i>Dump 1</i> =	Common data 0 rISMEST tuple.
<i>Dump 2</i> =	Common data 1 rISMEST tuple.
<i>Dump 3</i> =	PCiscdconfig[] array.

Table 11 Error Code: PIDB

Error Code:	PIDB
Error Description:	The key of rISLUPIDB generated from PClusgname[] is not correct.
Possible Error Effect:	Audit ISDCDCFG for this line unit will be skipped.
Corrective Action Taken:	The PICB audit will be scheduled to correct the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PICB number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Line unit number.
<i>Dump</i> =	NA.

Table 12 Error Code: RDREG

Error Code:	RDREG
Error Description:	The UCI register reading failed.
Possible Error Effect:	None.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Register address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	(1) General processor control register. (2) Peripheral service state register.
<i>Dump</i> =	NA.

Table 13 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The key of rISMEST tuple for the active common controller calculated for this equipped RISLU/ISLU was not found (no match from DB read).
Possible Error Effect:	Remainder of the ISDCFG audit for this line unit will be skipped.
Corrective Action Taken:	The CKTDATA audit will be scheduled to correct the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The key of rISMEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Line unit number.
<i>Dump</i> =	NA.

Table 14 Error Code: SMEST

Error Code:	SMEST
Error Description:	The key of rISMEST generated from PClusgname[] is not correct.
Possible Error Effect:	Audit ISDCDCFG for this line unit will be skipped.
Corrective Action Taken:	The PICB and CKTDATA audits will be scheduled to correct the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The key of rISMEST tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Line unit number.
<i>Dump</i> =	NA.

Table 15 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was invoked after taking a system reset.
Possible Error Effect:	Causes resets and/or ASSERTS.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Line unit number.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

ISILHB

Software Release: 5E10 and later

1. INCOMING LINE HISTORY BLOCK FOR INTEGRATED SERVICES DIGITAL NETWORK (ISDN) AND SHARED ANALOG LINES (ISILHB) AUDIT

ISDN incoming line history blocks are used by several of the LASS features to store incoming call data, associated with the ISDN and shared analog lines. The ISILHB audit performs dynamic to static existence checks of the dynamic RLIS_ILHB relation. The audit is also scheduled as a post initialization audit, as part of a selective initialization.

2. DATA STRUCTURES

The RLIS_ILHB relation is a dynamic indirect access nonsequential keyed relation. The ISILHB audit checks several pieces of static data to verify the proper existence of rlIS_ILHB tuples. The static data, referenced, is shown. Global Variables:

Gl_pcrd
 Glac_access
 Glcot_access
 Glascaac, Glscfac, Glscrac, Glldaac

Static Relations:

RLLPT_PT
 RLFC_LINE
 RLTRMFEATS
 DRINDEX

3. AUDIT CHECKS

The ISILHB audit steps through the RLIS_ILHB relation and verifies whether or not a tuple in the relation should exist.

4. ERROR HANDLING

If a rlIS_ILHB tuple exists that should not, the audit removes the tuple from the RLIS_ILHB relation.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rlAISILHB dynamic tuple without a corresponding static tuple or global parameter in the system. • rlISILHB.lport = rlLPT_PT.lport (with lpt_type = LPT_ISDN) • Glac_access Glcot_access is set to unavailable. • rlFC_LINE.ac_alw = DBNO (with rlFC_LINE.port = rlLPT_PT.port)

	• rIFC_LINE.cot_alw = DBNO (with rIFC_LINE.port = rILPT_PT.port)
Possible Error Effect:	If call processing attempts to access this tuple, services/operations may be disrupted.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rILSILHB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rILSILHB tuple in question.
<i>Dump =</i>	rILSILHB tuple in question.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has found an inconsistency between the static relations rILPT_PT and rIFC_LINE. The audit is using the port and party fields of a rILPT_PT tuple to access a tuple in rIFC_LINE, however it is not able to access the rIFC_LINE tuple so the audit cannot continue to do checks on this tuple.
Possible Error Effect:	The rILSILHB audit will not be able to complete its checks on this tuple.
Corrective Action Taken:	The audit will print an error and continue onto the next tuple.
Manual Action Required:	Check the rILPT_PT and rIFC_LINE relations in the ODD. The rILPT_PT key, lport, found in the "GOOD DATA" field of the audit report, should be used to access the rILPT_PT tuple. The rILPT_PT tuple should have a port and party field which serve as the keys to the rIFC_LINE relation. There should exist a corresponding tuple in rIFC_LINE with the port and party found in rILPT_PT.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the port field of the buffer used to access the rIFC_LINE tuple.
<i>Bad Data =</i>	Party field of the buffer used to access the rIFC_LINE tuple.
<i>Good Data =</i>	lport field of the rILPT_PT tuple used access the associated rIFC_LINE tuple.
<i>Logical Index =</i>	The key of the rILSILHB tuple in question.
<i>Dump 1 =</i>	rILSILHB tuple in question.
<i>Dump 2 =</i>	Associated rILPT_PT tuple.
<i>Dump 3 =</i>	Buffer used to access rIFC_LINE.

ISLIDBTS

Software Release: 5E10 and later

1. ISLU LINE CARD INTERFACE DATA BUS TIME SLOT (ISLIDBTS) AUDIT

The line card interface data bus (LIDB) is a data bus that carries two types of call data. Pulse code modulation (PCM) voice data is carried over 64 kilobit/second B-channels. Protocol control data is carried over 16 Kbs D-channels, or one-quarter time slots. The LIDBs connect the integrated services line unit (ISLU) line group controllers (LGCs) with the common data (CD) circuits. The line cards connect to the line group controllers through group interface data buses (GIDBs).

The ISLIDBTS audit is part of the peripheral control audits subgroup. Refer to the **AUDIT CHECKS** section for more information about the purpose of the ISLIDBTS audit.

2. DATA STRUCTURES

Figure 1 shows the data structures the audit uses. The primary data structure checked by the audit is the peripheral control ISLU line card interface data bus time slot array (PCislidbts). The array is made up of bit map arrays containing the statuses of completely idle time slots and partially full time slots for line card interface databuses (LIDBs) associated with the line group controllers, and static masks for direct D-channel reserved time slot path hunting. The array also contains the relative peripheral interface data bus (PIDB) number associated with each of the 32 LIDB time slots and the maintenance state of each LGC, which is a copy of the base state from the switch maintenance equipment status table relation (RLSMEST). The RLSMEST relation contains the equipment status (active, out-of-service, and so forth) for each circuit in the SM. Circuits of interest to the ISLIDBTS audit are the LGCs or STIs. Other data structures the audit uses include the peripheral control line unit service group name array (Pclusgname) and the PIDB time slot block relation (RLPTSB). The Pclusgname array holds a mask containing the upper eight bits [peripheral interface control bus (PICB) number plus one] for circuits in a line unit service group. The RLPTSB relation contains busy/idle path information for each of the four one-quarter time slots (D-channels) in a full 64 Kbs PIDB time slot. It also holds B-channel information, depending on the use of the PIDB time slot. The relation contains the digital subscriber line (DSL) group number for the PIDB time slot as well as port names for each one-quarter time slot member, transient timing flags for each one-quarter time slot, and linkage fields.

The audit also uses the following relations. The ISLU PIDB relation (RLISLUPIDB) contains the type of D-channels (direct or cut through), CD configuration of the ISLU (active/standby or active/active), and an array of relative PIDBs assigned for the ISLU. The key is the PICB assigned for the ISLU. The ISLU LGC relation (RLISLULGC) contains a line card array that indicates the type of line cards used in the ISLU, and the static direct PIDB path hunt mask for each LGC in the ISLU. The DSL equipage relation (RLEQUIPDSL) contains the DSL group and member assigned for a particular D-channel port. The circuit data relation (RLCKTDATA) contains the equipage state of the circuits in the SM (operational, unequipped, growth, and so forth). The circuits of concern are the LGCs.

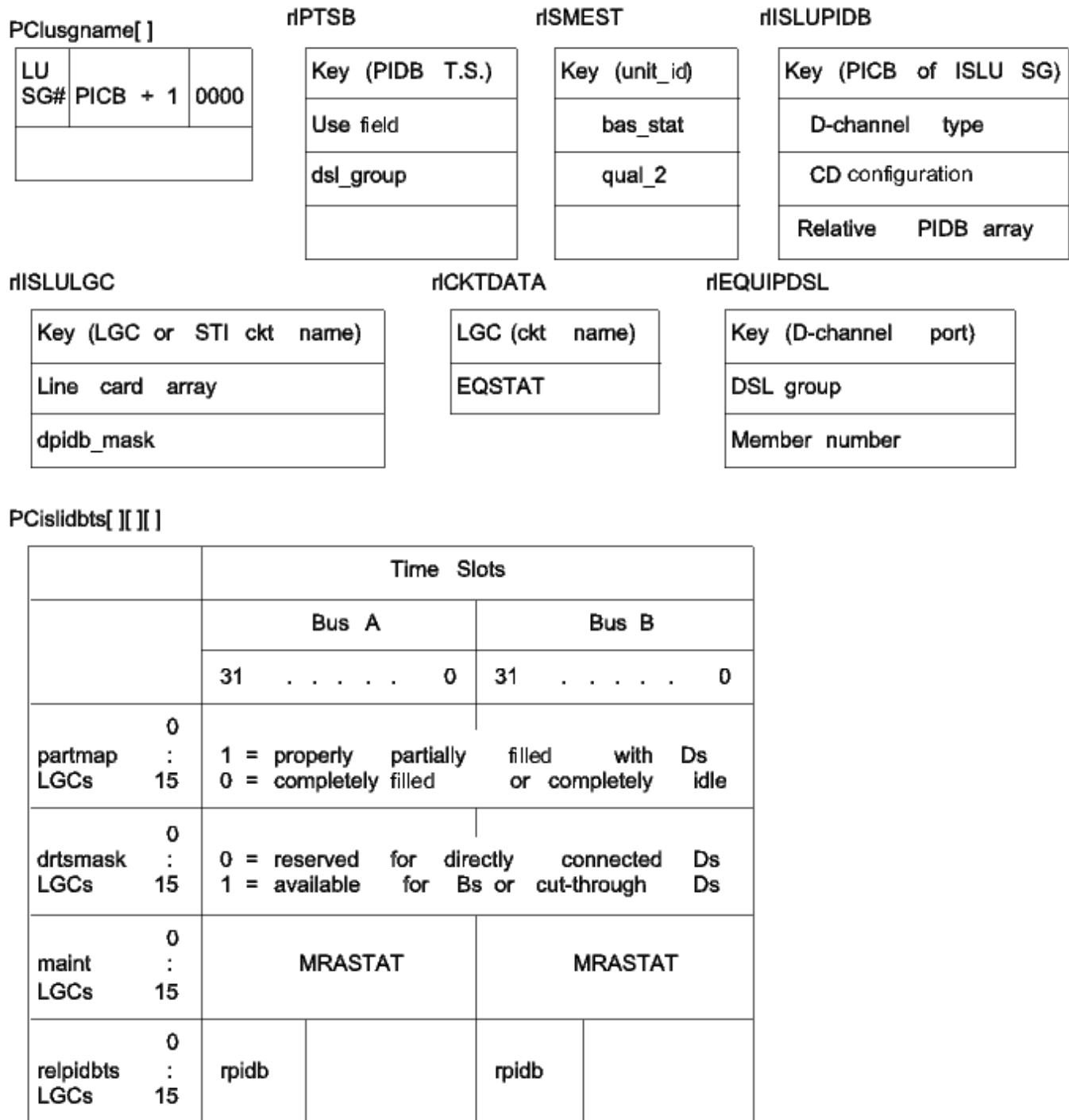


Figure 1 (ISLIDBTS-1) Detailed Data Structures

3. AUDIT CHECKS

The audit steps through the eight possible line units in the SMs and makes the following checks for the ISLU, and RISLU, and DCLU-5.

- Loop through all 32 LIDB time slots on the LGC or STI, taking a time break after auditing groups of four time slots, checking each common data (CD) side (service group 0 or 1) in the ISLU.

- (b) Read rCKTDATA to obtain the equipage state of the STI or LGC. If it is unequipped or unpopulated, the PCislidbts map maintenance state for the LGC/STI should be out-of-service (OOS). If it is not OOS, an error is reported. If the LGC/STI is operational, the audit moves on to make further checks.
- (c) Read rISLUPIDB using PClusname to get the PICB number. If the read fails, report a manual action ODD error. If the read is successful, the audit saves the data and proceeds.
- (d) Read rISLULGC. If the read fails, report a manual action ODD error. If the read succeeds, verify the static copy of the directly connected PIDB time slot path hunt mask with the dynamic copy stored in the PCislidbts[][] map. If they are inconsistent, the audit reports an error and updates the dynamic copy.
- (e) Read rSMEST for the LGC. If the LGC is not MRA BUSY, check the maintenance state of the LGC in PCislidbts[][] against the rSMEST base state. If it is in error, report the error and recover the maintenance state to that of rSMEST. If the read of rSMEST fails, report the error and schedule the CKTDATA audit.

NOTE: The following checks are made for each time slot. The state of the PCislidbts map is represented as idlemmap, partialmap, and relpidbts.

- (a) Check that LGC control time slots are marked complete, full, or null. If they are not marked, (report an error) and recover the time slot to the proper state.
- (b) Check that a null relative PIDB time slot entry has a corresponding value of complete, idle, or null in PCislidbts. If it does not, (report error) and recover.
- (c) For non-null relative PIDB time slot entries, get the PIDB type from rISLUPIDB. Construct the key of the corresponding rIPTSB tuple based on the type of PIDB (CD configuration, active/active or active/standby). If there is no PIDB type, recover the LIDBTS time slot to complete/idle/null.
- (d) Perform a key check of the rIPTSB tuple. If there is an error, recover the PCislidbts time slot to complete/idle/null.
- (e) Check the usage of the rIPTSB tuple.
 - (1) If the rIPTSB tuple usage is null, recover the associated time slot because relative PIDB is non-null.
 - (2) If it is used for duplex or simplex B-channels, verify that the port name in the rIPTSB matches this LGC or STI. If it does not match, recover the LIDBTS time slot. Next, verify that the LIDBTS map is complete/full/non-null. If it is not, (report error) and recover LIDBTS to complete/full/non-null.
 - (3) If the tuple is used for D-channels or packet B-channels, check to see if any transient conditions exist. If a transient condition exists, move on to next time slot. If a transient condition does not exist, count the number of busy one-quarter time slots in rIPTSB. If a member is assigned a DSL group, get the DSL group number from rIPTSB and read rEQUIPDSL to verify it. If the read fails, (report an error) and recover PCislidbts to complete/idle/null. If the read succeeds, check that the D-channel port matches this LGC or STI. If it does not, (report an error) and recover PCislidbts to complete/idle/null. If no member is assigned, verify that the number of busy one-quarter time slots is consistent with the status of the PCislidbts map. If it is inconsistent, recover PCislidbts to the correct state.
 - (4) If it is used for maintenance, verify that the B-channel port name in rIPTSB matches this LGC or STI. If it does not, (report an error) and recover PCislidbts to complete/idle/null. Next, verify that PCislidbts[][] is complete/full. If it is not, report the error and recover to complete/full.
 - (5) If the rIPTSB tuple is used for a packet switch unit (PSU) PIDB time slot, report an error and recover PCislidbts[][] to complete/idle/null.

- (6) If the usage field is out of range, report an error and recover PCislidbts to complete/idle/null.

4. ERROR HANDLING

The audit in general handles errors as follows. For details, see the explanations of the checks in the Audit Checks section. The audit changes inconsistencies between rISMEST and the maintenance state in PCislidbts to the value in rISMEST. The audit sets the control time slot bit to busy when it detects control time slot errors. For other time slot bit inconsistencies, the audit sets the idlemap/partmap status to idle/complete. The corresponding rIPTSB tuples will be recovered by the peripheral interface data bus time slot block (PTSB) audit.

The audit schedules the CKTDATA and PTSB audits as related audits when it detects errors.

The audit produces the following error code(s).

Table 1 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The dynamic copy of DPIDB time slot mask does not match the static copy.
Possible Error Effect:	The mask is out of date.
Corrective Action Taken:	Causes an interrupt in checking the data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of LIDBTS entry.
<i>Bad Data =</i>	Dynamic copy of DPIDB reserved TS mask.
<i>Good Data =</i>	Static copy of DPIDB reserved TS mask.
<i>Logical Index =</i>	LGC or STI circuit name.
<i>Dump =</i>	NA.

Table 2 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	Static data inconsistency between rIEQUIPDSL.
Possible Error Effect:	There could be a loss of resources or possible path hunt problems.
Corrective Action Taken:	Recover the status of the relative PIDB to idle and set idlemap to 1 and partmap to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the LIDBTS entry.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LGC or STI circuit name.
<i>Dump =</i>	LIDB map entry.

Table 3 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The partmap is not completely full and the idlemap is marked idle.
Possible Error Effect:	Call will not be completed correctly.
Corrective Action Taken:	Resets the partmap and idlemap to the correct status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of LIDBTS entry.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LGC or STI circuit name.
<i>Dump =</i>	rIPTSB tuple and the LIDB map entry.

Table 4 Error Code: KEY

Error Code:	KEY
Error Description:	The key in the rIPTSB tuple does not match the key passed.
Possible Error Effect:	The rIPTSB tuple has probably been mutilated.

Corrective Action Taken:	Recover the correct state of the maps for this time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	Contents of rIPTSB tuple (dynamic).
<i>Good Data</i> =	Contents of rIPTSB tuple (static).
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	NA.

Table 5 Error Code: LGC_CTS

Error Code:	LGC_CTS
Error Description:	The relative PIDB fields are not idle or the partial and idle maps are not equal to zero.
Possible Error Effect:	CTS (control time slot) may be invalidly used for path hunt.
Corrective Action Taken:	Recover the CTS by setting the idlemap, partialmap, and relpidb.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 6 Error Code: LGCMAINT

Error Code:	LGCMAINT
Error Description:	(1) The LGC is not operational for rICKTDATA and the maintenance field is not marked OOS. (2) The LIDB map should be marked in-service for the rISMEST.
Possible Error Effect:	Operational code may improperly access the circuit.
Corrective Action Taken:	Recover the LIDB map to SMOOS for rICKTDATA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the LIDBTS entry.
<i>Bad Data</i> =	Maintenance state found in LIDBTS map entry.
<i>Good Data</i> =	Corresponds to Error Description as follows:
	(1) OOS base state value. (2) ACT base state value.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	NA.

Table 7 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The LGC name from the rIPTSB tuple does not match the LGC currently being audited.
Possible Error Effect:	Call will not be completed correctly.
Corrective Action Taken:	Idle the partial and idlemaps and also relpidb and continue to next time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	rIPTSB tuple and the LIDB map entry.

Table 8 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The idlemaps and partmap should be in the busy state.
Possible Error Effect:	Loss of resources.
Corrective Action Taken:	Print an error and set idlemaps and partmaps to busy state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 9 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	When the rIPTSB tuple is idle, the relpidb in the LIDB map is idle.
Possible Error Effect:	There will be a loss of resources and possible path hunt problems.
Corrective Action Taken:	Recover the LIDB map entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Contents of the rIPTSB tuple.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 10 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	CD configuration other than shared load and active/standby is used which is not known by audit.
Possible Error Effect:	CTS will be invalidly used for path hunt.
Corrective Action Taken:	Print an error and continue to the next LGC or STI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 11 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Static data inconsistency with rICKTDATA.
Possible Error Effect:	Loss of resources and path hunt will be affected.
Corrective Action Taken:	None.
Manual Action Required:	Find the nature of the inconsistency and correct it. This error will continue until corrected.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	pidb circuit name.
<i>Dump</i> =	NA.

Table 12 Error Code: PIDBTS

Error Code:	PIDBTS
Error Description:	The data in the LIDB map (PCislidbts[] []) indicated that this time slot was being used for line card diagnostics. However, either the rISMEST tuple associated with the line card was not found to be busy, (stat_act == DBBUSY), the idlemmap indicates idle, or the partmap indicates that it is not completely full
Possible Error Effect:	The LIDB time slot that was under audit will be lost to the system and can lead to degradation of ISDN type calls.

Corrective Action Taken:	Set the idlemap for this time slot to idle (1), set the partmap top indicate completely idle (0), and set the relative pidb field to idle (0xf). This will be done for both sides of the map. To insure proper recovery in the event that the rISMEST tuple was in error, the CKTDATA audit will be scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDB map found to be in error (PCidlibts[lu] [gc]).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC ircuit name.
<i>Dump</i> =	NA.

Table 13 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	(1) The port name from the rIPTSB tuple is not on the specified LGC. (2) The port name from the rEQUIPDSL tuple is not on the specified LGC.
Possible Error Effect:	There will be a loss of resources and possible path hunt problems.
Corrective Action Taken:	Idle all the ports and continue to the next time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	Dynamic value of LGC.
<i>Good Data</i> =	Static value of LGC.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 14 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The data in the LIDB map (PCislibts[] []) indicated that this time slot was being used for line card diagnostics. However, the rISMEST tuple associated with the line card was not found in the data base by the Data Base Manager (DBM). Either the data in the LIDB map was corrupt or the rISMEST tuple is erroneously missing from the data base.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Set the idlemap for this time slot to idle (1), set the partmap to indicate completely idle (0), and set the relative pidb field to idle (0xf). This will be done for both sides of the map. To insure proper recovery in the event that the rISMEST tuple is missing from the data base, the CKTDATA audit will be scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDB map found to be in error (PCidlibts[lu] [gc]).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC circuit name.
<i>Dump</i> =	NA.

Table 15 Error Code: RELPIDB

Error Code:	RELPIDB
Error Description:	The relpidb fields do not match for active/standby check.
Possible Error Effect:	The standby bus copy will not match the active bus copy.
Corrective Action Taken:	Update the standby bus copy so it matches the active bus copy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.

<i>Bad Data</i> =	relpidb fields for side 1.
<i>Good Data</i> =	relpidb fields for side 0.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	NA.

Table 16 Error Code: SMEST

Error Code:	SMEST
Error Description:	Unable to read rSMEST.
Possible Error Effect:	rCKTDATA is able to be read but rSMEST is unable to be read.
Corrective Action Taken:	Schedule the SMEST audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	NA.

Table 17 Error Code: STATE

Error Code:	STATE
Error Description:	Either the idlemmap is not busy or the partmap is full for the LGC or STI.
Possible Error Effect:	Loss of resources and possible path hunt problems.
Corrective Action Taken:	Print an error and recover the maps to the correct states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 18 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	For PCPPSCUT (PSU default) case maps are set to incorrect states.
Possible Error Effect:	Loss of resources and possible hunt path problems.
Corrective Action Taken:	Print an error and recover the maps to idle state and then continue onto the next time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	Contents of the use field in the rIPTSB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	rIPTSB tuple and the LIDB map entry.

Table 19 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	When the LIDBTS is used for maintenance, the partmap is not full or idlemmap is not busy.
Possible Error Effect:	Loss of resources or possible path hunt problems.
Corrective Action Taken:	Recover the maps to busy and report an error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	rIPTSB tuple and LIDB map entry.

Table 20 Error Code: TSIDLE_BUSY

Error Code:	TSIDLE_BUSY

Error Description:	With the relative PIDB field set to 0xf, the LIDB map is not set to idle.
Possible Error Effect:	CTS may be invalidly used for path hunt.
Corrective Action Taken:	Recover the CTS by setting the idlemmap, and partmap to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LIDBTS entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	LIDB map entry.

Table 21 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was invoked after taking a system reset.
Possible Error Effect:	Causes system reset and/or asserts to fire.
Corrective Action Taken:	The audit has been returned to audit control and will be rescheduled for recovery activities.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LGC or STI circuit name.
<i>Dump</i> =	NA.

ISLINE

Software Release: 5E10 and later

1. INTEGRATED SERVICES LINE (ISLINE) AUDIT

The purpose of the ISLINE audit is to perform static-to-dynamic existence checks between rIIS_LINE and rIPPSAST. For each tuple in rIIS_LINE that has the lead module equal to the switching module (SM) and the port processor (PP) bit, RLis_line.pp_stat is equal to DBYES, the rIPPSAST tuple will be checked for existence. Any missing tuples in rIPPSAST are inserted.

Another added purpose of the ISLINE is to maintain data consistency between the static rIIS_LINE tuple and the dynamic rICPNBNCNT tuple.

2. DATA STRUCTURES

The ISLINE audit runs in the segmented mode and operates in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The purpose of this audit is to perform static-to-dynamic existence checks between rIIS_LINE and rIPPSAST. For each tuple in rIIS_LINE that has the lead module equal to that SM and the PP feature indicator and RLis_line.pp_stat equal to DBYES, the rIPPSAST tuple is checked for existence. Any missing dynamic tuple in rIPPSAST is inserted.

The audit also performs a check on the existence of the dynamic rICPNBNCNT tuple when the "bill_opt" attribute is set to USAGE_SENS in the rIIS_LINE tuple.

4. ERROR HANDLING

If no tuple is found in rIPPSAST and rIIS_LINE indicates it should exist, and if recent change is not active on rIIS_LINE, the missing tuple is inserted and an error message is printed. An assert is fired if the database insert fails. The PP flags are initialized to zero.

If the ISLINE audit finds an error, then the CPNBNCNT audit is run as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the dynamic RLPPSAST/RLCPNBNCNT relation for a corresponding static RLIS_LINE relation. Cannot establish priority call for missing rIPPSAST tuple. The BAD DATA field in the dump indicate the missing dynamic tuple: 997 is the relid for rIPPSAST and 1291 is for rICPNBNCNT.
Possible Error Effect:	Inability to do existence checking between ISLINE and PPSAST and ASSERT failures due to call termination. If missing rICPNBNCNT, then lost information for billing.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIIS_LINE tuple.
<i>Bad Data =</i>	Relation ID of missing tuple.
<i>Good Data =</i>	0 or one key of lost rICPNBNCNT tuple.

<i>Logical Index</i> =	Key of lost tuple.
<i>Dump</i> =	Dump of the rllS LINE tuple.

ISLPCB

Software Release: 5E14 and later

1. ISLP CONNECTION CONTROL BLOCK (ISLPCB) AUDIT

The ISLPCB is the control block for the ISLP channels and contains the host-to-DSP and DSP-to-host ISLP buffer descriptors (ISLPBD). This audit verifies the integrity of the host-to-DSP and DSP-to-host pointers to the head and tail queues.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: PTR1

Error Code:	PTR1
Error Description:	One or more of the host-to-DSP or DSP-to-host head and tail pointers are mis-aligned.
Possible Error Effect:	Invalid data may be passed for a call.
Corrective Action Taken:	The head pointers are reinitialized to point to the first entry in the arrays and the tail pointers to the last entry in the arrays.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Host-to-DSP address (PVislpcb[index].hd_queue.hdq[0]).
<i>Bad Data =</i>	DSP-to-host address (PVislpcb[index].dh_queue.dhq[0]).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The array number that failed.
<i>Dump =</i>	PVislpstack[index].

ISLUTRAN

Software Release: 5E10 and later

1. ISLU TRANSACTION (ISLUTRAN) AUDIT

The integrated services line unit (ISLU) Z analog interface (ISLU-Z) provides analog call processing on the ISLU, including plain old telephone service (POTS) calls, multi-party calls, and coin phone operations on the ISLU. Several audits check the data structures used by the ringing generators (RGs), metallic access network (MAN), and sparing of faulty line cards. The audits include the peripheral control (PC) ringing occupancy table audit (RGOCUP), the PC ringing task data block audit (RGTDDB), the ringing generator bus audit (RGBUS), the channel control block user audit (CCBUSR), the ISLU-Z access network bus audit (ISANBUS), the spare line card audit (SPRCRD), and the ISLU transaction audit (ISLUTRAN). The ISLUTRAN audit checks the global PC array that contains information about the line units that are ISLUs or remote ISLUs (RISLUs). The ISLUTRAN audit is part of the peripheral control audits subgroup.

2. DATA STRUCTURES

The ISLUTRAN audit checks the information in the PC ISLU transactions array (PCislutrans). The PCislutrans array has eight indices that indicate which line units in the SM are ISLUs or RISLUs. For example, if line units 4 and 6 are ISLUs, then PCislutrans[4] = 4 and PCislutrans[6] = 6. The rest of the indices would be initialized to 0xffff. The PCislutrans array is used by operational code to quickly determine which line unit to use for ringing.

3. AUDIT CHECKS

IDL_STATE = PCislutrans[] entry should be empty but is not.

STATUS = PCislutrans[] entry is incorrect for an ISLU/RISLU.

ZAP = The audit was entered after a memory violation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: IDL_STATE

Error Code:	IDL_STATE
Error Description:	The PCislutrans global array indices that should be 0xffff, meaning that there no more RISLU/ISLU's in the switch, have some other number in them.
Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The PCislutrans index is set to 0xffff.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCislutrans array entry.
<i>Bad Data</i> =	Contents of PCislutrans entry.
<i>Good Data</i> =	0xffff.
<i>Logical Index</i> =	Index into PCislutrans.
<i>Dump</i> =	PCislutrans array.

Table 2 Error Code: STATUS

Error Code:	STATUS
Error Description:	The PCislutrans global array does not contain the right information about which LUs are (R)ISLU'S.

Possible Error Effect:	Ringing data has been made invalid.
Corrective Action Taken:	The (R)ISLU number is put into the appropriate PCislutrans index.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCislutrans array entry.
<i>Bad Data =</i>	Contents of PCislutrans entry.
<i>Good Data =</i>	(R)ISLU number.
<i>Logical Index =</i>	LU number.
<i>Dump =</i>	PCislutrans array.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was invoked after taking a system reset.
Possible Error Effect:	Causes asserts to fire and/or a system reset.
Corrective Action Taken:	The audit has been returned to audit control and will be rescheduled for recovery activities.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

ISMDB

Software Release: 5E10 and later

1. INTEGRATED SERVICES DIGITAL NETWORK (ISDN) MAINTENANCE DATA BLOCK (ISMDB) AUDIT

The ISMDB relation is a dynamic indexed relation, which is used for all applications that support ISDN protocols.

2. DATA STRUCTURES

The data structure checked by the ISMDB audit is the ISDN maintenance data block relation (RLISMDB).

3. AUDIT CHECKS

ISMDB audit performs existence check between the dynamic data in RLISMDB and the static data in RLEQUIPDSL or RLPSX25LN.

4. ERROR HANDLING

Any extra ISMDB tuple is marked as deleted and initialized.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP					
Error Description:	The audit has detected an rIDPB dynamic tuple without a corresponding static tuple where: rISMDB Key Format 1514131211109876543210 <table><tr><td>0</td><td>0</td><td>DSL Group</td><td>0</td><td>DSL Member</td></tr></table> (1) rISMDB.dslgm.DSL_group = rEQUIPDSL.dsl_group rISMDB.dslgm.DSL_member = rEQUIPDSL.member (2) rISMDB.dslgm.DSL_group = rlpsx25ln.ph_number rISMDB.dslgm.DSL_member = rlpsx25ln.ph_ch	0	0	DSL Group	0	DSL Member
0	0	DSL Group	0	DSL Member		
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources for protocol monitoring, operations may be disrupted.					
Corrective Action Taken:	Tuple is deleted.					
Manual Action Required:	None.					
Dump Description:						
Error Address =	Address of the tuple.					
Bad Data =	NA.					
Good Data =	NA.					
Logical Index =	PSX25LN key.					
Dump =	RLPSX25LN.					

ISOLHB

Software Release: 5E10 and later

1. OUTGOING LINE HISTORY BLOCK FOR INTEGRATED SERVICES DIGITAL NETWORK (ISDN) AND SHARED ANALOG LINES (ISOLHB) AUDIT

ISDN outgoing line history blocks are used by several of the LASS features to store outgoing call data, associated with the ISDN and shared analog lines. The ISOLHB audit performs dynamic to static existence checks of the dynamic RLIS_OLHB relation. It is also scheduled as a post initialization audit, as part of a selective initialization.

2. DATA STRUCTURES

The RLIS_OLHB relation is a dynamic indirect access nonsequential keyed relation. The ISOLHB audit checks several pieces of static data to verify the proper existence of rLIS_OLHB tuples. The static data, referenced, is shown.

3. AUDIT CHECKS

The ISOLHB audit steps through the rLIS_OLHB relation and verifies whether or not a tuple in the relation should exist.

4. ERROR HANDLING

If a rLIS_OLHB tuple exists that should not, the audit removes the tuple from the RLIS_OLHB relation.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rLISOLHB dynamic tuple without a corresponding static tuple or global parameter in the system. • rLANOLHB.lport = rLPT_PT.lport (with lpt_type = LPT_ISDN). • Glar_access is set to unavailable. • rIFC_LINE.ar_alw = DBNO (with rIFC_LINE.port = rLPT_PT.port)
Possible Error Effect:	If call processing attempts to access this tuple, services/operations may be disrupted.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the IS_OLHB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rLIS_OLHB tuple in question.
<i>Dump =</i>	rLIS_OLHB tuple in question.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has found an inconsistency between the static relations rLPT_PT and rIFC_LINE. The audit is using the port and party fields of a rLPT_PT tuple to

	access a tuple in rIFC_LINE, however it is not able to access the rIFC_LINE tuple so the audit cannot continue to do checks on this tuple.
Possible Error Effect:	The ISOLHB audit will not be able to complete its checks on this tuple.
Corrective Action Taken:	The audit will print an error and continue on to the next tuple.
Manual Action Required:	Check the rLPT_PT and rIFC_LINE relations in the ODD. The rLPT_PT key, lport, found in the "GOOD DATA" field of the audit report, should be used to access the rLPT_PT tuple. The rLPT_PT tuple should have a port and party field which serve as the keys to the rIFC_LINE relation. There should exist a corresponding tuple in rIFC_LINE with the port and party found in rLPT_PT.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the port field of the buffer used to access the rIFC_LINE tuple.
<i>Bad Data =</i>	Party field of the buffer used to access the rIFC_LINE tuple.
<i>Good Data =</i>	lport field of the rLPT_PT tuple used access the associated rIFC_LINE tuple.
<i>Logical Index =</i>	The key of the rIIS_OLHB tuple in question.
<i>Dump 1 =</i>	rIIS_OLHB tuple in question.
<i>Dump 2 =</i>	Associated rLPT_PT tuple.

ISPABUF

Software Release: 5E10 and later

1. INTEGRATED SERVICES PORT ADMINISTRATION BUFFER (ISPABUF) AUDIT

There are several types of ports in the switch that need administering, including line unit ports, *SLC*® 96 ports, and integrated service digital network (ISDN) B- and D-channel ports. The integrated services port administrator (ISPA) takes care of port functions such as routine conditioning to ensure that the hardware and software are consistent. Two audits check the tuple buffers associated with the system and terminal processes involved with port administration, the ISPABUF audit the terminal process control buffer (TPCBUF) audit.

The ISPABUF audit is part of the terminal maintenance audits subgroup.

2. DATA STRUCTURES

The ISPABUF audit uses two data structures: the ISPA buffer relation (RLISPABUF) and the terminal process buffers relation (RLTPCBUF). The RLISPABUF relation ensures that different ISPA functions with different priorities are scheduled properly. It also is used for creating and monitoring the terminal processes. The audit also refers to the data base head cell relation (RLDB_HEADC), which contains tuples for the different priority levels in RLISPABUF. These data structures are shown in Figure 1.

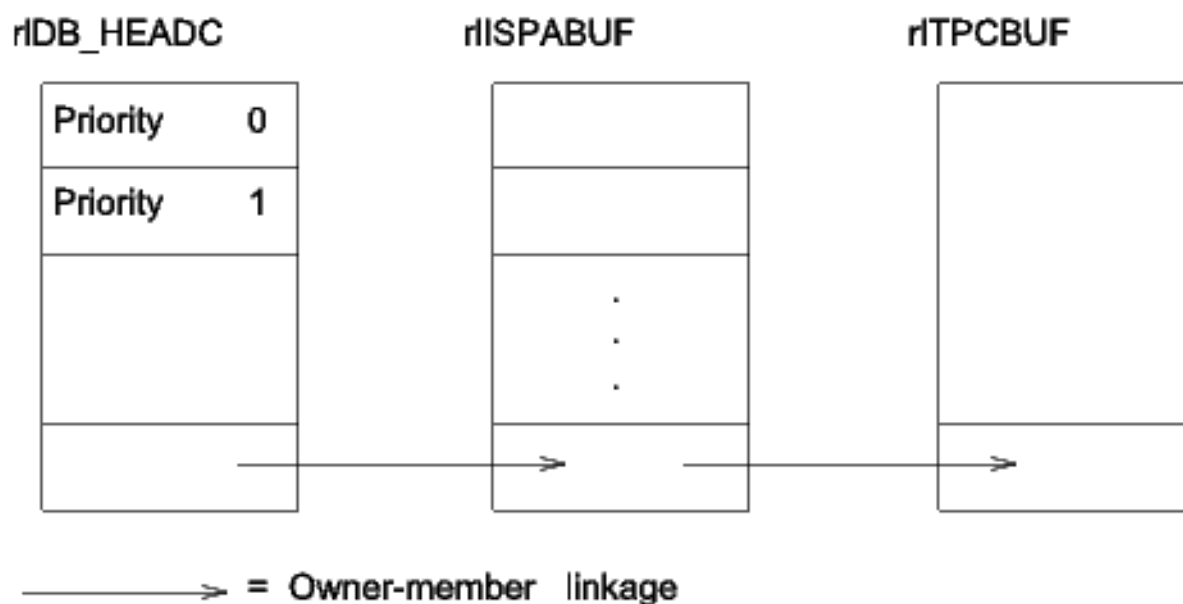


Figure 1 (ISPABUF-1) Data Structures of the ISPABUF Audit

3. AUDIT CHECKS

The ISPABUF audit invokes the linkage driver (DRLINK) to check the linkages of RLISPABUF to RLTPCBUF and RLDB_HEADC. It calls the semantic function to check the range of the priorities in RLISPABUF. The semantic function also ensures that tuples of a certain priority are linked to the proper tuple of RLDB_HEADC.

4. ERROR HANDLING

If the audit detects an error with the busy lists, the list is considered critical. The audit will patch the list, if possible.

The ISPABUF audits schedules the TPCBUF and DBHEADC audits as related audits when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The rISPABUF tuple has a job priority of TMPAUDIT (input.priority) , which is reserved for routine port conditioning (RPC) jobs. However the tuple is not being used for the RPC job(input.job not equal to TMRTNCOND).
Possible Error Effect:	ISPA could be preventing RPC jobs from running.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rISPABUF tuple.
<i>Bad Data =</i>	The current ISPA job type, input.job .
<i>Good Data =</i>	The ISPA job type for RPC jobs, TMRTNCOND.
<i>Logical Index =</i>	Key of the rISPABUF tuple, bufkey .
<i>Dump =</i>	The rISPABUF tuple.

Table 2 Error Code: PROCPRIOR

Error Code:	PROCPRIOR
Error Description:	The priority attribute found in the rISPABUF tuple is out of range of the possible priority values.
Possible Error Effect:	ISPA could be scheduling jobs at the wrong priority.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of input structure in rISPABUF tuple.
<i>Bad Data =</i>	Priority found in rISPABUF tuple.
<i>Good Data =</i>	Maximum priority value allowed.
<i>Logical Index =</i>	Key of rISPABUF tuple, bufkey .
<i>Dump =</i>	rISPABUF tuple.

Table 3 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	Only one ISPA Routine Port Conditioning(RPC) job is allowed to be scheduled at a given time. Audit found that the current rISPABUF tuple under audit is not the first and only tuple on the list.
Possible Error Effect:	Someone other than the RPC scheduler may be requesting RPC jobs.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad rISPABUF tuple.
<i>Bad Data =</i>	Logical key of the previous rISPABUF tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the bad rISPABUF tuple, bufkey .
<i>Dump =</i>	The rISPABUF tuple.

Table 4 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The linkage field in the rISPABUF tuple indicates the tuple is on one of the priority queues, but the priority attribute indicates the tuple is on a different priority queue.
Possible Error Effect:	ISPA could be scheduling jobs at the wrong priority.
Corrective Action Taken:	The rISPABUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of owner linkage in rISPABUF tuple.
<i>Bad Data =</i>	Key of owner tuple found in rISPABUF tuple linkage.

<i>Good Data</i> =	Key of owner tuple converted from priority attribute through use of TMispr_i_hc[] array.
<i>Logical Index</i> =	Key of rISPABUF tuple, bufkey .
<i>Dump</i> =	rISPABUF tuple.

ISPIDBTS

Software Release: 5E10 and later

1. ISLU PERIPHERAL INTERFACE DATA BUS TIME SLOT (ISPIDBTS) AUDIT

The integrated services line unit (ISLU) peripheral interface data bus (PIDB) time slot map array contains the busy/idle status of the PIDB time slots, from the common data (CDs) to the time slot interchanger (TSI), and a record of the remaining available time slots. It contains the PIDB type, maintenance state, and the data interface service group number. The data structure is audited by the PIDBTS audit to ensure that the status of the bits in the array are consistent with the hardware state. This map is used for ISLUs, remote integrated services line units (RISLUs), and directly connected line units-5 (DCLU-5s).

The ISPIDBTS audit is part of the peripheral control audits subgroup.

Refer to the **AUDIT CHECKS** section for more information about the use and purpose of the ISPIDBTS audit.

2. DATA STRUCTURES

The main data structure checked by the audit is the peripheral control ISLU PIDB time slot map array (PCispidbts). The audit obtains the PIDB circuit name from the line unit service group name array (Pclusgname). The PIDB type is checked against data contained in the circuit data relation (RLCKTDATA). Several fields are checked against the switch maintenance equipment status table relation (RLSMEST). The audit also checks the B- and D-channel time slots against information contained in the PIDB time slot block relation (RLPTSB). These data structures are shown in Figure 1 .

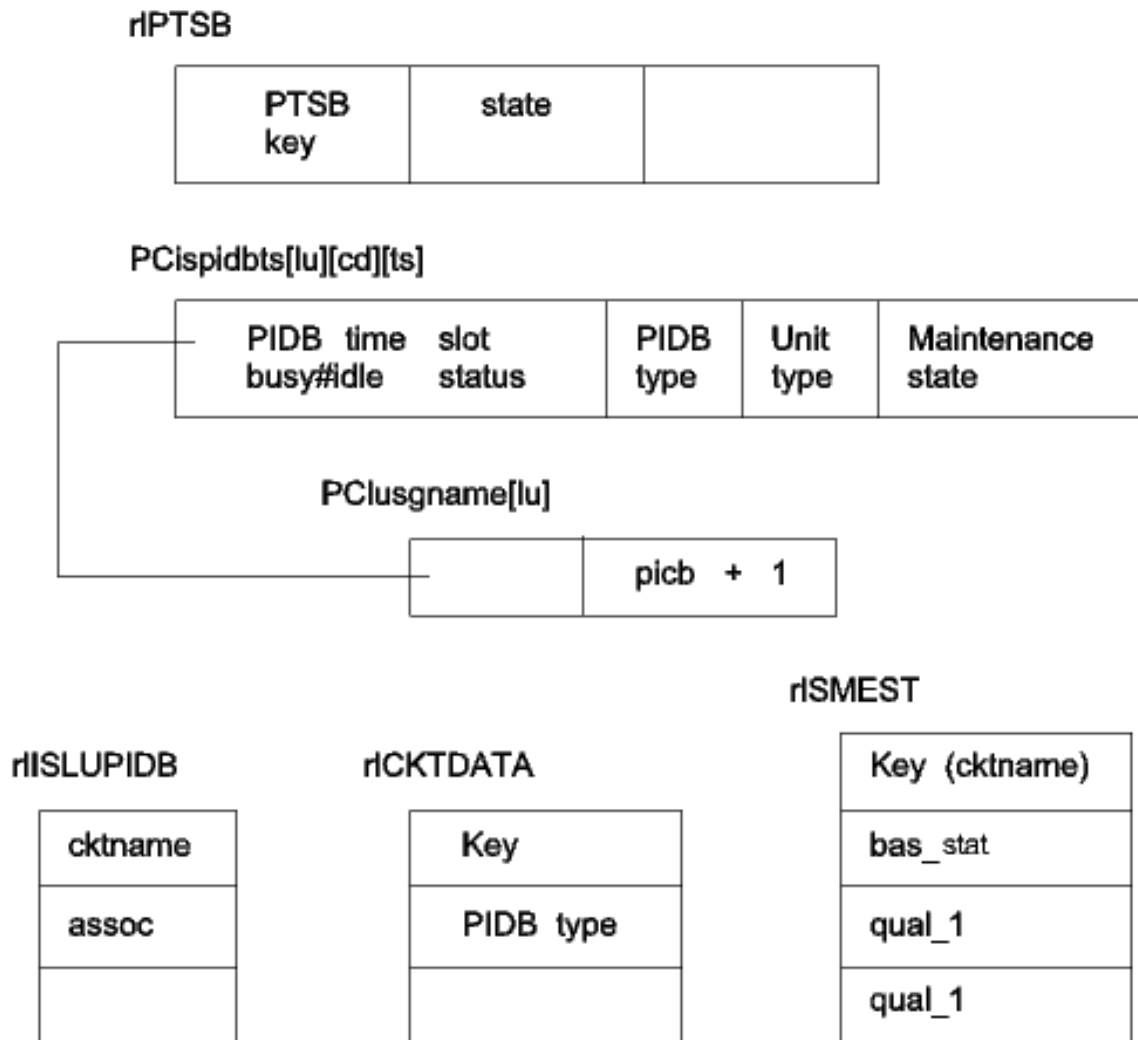


Figure 1 (ISPIDBTS-1) Data Structures

3. AUDIT CHECKS

For each PCispdbts array entry, the audit makes the following checks:

- Check that the unit is an ISLU, RISLU, or DCLU-5. If it is not, the audit proceeds to the next entry.
- Check that the PIDB is equipped. If it is not equipped, the audit proceeds to the next entry.
- Check the PIDB type against the PIDB type in the corresponding tuple of rCKTDATA. If there is an inconsistency, and error is reported and the PCispdbts array is updated to agree with rCKTDATA.
- Check the system PIDB (syspidb) in the PCispdbts[][] array against the system PIDB (pidb) in the corresponding tuple of rCKTDATA. If there is an inconsistency, and error is reported and the PCispdbts array is updated to agree with rCKTDATA.
- Check the associated PICB (asspicb) in the PCispdbts[][] array if this PIDB is associated with a RISLU.

The associated PICB should be the same as the associated PICB (assoc) field in the rLISLUPIDB tuple. If the unit is an ISLU or DCLU-5, then the associated PICB (ascpicb) field in PCispdbts[][] should be 0x7f. If inconsistencies are found, the ascpicb field is recovered to either 0x7f or to the corresponding tuple of RLISLUPIDB.

- (f) Access rSMEST and verify the qualifier and base state fields. The key of rSMEST (cktname) is formed as follows:



Figure 2 rSMEST Key

- (g) If the maintenance state is in-service, the audit checks the entry against rIPTSB. The busy/idle state of the PCispdbts should match the busy/idle status of the rIPTSB. If inconsistencies are found, PCispdbts will be updated to agree with the rIPTSB state.
- (h) Check the clear channel (clrchan) in the PCispdbts array. ISLUs and DCLUs always have 64kb/s clear channels and thus clrchan should indicate this. RISLUs either contain 64kb/s clear channels or 56kb/s non-clear channels. For RISLUs the clrchan field is audited against data in rCKTDATA.

NOTE: In each of the above instances, errors are reported when inconsistencies arise.

4. ERROR HANDLING

The PCpidbts array is changed to agree with the data in RLCKTDATA, RLSMEST, RLPTSB as appropriate.

The PTSB and CKTDATA audits are scheduled as related audits.

If an error is found in the active/standby state, the time slot use is corrected. Otherwise, the map is idled for the load shared configuration. The protocol handler (PH) status and member status will be corrected to match the values in rIODPSOP.

The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The maintenance state in PCispdbts[][] is not OOS when rCKTDATA tuple is not found.
Possible Error Effect:	Call processing may try to use an unequipped ISLU PIDB time slot.
Corrective Action Taken:	The maintenance state in PCispdbts[][] will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispdbts[][].
<i>Bad Data =</i>	The contents of maintenance state.
<i>Good Data =</i>	The corrected maintenance state.
<i>Logical Index =</i>	The circuit name of rCKTDATA tuple.
<i>Dump =</i>	The PCispdbts[][].

Table 2 Error Code: ASSOC_PTSB

Error Code:	ASSOC_PTSB
Error Description:	The PTSB cduse was found to be inconsistent with the PCispdbts[][] tsuse.

Possible Error Effect:	The CD side in use is not known to Call Processing, since it is not correctly reflected in dynamic data.
Corrective Action Taken:	If the PTSB cduse indicates ACTIVE/STANDBY the PCispidbts[][] tsuses for both sides are idled out. If the PTSB cduse is LOAD SHARED or some state other than the legal states previously mentioned, the rIPTSB tuple is quarantined, and the PCispidbts[][] time slot for side 0 is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCispidbts[] array entry found in error.
<i>Bad Data =</i>	The cduse field from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key (ptsb) of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The PCispidbts[] array entry found in error.

Table 3 Error Code: BAD_PTSB

Error Code:	BAD_PTSB
Error Description:	The Peripheral Interface Data Bus (PIDB) associated with this PCispidbts[] array entry is in an active/standby mode but the PIDB type (type) field is inconsistent between the PCispidbts[] array entry for side 0 versus side 1.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PIDB type (type) field in the PCispidbts[] array entry associated with the standby side is updated from the active side. The PIDB type (type) field of the active side has already been audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCispidbts[] array entry found in error.
<i>Bad Data =</i>	The PIDB type (type) field from the PCispidbts[] array entry associated with the standby side.
<i>Good Data =</i>	The PIDB type (type) field from the PCispidbts[] array entry associated with the active side.
<i>Logical Index =</i>	The PIDB circuit name.
<i>Dump =</i>	NA.

Table 4 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The type field in the PCispidbts[][] map for an unequipped Integrated Services Line Unit (ISLU) Peripheral Interface Data Bus (PIDB) is not set to PCNULLPIDB.
Possible Error Effect:	Can cause asserts from Switch Maintenance (SMIM) when restoring an ISLU Common Data (CD) circuit to in-service.
Corrective Action Taken:	The type field in the PCispidbts[][] map will be recovered to PCNULLPIDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCispidbts[][] array.
<i>Bad Data =</i>	The circuit name of the corresponding ISLU PIDB.
<i>Good Data =</i>	(0) PCNULLPIDB.
<i>Logical Index =</i>	The circuit name of the corresponding ISLU PIDB.
<i>Dump =</i>	The PCispidbts[][] global array.

Table 5 Error Code: BUSMATCH

Error Code:	BUSMATCH
Error Description:	The standby PIDB time slot usage is inconsistent with active PIDB.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The standby PIDB time slot usage will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispidbts[][].
<i>Bad Data =</i>	The contents of standby PIDB time slot usage.

<i>Good Data</i> =	The corrects of active PIDB time slot usage.
<i>Logical Index</i> =	PIDB circuit name.
<i>Dump</i> =	NA.

Table 6 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The state field (usage) in the PCispidbts[] array for this time slot is marked busy but the state field (use) in the corresponding rIPTSB tuple indicates this time slot is idle.
Possible Error Effect:	Call Processing would be accessing invalid data and asserting.
Corrective Action Taken:	The state field (usage) in the PCispidbts[] array for this time slot is marked idle to match the state of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCispidbts[] array entry found in error.
<i>Bad Data</i> =	The state field (usage) from the PCispidbts[] array.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple.
<i>Dump 1</i> =	The rIPTSB tuple.
<i>Dump 2</i> =	The PCispidbts[] array entry found in error.

Table 7 Error Code: CDSEC

Error Code:	CDSEC
Error Description:	The Peripheral Interface Data Bus (PIDB) associated with this PCispidbts[] array entry is in an active/standby mode but the associated PICB (ascpicb) field is inconsistent between the PCispidbts[] array entry for side 0 versus side 1.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The associated PICB (ascpicb) field in the PCispidbts[] array entry associated with the standby side is updated from the active side. The associated Peripheral Interface Control Bus (PICB) field (ascpicb) for the active side has already been audited.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCispidbts[] array entry found in error.
<i>Bad Data</i> =	The associated PICB (ascpicb) field from the PCispidbts[] array entry associated with the standby side.
<i>Good Data</i> =	The associated PICB (ascpicb) field from the PCispidbts[] array entry associated with the active side.
<i>Logical Index</i> =	The PIDB circuit name.
<i>Dump</i> =	NA.

Table 8 Error Code: CLRCH

Error Code:	CLRCH
Error Description:	The data in the clrchan field of PCispidbts[][] is incorrect.
Possible Error Effect:	Customer service is degraded.
Corrective Action Taken:	Clrchan is rewritten to agree with data from RLCKTDATA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of PCispidbts[][].
<i>Bad Data</i> =	The contents of PCispidbts[][].clrchan[].
<i>Good Data</i> =	The correct value of PCispidbts[][].clrchan[].
<i>Logical Index</i> =	LU, PIDB, CD.
<i>Dump</i> =	PCispidbts[][].

Table 9 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The DPIDB mask stored in the PCispidbts[][] global array, which indicated that this time slot for a DPIDB is assigned a nailed up connection, did not match the one stored in static data (rIDPIDB_PT).
Possible Error Effect:	Call Processing would be accessing invalid data and asserting.

Corrective Action Taken:	The mask in PCispdbts[][] is corrected to match the one in static data (rIDPIDB_PT).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCispdbts[][] global array.
<i>Bad Data =</i>	The mask found in PCispdbts[][] .
<i>Good Data =</i>	The mask found in static data (rIDPIDB_PT).
<i>Logical Index =</i>	The key of the rIDPIDB_PT static tuple.
<i>Dump 1 =</i>	The PCispdbts[][] global array.
<i>Dump 2 =</i>	The rIDPIDB_PT static tuple.

Table 10 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The state field (usage) in the PCispdbts[] array for this time slot is marked idle but the state field (use) in the corresponding rIPTSB tuple indicates this time slot is busy.
Possible Error Effect:	Call Processing would be accessing invalid data and asserting.
Corrective Action Taken:	The state field (usage) in the PCispdbts[] array for this time slot is marked busy to match the state of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCispdbts[] array entry found in error.
<i>Bad Data =</i>	The state field (usage) from the PCispdbts[] array.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple.
<i>Dump 1 =</i>	The PCispdbts[] array entry found in error.
<i>Dump 2 =</i>	The rIPTSB tuple.

Table 11 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The maintenance state in PCispdbts[][] is not correct.
Possible Error Effect:	If the maintenance status (bas_stat = SMEST) in rSMEST is active, the PIDB is lost to the system. If the maintenance status is out-of-service (bas_stat = SMEST), then call processing may try to use an out-of-service PIDB time slot, which may lead to a peripheral interrupt.
Corrective Action Taken:	The maintenance state in PCispdbts[][] will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispdbts[][] .
<i>Bad Data =</i>	The contents of maintenance state.
<i>Good Data =</i>	The corrected maintenance state.
<i>Logical Index =</i>	The circuit name of rCKTDATA tuple.
<i>Dump 1 =</i>	The rCKTDATA tuple or rSMEST tuple.
<i>Dump 2 =</i>	The PCispdbts[][] .

Table 12 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit could not read the rIDPIDB_PT static tuple.
Possible Error Effect:	Asserts from Call Processing.
Corrective Action Taken:	None.
Manual Action Required:	There is an inconsistency in the static data since there should be a corresponding rIDPIDB_PT tuple for the rCKTDATA tuple previously read, this error will fire until the missing rIDPIDB_PT tuple is inserted by office personnel, which has the same key as the rCKTDATA tuple dumped by this error.
Dump Description:	
<i>Error Address =</i>	The address of the PCispdbts[][] global array.
<i>Bad Data =</i>	The key of the rCKTDATA tuple for which there was no corresponding rIDPIDB_PT tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rCKTDATA static tuple.
<i>Dump 1 =</i>	The PCispdbts[][] global array.
<i>Dump 2 =</i>	The rCKTDATA static tuple.

Table 13 Error Code: PIDB

Error Code:	PIDB
Error Description:	The static rISLUPIDB tuple associated with the rIPTSB tuple under audit has been found to be missing from the data base.
Possible Error Effect:	May cause asserts from application and maintenance programs.
Corrective Action Taken:	The audit will skip to the next PCispdbts[] array entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key (picb) of the rISLUPIDB tuple.
<i>Dump =</i>	The PCispdbts[] array entry.

Table 14 Error Code: PIDBTS

Error Code:	PIDBTS
Error Description:	The associated Peripheral Interface Control Bus (PICB) field (ascpicb) in the PCispdbts[] array is not set to 0x7F for an Integrated Services Line Unit (ISLU) Peripheral Interface Data Bus (PIDB) or set to the associated PICB field (assoc) from the static rISLUPIDB field for a RISLU PIDB.
Possible Error Effect:	May cause asserts from application and maintenance programs.
Corrective Action Taken:	The associated PICB field (ascpicb) in the PCispdbts[] array will be set to 0x7F for an ISLU PIDB or set to the associated PICB field (assoc) from the static rISLUPIDB field for a RISLU PIDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCispdbts[] array entry found in error.
<i>Bad Data =</i>	The associated PICB field (ascpicb) from the PCispdbts[] array.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The circuit name of the PIDB associated with this PCispdbts[] array entry.
<i>Dump =</i>	The PCispdbts[] array entry.

Table 15 Error Code: SMEST

Error Code:	SMEST
Error Description:	The key generated from PClusname[] for rISMEST is not correct.
Possible Error Effect:	The PIDB will be skipped in ISPIDBTS audit.
Corrective Action Taken:	Related audits will be scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispdbts[][].
<i>Bad Data =</i>	The contents of maintenance state.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rISMEST key.
<i>Dump =</i>	The PCispdbts[][].

Table 16 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The PIDB has an invalid use type.
Possible Error Effect:	All time slots on this PIDB are lost to the system.
Corrective Action Taken:	The PIDB time slot usage in PCispdbts[] will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispdbts[][].
<i>Bad Data =</i>	The contents of PIDB time slot usage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	PIDB circuit name.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The PCispdbts[][].

Table 17 Error Code: SYSPIDB

Error Code:	SYSPIDB
Error Description:	The system PIDB number in PCispidbts[][] is not correct.
Possible Error Effect:	May cause peripheral control to set up a path on a wrong PIDB which will cause parity errors and the call to fail.
Corrective Action Taken:	The system PIDB number in PCispidbts[][] will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispidbts[][].
<i>Bad Data =</i>	The contents of system PIDB number.
<i>Good Data =</i>	The corrected system PIDB number.
<i>Logical Index =</i>	The circuit name of rCKTDATA tuple.
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The PCispidbts[][].

Table 18 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	The PIDB time slots usage in PCispidbts[] is not correct.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PIDB time slot usage in PCispidbts[] will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispidbts[][].
<i>Bad Data =</i>	The contents of PIDB time slot usage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	PIDB circuit name.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The PCispidbts[][].

Table 19 Error Code: TSMAP2

Error Code:	TSMAP2
Error Description:	The state field (usage) in the PCispidbts[] array for this time slot is marked idle but the state field (use) in the corresponding rIPTSB tuple indicates this time slot is busy.
Possible Error Effect:	Call Processing would be accessing invalid data and asserting.
Corrective Action Taken:	The state field (usage) in the PCispidbts[] array for this time slot is marked busy to match the state of the corresponding rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCispidbts[] array entry found in error.
<i>Bad Data =</i>	The state field (usage) from the PCispidbts[] array.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The PCispidbts[] array entry found in error.

Table 20 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PIDB type in PCispidbts[][] is not correct.
Possible Error Effect:	All time slots on this PIDB are lost to the system.
Corrective Action Taken:	The PIDB type in PCispidbts[][] will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of PCispidbts[][].
<i>Bad Data =</i>	The contents of PIDB type.
<i>Good Data =</i>	The corrected PIDB type.
<i>Logical Index =</i>	The circuit name of rCKTDATA tuple.
<i>Dump 1 =</i>	The rCKTDATA tuple.
<i>Dump 2 =</i>	The PCispidbts[][].

Table 21 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was invoked after taking a system reset.

Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	The audit has been returned to audit control and will be rescheduled for recovery activities.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Line unit number.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Time slot number.
<i>Dump =</i>	NA.

ISTERM

Software Release: 5E10 and later

1. TERMINAL (ISTERM) AUDIT

The static relation RLIS_TERM contains digital subscriber line terminal feature data. There is a field in this relation which is used to determine whether tuples in the dynamic relation RLATTM_BLF should be populated. The new switching module (SM) audit, ISTERM, has been introduced to detect, report, and correct data inconsistencies found between this static relation and the associated dynamic relation.

The static relation RLIS_TERM is being used to check whether tuples in dynamic relation RLATTM_BLF should be populated as well as to do some semantic data checking. If the ISTERM audit encounters any errors, the ATTM_BLF audit will be run as a related audit at elevated mode.

2. DATA STRUCTURES

For details on the fields contained in the RLIS_TERM relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

The ISTERM audit performs the following functions:

- (a) The ISTERM audit will invoke the AUdrstatic driver to step through the tuples of RLIS_TERM. No error checking will be performed by the driver. (It is used only to step through the static relation.) The static relation RLIS_TERM is being used to check whether tuples in dynamic relation RLATTM_BLF should be populated as well as to do some semantic data checking.
- (b) Performs static to dynamic checks between static RLIS_TERM and dynamic RLATTM_BLF.

4. ERROR HANDLING

If the static relation RLIS_TERM indicates that a tuple in RLATTM_BLF should be populated and it is not, an error will be reported and function RTattmblf() will be called. This function, given a latidx, will insert, link to the appropriate idle list, and initialize the semantic RLATTM_BLF tuple.

If the static relation RLIS_TERM latidx field does not match that found in RLATTM_BLF, an error will be reported and the RLATTM_BLF latidx will be corrected to match that in RLIS_TERM.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: HEAD

Error Code:	HEAD
Error Description:	The audit is trying to link an RLATTM_BLF tuple back to its idle list but is finding that the RLATTH_BLF tuple that serves as its idlelist headcell is in error. The ATTM_BLF will not link tuples back to the idlelist as long as the RLATTH_BLF tuple serving as the idlelist headcell is in error. This could result in numerous ATTM_BLF audit errors until the idlelist headcell is recovered.
Possible Error Effect:	An available RLATTM_BLF tuple may never be used to reflect attendant console information.

Corrective Action Taken:	The audit will schedule the ATTHBLF audit to recover the idlelist head cell in error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Local pointer to buffer to hold idlelist headcell tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	latidx field of the related ODD rIIS_TERM tuple.
<i>Dump =</i>	The static data rIISTERM tuple.

Table 2 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The audit is trying to link an rIATTM_BLF tuple back to it's idle list but is finding that the ODD variable Gllatiablf is indicating that no rIATTH_BLF tuples should exist to serve as the idlelist headcell. There is an inconsistency between the ODD rIIS_TERM which has a latidx > 1 indicating that rIATTM_BLF tuples should exist and the ODD variable Gllatiablf that says no rIATTH_BLF tuples should exist. This could result in numerous ATTMBLF and ISTERM audit errors until the ODD is made consistent.
Possible Error Effect:	Attendant console operation will be impaired. (that is, wrong statuses will show up on the console: attendant console call processing may not work.)
Corrective Action Taken:	The ODD needs to be made consistent. The bad data in the audit dump is the value found in Gllatiablf while the good data indicates the value of the latidx field of a rIIS_TERM tuple. You can not have Gllatiablf set to less than 1 while rIIS_TERM has latidx fields set to > 0.
Manual Action Required:	Correct the ODD to make Gllatiablf and rIIS_TERM consistent. Check other attendant console ODD for consistency.
Dump Description:	
<i>Error Address =</i>	Local pointer to rIATTM_BLF tuple.
<i>Bad Data =</i>	ODD Gllatiablf.
<i>Good Data =</i>	ODD rIIS_TERM latidx field.
<i>Logical Index =</i>	latidx field of the rIATTM_BLF tuple, may be -1 if no rIATTM_BLF tuple exists when it should exist.
<i>Dump =</i>	The static data rIISTERM tuple.

Table 3 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The audit is trying to get a rIATTM_BLF tuple inserted, initialized and linked to the appropriate idle list. It seems that the function to do this has failed. The failure could be due to inconsistent ODD between rIIS_TERM latidx fields and Gllatiablf ODD variable or it could be the result of data base primitives failure to insert tuples; or parameters have been garbaged during the calling of the function.
Possible Error Effect:	Attendant console operation will be impaired. (that is, wrong statuses will show up on the console; attendant console call processing may not work.)
Corrective Action Taken:	The audit will try and get the data corrected the next time it is called. If this error continues check the ODD and the action of the data base primitives.
Manual Action Required:	If the error continues check the ODD to make sure Gllatiablf and rIIS_TERM are consistent.
Dump Description:	
<i>Error Address =</i>	Not an address; latidx of the related rIIS_TERM tuple
<i>Bad Data =</i>	Return value from the RTA function that does the insert, initialize and link. 0 = GLSUCCESS, everything is ok. -1 = GLFAIL, Gllatiablf is < 1. -800 = RTA_FAIL, rIIS_TERM latidx is < 1. Any other value is the return value from database primitives used to insert and link the rIATTM_BLF tuple.

<i>Good Data</i> =	ODD variable Gllatiablf.
<i>Logical Index</i> =	latidx field of the rIATTM_BLF tuple, may be -1 if no rIATTM_BLF tuple exists when it should exist.
<i>Dump</i> =	The static data rIISTERM tuple.

Table 4 Error Code: KEY

Error Code:	KEY
Error Description:	The audit has detected an inconsistency between the static data in rIISTERM and the dynamic data in rIATTM_BLF. The inconsistency indicate that the static data latidx field does not match the key field of the corresponding dynamic rIATTM_BLF tuple.
Possible Error Effect:	An available rIATTM_BLF tuple may never be used to reflect attendant console information.
Corrective Action Taken:	The audit will correct the key field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rIATTM_BLF tuple.
<i>Bad Data</i> =	latidx field (key) of rIATTM_BLF tuple being checked.
<i>Good Data</i> =	latidx field of rIISTERM (static data) tuple.
<i>Logical Index</i> =	The lport (logical port) field of the rIISTERM tuple.
<i>Dump</i> =	The dynamic data rIATTM_BLF tuple; the static data rIISTERM tuple.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit has detected an inconsistency between the static data in rIISTERM and the dynamic data in rIATTM_BLF. The inconsistency indicate that the static data has an entry in it's latidx field and yet no corresponding dynamic rIATTM_BLF tuple exists.
Possible Error Effect:	Not enough rIATTM_BLF tuples may be available to reflect attendant console information.
Corrective Action Taken:	The audit will insert a rIATTM_BLF tuple, get it initialized as well as linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	latidx field of rIISTERM (static data) tuple.
<i>Logical Index</i> =	The key of the rIATTM_BLF tuple that is missing.
<i>Dump</i> =	The static data rIISTERM tuple.

Table 6 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit is trying to link an rIATTM_BLF tuple back to it's idle list but is finding that the rIATTH_BLF tuple that serves as its idllelist headcell does not exist. The reason it does not exist could be due to inconsistent ODD related to the global variable Gllatiablf and the RLIS_TERM relation. If the Gllatiablf indicates that no rIATTH_BLF tuples should exist, (that is, no ISDN attendant consoles) then the ODD rIIS_TERM relation should likewise reflect that no attendant consoles exist.
Possible Error Effect:	An available rIATTM_BLF tuple may never be used to reflect attendant console information.
Corrective Action Taken:	The audit will insert a rIATTM_BLF tuple, get it initialized as well as linked to the appropriate idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Local pointer to buffer to hold idllelist headcell tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	latidx field of the related ODD rIIS_TERM tuple.
<i>Dump</i> =	The static data rIISTERM tuple.

ISTF

Software Release: 5E10 and later

1. INTEGRATED SERVICES TEST FUNCTION (ISTF) AUDIT

The integrated services test function device measures the error rate of digital transmission devices such as digital subscriber lines (DSLs) and digital trunks. To ensure that the device is performing its full range of checks, it is audited periodically by the ISTF audit. Recovery on the device is performed only on the entire device.

The ISTF audit is part of the terminal maintenance audits subgroup.

2. DATA STRUCTURES

The ISTF audit checks three arrays and the integrated service test function relation (RLISTF): the active services count array (PCistfas), the busy/idle array (PCistf_chan), and the message pending queue array (PCistf_mpq). These data structures are shown in Figure 1.

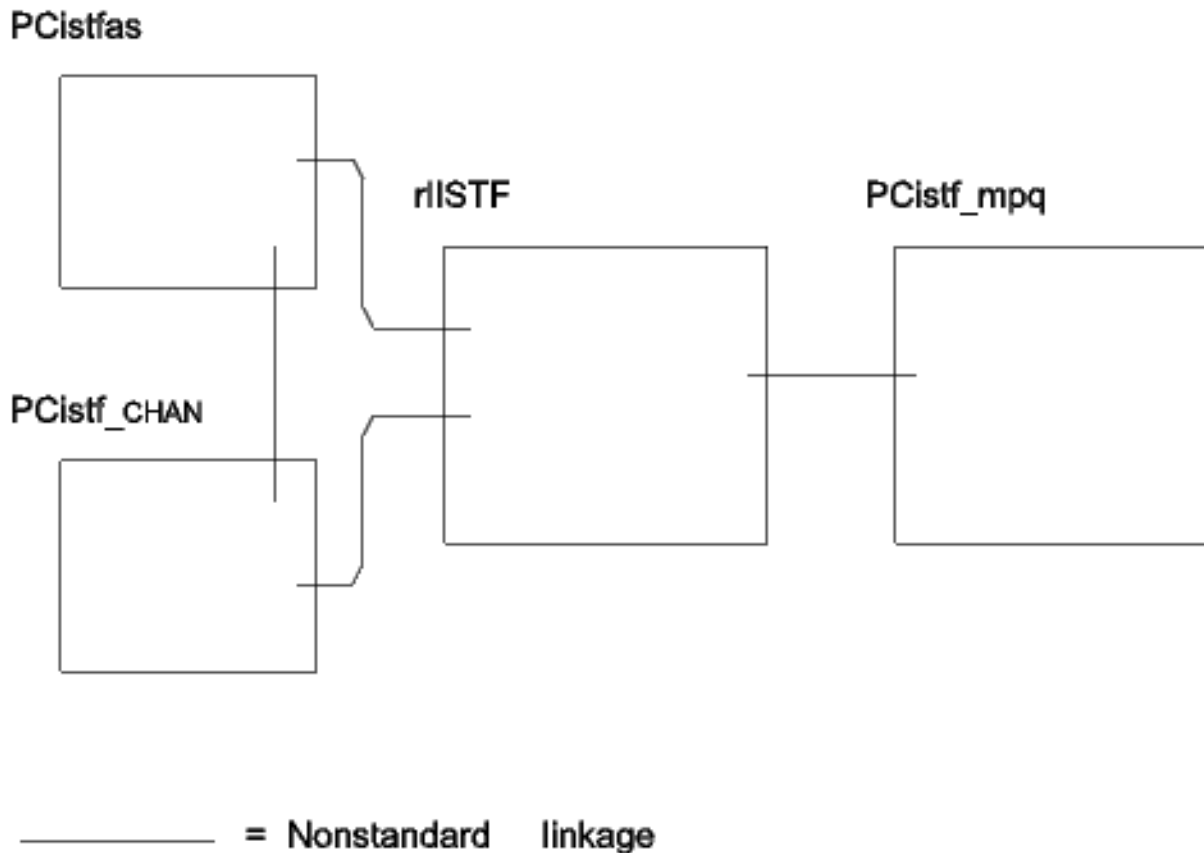


Figure 1 (ISTF-1) Data Structures of the ISTF Audit

3. AUDIT CHECKS

The audit makes the following checks.

- The audit verifies the entries of PCistfas against the entries in PCistf_chan. If there are no rlISTF tuples, the PCistfas and PCistf_chan arrays should have null entries.
- The audit verifies that the entries in PCistf_chan and PCistfas are set to idle when an ISTF unit is grown into

the switch.

- (c) The audit checks the PCistf_mpq array when blockage occurs in sending messages to the test function device.

4. ERROR HANDLING

The audit initializes the array entries when it detects errors.

The ISTF audit schedules no related audits when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	The rSMEST tuple for the common board indicates that the ISTF unit is out of service; however, the active service array entry PCistfas[unit] does not agree. The rSMEST tuple is the master data.
Possible Error Effect:	Programs attempting to use this OOS unit will be connected to an OOS unit and will not work.
Corrective Action Taken:	The audit will set the active services array entry to OOS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of active service entry - PCistfas[unit].
<i>Bad Data =</i>	Value in active service entry.
<i>Good Data =</i>	Value of PCOOS_ISTF.
<i>Logical Index =</i>	ISTF unit number.
<i>Dump =</i>	rSMEST tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of busy states in the busy/idle array PCistf_chan[unit][chan] do not add up to the active services array entry count in PCistfas[unit].
Possible Error Effect:	The active services array entry may incorrectly assign channels in its fair share scheme.
Corrective Action Taken:	The ISTF unit and associated data structures are idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of active service entry - PCistfas[unit].
<i>Bad Data =</i>	Value of active service entry.
<i>Good Data =</i>	Actual count from adding up entries in PCistf_chan[][].
<i>Logical Index =</i>	ISTF unit number.
<i>Dump =</i>	PCistf_chan[unit][].

Table 3 Error Code: NULLCNTL

Error Code:	NULLCNTL
Error Description:	The active services array entry PCistfas[unit] indicates that the unit does not exist even though the rISTF tuple does exist.
Possible Error Effect:	Programs will never be allocated this unit even though it is active.
Corrective Action Taken:	The active services array entry PCistfas[unit] is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of active service entry - PCistfas[unit].
<i>Bad Data =</i>	Value in active service entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ISTF unit number.

<i>Dump</i> =	riSMEST tuple.

Table 4 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The message pending queue has the dead pid.
Possible Error Effect:	The dead pid occupied the message pending queue cause the resource lost.
Corrective Action Taken:	The dead pid has been reported and the message pending queue has been reshuffled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Message pending queue dead pid address.
<i>Bad Data</i> =	Message pending queue dead pid.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ISTF unit number.
<i>Dump</i> =	NA.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The message pending queue pointers PCmpqFptr[unit] and PCmprRptr[unit] have values outside the range (size) of the circular queue.
Possible Error Effect:	Wild writes can occur if the pointer is used to insert a process ID.
Corrective Action Taken:	The pointers are reset to the beginning of the queue and the queue is emptied.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Message pending queue front pointer PCmpqFptr[unit].
<i>Bad Data</i> =	Front pointer (high byte) - rear pointer (low byte).
<i>Good Data</i> =	Value which pointer should be within - PCMAXCHANISTF.
<i>Logical Index</i> =	ISTF unit number.
<i>Dump</i> =	NA.

Table 6 Error Code: OWNER

Error Code:	OWNER
Error Description:	An entry in the busy/idle array PCistf_chan[unit][chan] had a valid busy state, but there was no owning process traceable through the rICDBCOM and rICCB COM tuples.
Possible Error Effect:	The channel will not be allocated for use.
Corrective Action Taken:	The busy/idle array PCistf_chan[unit] and active services array entry PCistfas[unit] are idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rICCB COM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rICDB COM.
<i>Dump</i> =	NA.

Table 7 Error Code: SMEST

Error Code:	SMEST
Error Description:	A riISTF tuple exists, but the corresponding riSMEST tuple does not exist and the active services array entry PCistfas[unit] is not null.
Possible Error Effect:	Programs may attempt to use this unit, but they would eventually fail.
Corrective Action Taken:	Set the active services array entry PCistfas[unit] to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of active service entry - PCistfas[unit].
<i>Bad Data</i> =	Value in active service entry.
<i>Good Data</i> =	Value of PCNULL ISTF.
<i>Logical Index</i> =	ISTF unit number.
<i>Dump</i> =	NA.

Table 8 Error Code: STATE

Error Code:	STATE
Error Description:	The rICCBCom Ipstate attribute did not have a matching state to the busy/idle array entry PCistf_chan[unit][chan].
Possible Error Effect:	The rICCBCom tuple may have been assigned to another channel with a different service.
Corrective Action Taken:	The quarantine bit in rICCBCom is set, which causes this tuple to be received in a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rICCBCom tuple.
<i>Bad Data</i> =	Ipstate attribute in rICCBCom tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rICDBCom tuple.
<i>Dump 1</i> =	PCistf_chan[unit][chan].
<i>Dump 2</i> =	rICCBCom tuple.

Table 9 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	An entry in the busy/idle array PCistf_chan[unit][chan], did not have a valid state.
Possible Error Effect:	The corresponding channel may be avoided by the allocation program.
Corrective Action Taken:	The busy/idle array PCistf_chan[unit] and active services array PCistfas[unit] entry are idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of service entry for channel - PCistf_chan[unit][chan].
<i>Bad Data</i> =	Value in service entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ISTF unit number (high byte) - channel number (low byte).
<i>Dump</i> =	NA.

Table 10 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The active services array element PCistfas[unit] was non-null, even though the rLISTF tuple does not exist.
Possible Error Effect:	A channel may be allocated to a unit which does not exist.
Corrective Action Taken:	The active services array entry PCistfas[unit] is nulled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of active service entry - PCistfas[unit].
<i>Bad Data</i> =	Value in active service entry.
<i>Good Data</i> =	Value of PCNULL_ISTF.
<i>Logical Index</i> =	ISTF unit number.
<i>Dump</i> =	NA.

JBLH

Software Release: 5E10 and later

1. JOB LIST HEAD CELL (JBLH) AUDIT

Call processing work, which is done primarily by terminal processes, falls into two categories: background (high-level processing) and foreground (low-level processing). Background processing involves customer features; foreground processing manipulates hardware. Background processes request foreground process work. If the foreground process is busy, the channel control block that represents the background process is buffered on a job list head cell. There is one job list head cell for each tuple of foreground work requested.

The job list head cell audit (JBLH) verifies the consistency of the job list head cell relation (RLJBLH). This audit is part of the peripheral control audits subgroup.

2. DATA STRUCTURES

The JBLH audit operates on the job list head cell relation (RLJBLH) and its owner linkage with the channel control block common area relation (RLCCBCOM). Figure 1 illustrates the rJBLH linkages.

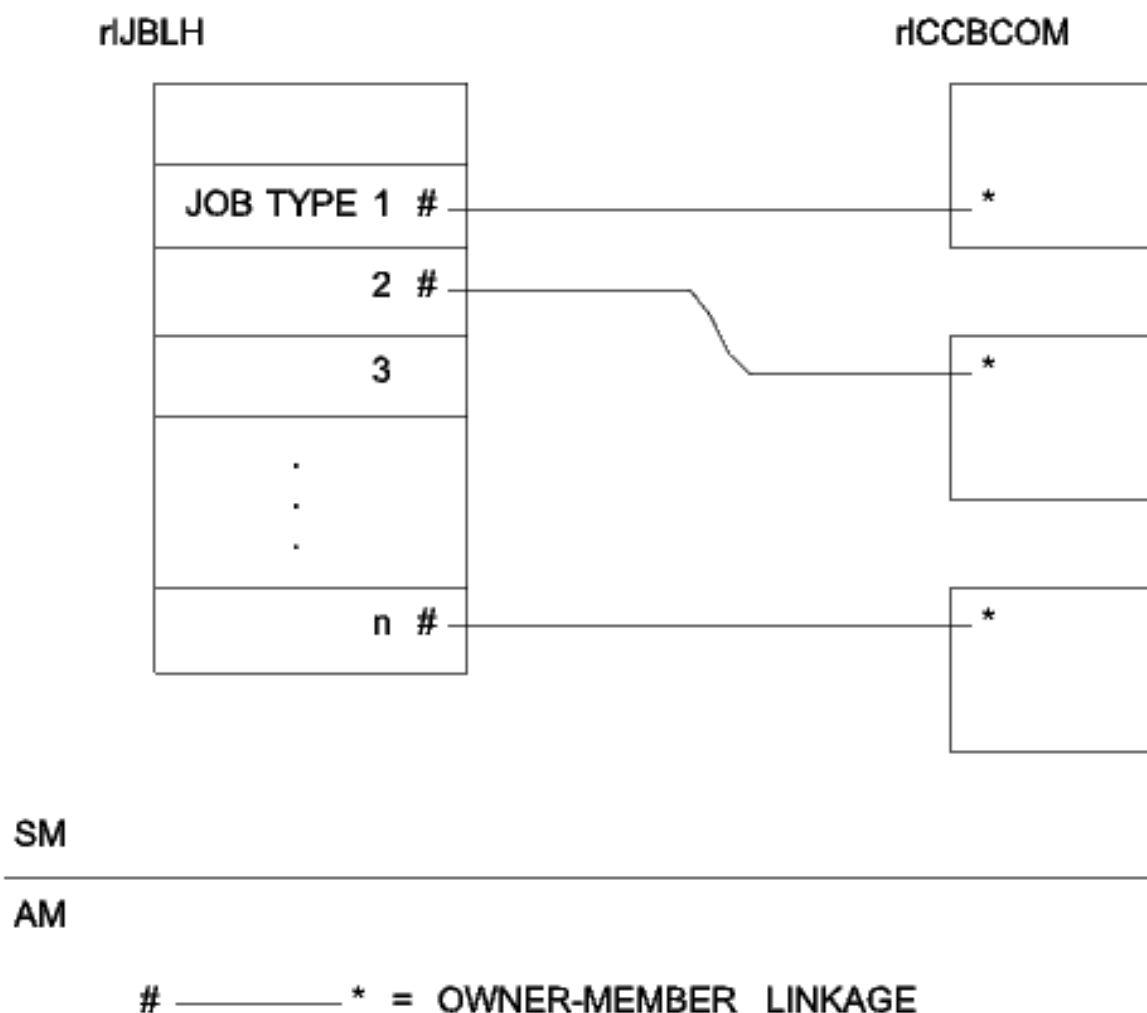


Figure 1 (JBLH-1) rJBLH Linkages

3. AUDIT CHECKS

To verify the linkages associated with the job list head cells, the JBLH audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the driver invokes the JBLH semantic check function to perform the following checks:

- (a) The number in the total field of the semantic data should equal a count of channel control block common areas linked to that job list head cell; if it does not, an assert will occur.
- (b) When the audit counts channel control blocks linked to a job list head cell, it traverses channel control block linkages and performs a point-to-point-back check for the channel control blocks. If a check fails, the audit prints an error message.

Any tuples that fail the checks are reinitialized and placed on the idle list.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	A count of rCCBCOM tuple's linked to a rJBLH tuple does not equal the total field in the rJBLH tuple's semantic data. This total field is the count of how many rCCBCOM tuple's are linked to a particular rJBLH tuple.
Possible Error Effect:	There is a strong possibility of asserts occurring.
Corrective Action Taken:	The rJBLH tuple is zeroed and rCCBCOM tuple's linked to the rJBLH tuple are corrected when the CCBCOM audit is run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rJBLH tuple address.
<i>Bad Data =</i>	The total field of the rJBLH tuple semantic data.
<i>Good Data =</i>	The count of rCCBCOM tuple's linked to the rJBLH tuple.
<i>Logical Index =</i>	Logical key of rJBLH tuple.
<i>Dump =</i>	rJBLH tuple.

Table 2 Error Code: SLAVEMAP

Error Code:	SLAVEMAP
Error Description:	The rCCBCOM point-to-point-back check failed.
Possible Error Effect:	The list is broken.
Corrective Action Taken:	The rJBLH tuple is zeroed.
Manual Action Required:	Request the CCBCOM audit if it has not already run.
Dump Description:	
<i>Error Address =</i>	rCCBCOM address.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rCCBCOM.
<i>Dump =</i>	rCCBCOM tuple.

18. K - LC

L2TL

Software Release: 5E10 and later

1. LEVEL 2 TIMING LIST (L2TL) AUDIT

The L2TL audit is part of the switch maintenance audits subgroup. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The L2TL audit uses the packet switching level 2 timing list head cell array (PStlists).

3. AUDIT CHECKS

The audit makes linkage checks within the PShists array.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	An error was found in the owner field of either the first or last field of the LLCB linked headcell.
Possible Error Effect:	Creates an error dump.
Corrective Action Taken:	Returns AUFALSE which reinitializes the free linked list owned by the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	"owner" value of the first element of the headcell.
<i>Bad Data =</i>	"owner" value of the last element of the headcell.
<i>Good Data =</i>	Valid "owner" value expected (timing list, timeout list, LAPB free linked list, LAPD free linked list).
<i>Logical Index =</i>	L2TL index.
<i>Dump =</i>	LLCB free linked headcell (PSRHEADCELL type).

Table 2 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The specified head cell's first or last element is out of range.
Possible Error Effect:	Prints an error report and continues.
Corrective Action Taken:	AUFALSE is returned and the head cell is nulled. The linked list owned by the head cell is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting range for the first/last head cell element.
<i>Bad Data =</i>	Ending range for the first/last head cell element.
<i>Good Data =</i>	Null value of the first and last.
<i>Logical Index =</i>	Head cell index.
<i>Dump =</i>	Head cell dump or PSRHEADCELL type: - unsigned short PSrllcbhd.first. - unsigned short PSrllcbhd.last.

Table 3 Error Code: HEAD

Error Code:	HEAD
Error Description:	Either 1) the level 3 timing headcell first's owner does not match the headcell value or 2) the first's ptimer is not null or 3) the level 3 timing headcell last's owner does not match the headcell value or 4) the last's ntimer is not null.
Possible Error Effect:	The audit reports the error and continues.
Corrective Action Taken:	The audit reports the error, corrects the level 3 timing headcell and rebuilds the timing linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	PST3NULL, null value that ntimer and ptimer are being compared against.
<i>Logical Index =</i>	Level 3 timing headcell number.
<i>Dump =</i>	AULNK (all sized as unsigned short): • "back" struct.'s owner field, • "back" struct.'s next field, • "back" elements back field, • current element being audited, • current element's owner, • current element's on this link list, • current element's back field, • data element null value, • head cell null value, • headcell's first element, • headcell's last element, • "next" element's owner field, • "next" element on this linked list, • "next" element's back field. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and is filled with 0xaa.

Table 4 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	An error was found in the LLCB timing or timeout headcell.
Possible Error Effect:	Create an error dump.
Corrective Action Taken:	Returns AUFALSE which reinitializes the linked list owned by the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	AUcount - the count indicating which headcell.
	4 = LAPD linked list headcell.

	5 = LAPB linked list headcell.
<i>Good Data</i> =	Linked list null value.
<i>Logical Index</i> =	Linked list owner.
<i>Dump</i> =	AULNK type as defined in au/AUps/AUextern.h (all sized as unsigned short): • "back" struct.'s owner field, • "back" struct.'s next field, • "back" elements back field, • current element being audited, • current element's owner, • current element on this linked list, • current element's back field, • head cell null value, • headcell's first element, • headcell's last element, • data element null value, • next element's owner field, • "next" element on this linked list, • next element's back field. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and is filled with 0xaa.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, the audit ID that was running at the time of the interrupt.
<i>Dump</i> =	NA.

L3TE

Software Release: 5E10 and later

1. LEVEL 3 TIMING ELEMENT (L3TE) AUDIT

The L3TE audit is part of the switch maintenance audits subgroup. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The L3TE audit uses the packet switching level 3 timing element array (PSt3elem) and the packet switching level 3 timing list head cell array (PSt3head).

3. AUDIT CHECKS

The audit makes linkage checks within the PSt3elem array and with the PSt3head array.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The owner field of the level 3 timing element is out of range.
Possible Error Effect:	Prints an error report.
Corrective Action Taken:	Returns AUFALSE and the level 3 timing element is nulled out and the owner set to PST3NULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Lowest owner range value.
<i>Bad Data =</i>	Bad "owner" value.
<i>Good Data =</i>	Highest owner range value.
<i>Logical Index =</i>	Level 3 timing element number.
<i>Dump =</i>	NA.

Table 2 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	For a level 3 timing element on NO LINKED LIST, either the next timer field, the previous timer field or the "owner" field is not a null value.
Possible Error Effect:	Prints an error report and corrects the appropriate field(s).
Corrective Action Taken:	The next timer field, the previous timer field and the list_owner fields are all nulled properly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The previous timer value.
<i>Bad Data =</i>	The next timer value.
<i>Good Data =</i>	PST3NULL - valid null value for timer fields.
<i>Logical Index =</i>	Level 3 timing element number.
<i>Dump =</i>	cap_owner (PSCAPTYPE), the owner of the level 3 timing element.

Table 3 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit found a member error in the linked list of the specified audit.
Possible Error Effect:	Prints an error report and corrects the linked list.
Corrective Action Taken:	Return AUFALSE to calling function which rebuilds the linked list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Linked list null value.
<i>Logical Index =</i>	AUcount, the current element number.
<i>Dump =</i>	AULNK defined in au/AUps/AUextern.h (all sized as unsigned short): • "back" struct.'s owner field, • "back" struct.'s next field, • "back" elements back field, • current element being audited, • current element's owner, • current element on this linked list, • current element's back field, • data element null value, • head cell null value, • headcell's first element, • headcell's last element, • "next" element's owner field, • "next" element on this linked list, • "next" element's back field.

Table 4 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The level 3 timing element's next timer field or previous timer field is out of range.
Possible Error Effect:	Prints an error report.
Corrective Action Taken:	The function returns AUFALSE and the timing linked list is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The previous timer value.
<i>Bad Data =</i>	The next timer value.
<i>Good Data =</i>	PST3ALLTE - Maximum timer value.
<i>Logical Index =</i>	Level 3 timing element number.
<i>Dump =</i>	NA.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, Audit id that was running at the time of the interrupt.
<i>Dump</i> =	NA.

L3TL

Software Release: 5E10 and later

1. LEVEL 3 TIMING LIST (L3TL) AUDIT

The L3TL audit is part of the switch maintenance audits subgroup.

2. DATA STRUCTURES

The L3TL audit uses the packet switching level 3 timing list head cell array (PSt3head) and the level 3 timing elements (PSt3elem).

3. AUDIT CHECKS

The audit makes linkage checks within the PSt3head array and with the packet switch level 3 timing list element array (PSt3elem). It invokes its semantic check function to verify the timer fields within the array.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The specified head cell's first or last elements are out of range.
Possible Error Effect:	Reports and error report and continues.
Corrective Action Taken:	AUFALSE is returned and the head cell is nulled and linked list owned by the head cell is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting range for the first/last head cell element.
<i>Bad Data =</i>	Ending range for the first/last head cell element.
<i>Good Data =</i>	Null value of the first and last.
<i>Logical Index =</i>	Head cell index.
<i>Dump =</i>	Head cell dump or PSRHEADCELL type.

Table 2 Error Code: HEAD

Error Code:	HEAD
Error Description:	Either (1) the level 3 timing head cell first's owner does not match the head cell value or (2) the first's ptimer is not null or (3) the level 3 timing head cell last's owner does not match the head cell value or (4) the last's ntimer is not null.
Possible Error Effect:	The audit will report the error and continue.
Corrective Action Taken:	The audit will report the error, correct the level 3 timing head cell and rebuild the timing linked list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PST3NULL, null value that ntimer and ptimer are being compared against.
<i>Logical Index</i> =	Level 3 timing head cell number.
<i>Dump</i> =	AULNK, as defined in au/AUps/AUextern.h (all sized as unsigned short): • "back" struct.'s owner field, • "back" struct.'s next field, • "back" elements back field, • current element being audited, • current element's owner, • current element on this linked list, • current element's back field, • data element null value, • head cell null value, • headcell's first element, • headcell's last element, • "next" element's owner field, • "next" element on this linked list, • "next" element's back field.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, Audit ID that was running at the time of the interrupt.
<i>Dump</i> =	NA.

LAESCASE

Software Release: 5E14 and later

1. LAES CASE DATA BLOCK (LAESCASE) AUDIT

A lawfully authorized electronic surveillance (LAES) case data block (PSlaescase[]) is used by packet switching to keep track of switching to keep track of call data channel (CDC) chaining and surveillance specific information.

2. DATA STRUCTURES

For details on the fields contained in the PSlaescase[] array, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	A PSlaescase[] array entry is not in use by a ALDB (PSlaescase[].aldb_index = PSALDBNULL) but it is not linked to the idle list.
Possible Error Effect:	The PSlaescase[] array entry is not available for use and may contribute to resource overload in the PH.
Corrective Action Taken:	The audit unlinks the PSlaescase[] array entry from any list it is in, sets all fields to default values, and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	First entry on the idle list (PSlaescasehd.first).
<i>Bad Data =</i>	Last entry on the idle list (PSlaescasehd.last).
<i>Good Data =</i>	PSaldb[] array index (PSlaescase[].aldb_index)
<i>Logical Index =</i>	Index into the PSlaescase[] array.
<i>Dump 1 =</i>	PSlaescase[] array entry.
<i>Dump 2 =</i>	The "forward" PSlaescase[] array entry (PSlaescase[].fd_laescase_idx) if the index is in range (< PSMXLAESCASE).
<i>Dump 3 =</i>	The "backward" PSlaescase[] array entry (PSlaescase[].fk_laescase_idx) if the index is in range (< PSMXLAESCASE).

Table 2 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The PSlaescase[] array idle list head cell (PSlaescasehd) is not empty (either <i>first</i> or <i>last</i> is != PSIAESCASENULL) but is not valid (either <i>first</i> or <i>last</i> is ≥ PSMXLAESCASE).
Possible Error Effect:	PSlaescase[] array entries can not be allocated and may contribute to resource overload in the PH.
Corrective Action Taken:	The audit rebuilds the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	First entry on the idle list (PSlaescasehd.first).
<i>Bad Data =</i>	Last entry on the idle list (PSlaescasehd.last).
<i>Good Data =</i>	Value of PSMXLAESCASE.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	The PSlaescase[] array idle list head cell (PSrIaescasehd) contains the index (in either <i>first</i> or <i>last</i>) of a non-idle PSlaescase[] array entry (PSlaescase[].aldb_index != PSALDBNULL) or the PSlaescase[] array entry is not at the end of list as expected (PClaescase[].bk_laescase_idx or PClaescase[].fd_laescase_idx != PSIAESCASENULL depending on which end of the list the PSlaescase[] array entry is supposed to be at).
Possible Error Effect:	Allocation of the PSlaescase[] array entries may fail or the same PSlaescase[] array entry may be allocated more than once.
Corrective Action Taken:	The audit rebuilds the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	First entry on the idle list (PSrIaescasehd.first).
<i>Bad Data</i> =	Last entry on the idle list (PSrIaescasehd.last).
<i>Good Data</i> =	Value of PSIAESCASENULL.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	PSlaescase[] array entry at the head of the idle list (PSrIaescasehd.first).
<i>Dump 2</i> =	PSlaescase[] array entry at the tail of the idle list (PSrIaescasehd.last).

Table 4 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A PSlaescase[] array entry is in use by a ALDB (PSlaescase[].aldb_index != PSALDBNULL) but either it is still linked to the idle list or point-to point-back checks failed between this PSlaescase[] array entry and the ALDB.
Possible Error Effect:	Data in the PSlaescase[] array may cause the application to assert.
Corrective Action Taken:	The audit unlinks the PSlaescase[] array entry from any list it is in, sets all fields to default values, and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The "forward" PSlaescase[] array entry.
<i>Bad Data</i> =	The "backward" PSlaescase[] array entry.
<i>Good Data</i> =	PSaldb[] array index (PSlaescase[].aldb_index).
<i>Logical Index</i> =	Index into the PSlaescase[] array.
<i>Dump 1</i> =	PSlaescase[] array entry.
<i>Dump 2</i> =	The "forward" PSlaescase[] array entry (PSlaescase[].fd_laescase_idx) if the index is in range (< PSMXLAESCASE).
<i>Dump 3</i> =	The "backward" PSlaescase[] array entry (PSlaescase[].fk_laescase_idx) if the index is in range (< PSMXLAESCASE).
<i>Dump 4</i> =	The associated PSaldb[] array entry (PSlaescase[].aldb_index) if the index is in range (< PSMXALDB).

Table 5 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The recovery in-progress flag (PSlaescase[].rec_inprog) flag in the PSlaescase[] tuple has been stuck set (DBYES) for longer than 5 minutes.
Possible Error Effect:	May lead to lost resources and exhaustion of the PSlaescase[] resource.
Corrective Action Taken:	The audit unlinks the PSlaescase[] array entry from any list it is in, sets all fields to default values, and links it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSaldb[] array index (PSlaescase[].aldb_index).
<i>Bad Data</i> =	Content of the recovery in-progress flag (PSlaescase[].rec_inprog).
<i>Good Data</i> =	DBNO.
<i>Logical Index</i> =	Index into the PSlaescase[] array.
<i>Dump 1</i> =	PSlaescase[] array entry.
<i>Dump 2</i> =	The associated PSaldb[] array entry (PSlaescase[].aldb_index) if the index is in range (< PSMXALDB).

LCCB-A

Software Release: 5E10 - 5E11

1. LOGICAL CHANNEL CONTROL BLOCK (LCCB) AUDIT

The LCCB audit is part of the switch maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The LCCB audit uses the packet switching logical channel control block array (PSlccb) and the access link control block array (PSalcb).

3. AUDIT CHECKS

The audit makes linkage checks within the PSlccb array and with PSalcb. It makes semantic checks to ensure the correctness of data in PSlccb. The audit driver is drps.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: AUD_CH

Error Code:	AUD_CH
Error Description:	Audit error found for the channel. The audit has found an LCCB linked to an ALCB that has an out of range PH channel value.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit will report the error and correct the LCCB field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's alcb_index that it is linked to.
<i>Bad Data =</i>	LCCB's ALCB's PH channel number.
<i>Good Data =</i>	PSMAXCHN - one more than the maximum value the PH channel number can be.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The PBG aggregate LCN count (PSpbgstat[].aggr_lcn) is inconsistent with the number of LCNs (LCCBs) associated with this PSpbgstat[] entry. The number of LCCBs that have the PSlccb[].loc_lcn_act and/or the PSlccb[].rem_lcn_act fields set to DBYES, should agree with the count in PSpbgstat[].aggr_lcn. The index to the PSpbgstat[] array is the PBG number found in PSlccb[].rem_pbg and/or PSlccb[].loc_pbg fields.
Possible Error Effect:	Incorrect overflow treatment could be applied. If PSpbgstat[].aggr_lcn is greater than the real count, calls could be blocked or charged premium when they shouldn't be. If PSpbgstat[].aggr_lcn is smaller than the real count, calls could be allowed or not charged premium when they should be.
Corrective Action Taken:	The audit will correct the PSpbgstat[].aggr_lcn field to the number of LCCBs associated with this entry.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the PSpbgstat[] entry.
<i>Bad Data</i> =	PSpbgstat[].aggr_lcn.
<i>Good Data</i> =	The number of LCCBs associated with this PSpbgstat[] entry.
<i>Logical Index</i> =	Index into PSpbgstat[] (PBG number).
<i>Dump</i> =	NA.

Table 3 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	The audit found an LCCB that has the retransmission buffer or holdbuffer arrays that have more than one "hole" in the array. These array must be continuous with only one starting element.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Count of "start/stops" in retransmission array.
<i>Bad Data</i> =	Count of "start/stops" in hold buffer array.
<i>Good Data</i> =	AUTWO - Maximum plus one of the number of starts.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The virtual call throughput class for the sending (PSlccb[].vc_tpc[PSCED]) and/or receiving (PSlccb[].vc_tpc[PSCING]) directions of an LCCB is/are out of range (PSTPC75 > PSlccb[].vc_tpc[] > PSTPC19200).
Possible Error Effect:	Calls could be blocked or charged premium when they shouldn't be.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSlccb[].vc_tpc[PSCED].
<i>Bad Data</i> =	PSlccb[].vc_tpc[PSCING].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 5 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The PH audit found an LCCB with an invalid owner field.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The error report is printed and AUFALSE returned to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LCCB's owner field.
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 6 Error Code: BADSUPVSTAT

Error Code:	BADSUPVSTAT
Error Description:	Bad suspected pvc status. The audit has found an LCCB whose receiving window size or sending window size is 0. If either of these is 0, no packets can be transmitted or received properly. Recover the LCCB and request a routine port conditioning of the PVC.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which

	recovers the LCCB to the free linked list. Since this is a PVC, a routine port conditioning is requested to reset up the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's receiving window size "iwsze_rcv".
<i>Bad Data</i> =	LCCB's sending window size "iwsze_send".
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 7 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	The PBG aggregate throughput count (PSpbgstat[].aggr_tpc) is inconsistent with the throughput class of LCCBs associated with this PSpbgstat[] entry. The throughput class of the LCCBs that have the PSlccb[].loc_tpc_act and/or the PSlccb[].rem_tpc_act fields set to DBYES, should agree with the count in PSpbgstat[].aggr_tpc. The index to the PSpbgstat[] array is the PBG number found in PSlccb[].rem_pbg and/or PSlccb[].loc_pbg fields. Only the larger throughput classes between the sending (PSlccb[].vc_tpc[PSCED]) and the receiving (PSlccb[].vc_tpc[PSCING]) directions is included in PSpbgstat[].aggr_tpc.
Possible Error Effect:	Incorrect overflow treatment could be applied. If PSpbgstat[].aggr_tpc is greater than the real count, calls could be blocked or charged premium when they shouldn't be. If PSpbgstat[].aggr_tpc is smaller than the real count, calls could be allowed or not charged premium when they should be.
Corrective Action Taken:	The audit will correct the PSpbgstat[].aggr_tpc field to the throughput class collected from the LCCBs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PSpbgstat[] array entry.
<i>Bad Data</i> =	PSpbgstat[].aggr_tpc.
<i>Good Data</i> =	The throughput class data collect from the LCCBs.
<i>Logical Index</i> =	The index into PSpbgstat[].
<i>Dump</i> =	NA.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The remote and/or the local PBG number of an LCCB is out of range (PSlccb[].rem_pbg and/or PSlccb[].loc_pbg > PSPBGMAX).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSlccb[].loc_pbg.
<i>Bad Data</i> =	PSlccb[].rem_pbg.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An LCCB, in use for an X.75 (protocol for inter-network packet switching between international gateway packet switches) or X.75' (protocol for inter-switch packet switching network within the same switch network, but by switches of different manufacturers), has the huntflag bit set (lccbptr→huntflag) while the logical channel number field, lcn, (lccbptr→lcn) is out of range.
Possible Error Effect:	If this LCCB was used correctly and the lcn field got corrupted, we will lose this call else it is a waste of resource as this LCCB should be on the free list.
Corrective Action Taken:	The huntflag field is disabled (set to PSDISABLED).

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's huntflag field (lccbptr→huntflag).
<i>Bad Data =</i>	LCCB's lcn field (lccbptr→lcn).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LCCB's cpstate field (lccbptr→cpstate).
<i>Dump 1 =</i>	The layer 3 data of type PSL3DATA associated with this LCCB.
<i>Dump 2 =</i>	LCCB's protstate field (lccbptr→protstate).
<i>Dump 3 =</i>	The al_pchn field of the ALCB associated with this LCCB (PSalcb[lccbptr→alcb_index].al_pchn).

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has a non-null buffer descriptor (BD) field. On the free or billing linked list, there should not be any BDs attached to the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's qdsetup_index (BD field).
<i>Bad Data =</i>	LCCB's mptmbd (modem pool TM BD field).
<i>Good Data =</i>	PSNULLOFF - Valid null value for the BD fields.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	(PSOFFSET) the LCCB's rctl_bfr field (BD). (PSOFFSET) the LCCB's ictrl_bfr field (BD). (PSOFFSET) the LCCB's intrpt_bfr field (BD).

Table 11 Error Code: BDPCR

Error Code:	BDPCR
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has a non-null BD field in either the holding buffer or retransmission buffer arrays. On the free or billing linked list, there should not be any BDs attached to the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's holding buffer element (hold_buff[]).
<i>Bad Data =</i>	LCCB's retransmission buffer element (retran_buff[]).
<i>Good Data =</i>	Index into the buffer array that is non-null.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 12 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	An LCCB in use for an X.75' trunk has its billflg enabled.
Possible Error Effect:	A customer may be billed incorrectly.
Corrective Action Taken:	The field is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's billflg.
<i>Bad Data =</i>	The value in the LCCB's billflg (PSENABLED).
<i>Good Data =</i>	PSDISABLED.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 13 Error Code: BLINK

Error Code:	BLINK
Error Description:	BD linkage is corrupted. The audit has found an LCCB linked to an ALCB for a VC call with a protocol state (protstate) that must have a BD linked to the LCCB, but the qdsetup index BD field is null.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's protocol state (protstate).
<i>Bad Data =</i>	LCCB's qdsetup index (XACs BD).
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 14 Error Code: CHALLOC

Error Code:	CHALLOC
Error Description:	The audit found an LCCB linked to an ALCB that has an modem pool terminal maintenance (MPTM) or inter-switch (IT) PHC type but the channel type "chantype" is not a non-virtual type.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The call is torn down and associated resources (such as, the LCCB are returned to the appropriate idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's DCDB's PH channel type "phctype" - write protect data indicating IT or MPTM call type.
<i>Bad Data =</i>	LCCB's channel type - "chantype".
<i>Good Data =</i>	PSNVC TYPE - valid value for non-VC type for the LCCB's channel type.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	(size PSOFFSET) LCCB's modem pool terminal maintenance BD value.

Table 15 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has found an LCCB where the COUNT of retransmission buffers and the hold buffers is greater than PSBFRIND.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will return the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's count of retransmission buffers.
<i>Bad Data =</i>	LCCB's count of hold buffers.
<i>Good Data =</i>	PSBFRIND - maximum value for total buffer counts.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 16 Error Code: CURWFLG

Error Code:	CURWFLG
Error Description:	The audit has found an LCCB whose receiving window size or sending window size is out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to recover the LCCB. Since this is a PVC, a routine port conditioning will be requested to reset up the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's receiving window size "r_ wsize".
<i>Bad Data =</i>	LCCB's sending window size "s_ wsize".
<i>Good Data =</i>	LCCB index.

<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 17 Error Code: CURWFLG

Error Code:	CURWFLG
Error Description:	The audit found an LCCB whose receiving window size or sending window size is out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to recover the LCCB. Since this is a PVC, a routine port conditioning will be requested to reset up the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's receiving window size "r_ wsize".
<i>Bad Data</i> =	LCCB's sending window size "r_ wsize".
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 18 Error Code: DFLG

Error Code:	DFLG
Error Description:	The huntflag should be disabled. The audit has found an LCCB whose data control data block's (DCDB) write protected huntgroup flag is NOT set but the LCCB's flag is enabled.
Possible Error Effect:	Prints a report and corrects the LCCB's field.
Corrective Action Taken:	The audit will report the error and set the LCCB's huntflag to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's logical channel number.
<i>Bad Data</i> =	LCCB's "huntflag" value.
<i>Good Data</i> =	PSDISABLED - valid value the huntflag should have.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 19 Error Code: DISP

Error Code:	DISP
Error Description:	Disabled pointer. Write protected huntgroup flag is set but the LCCB's logical channel number "lcn" is within the multi-line hunt group value.
Possible Error Effect:	Prints a report and corrects the LCCB's field.
Corrective Action Taken:	The audit will report the error and set the LCCB's huntflag to PSENABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Highest LCN value of hunted range.
<i>Bad Data</i> =	LCCB's LCN field.
<i>Good Data</i> =	Lowest LCN of huntable range (HiPVC+1).
<i>Logical Index</i> =	LCCB's cpstate.
<i>Dump</i> =	LCCB's protstate.
	LCCB's PH channel number.

Table 20 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has either the huntflag, mcp_tear or mcp_touch bit enabled. On the free or billing linked list, there should not be any of these fields set.
Possible Error Effect:	Prints a report and recovers the LCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's huntflag.
<i>Bad Data</i> =	LCCB's modem pool call processing tear bit. Indicating that the MPCP call is

	being torn down.
<i>Good Data</i> =	LCCB's modem pool call processing touch bit. Audit had previously seen the mcp_tear bit set and is timing the transient state.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 21 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The specified headcell's first or last elements are out of range.
Possible Error Effect:	Prints an error report and continues.
Corrective Action Taken:	AUFALSE is returned and the headcell is nulled. The linked list owned by the headcell is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting range for the first/last headcell element.
<i>Bad Data</i> =	Ending range for the first/last headcell element.
<i>Good Data</i> =	Null value of the first and last.
<i>Logical Index</i> =	headcell index.
<i>Dump</i> =	headcell dump or PSRHEADCELL type.

Table 22 Error Code: IMPID

Error Code:	IMPID
Error Description:	Invalid modem pool timing index. The audit has found an LCCB linked to an ALCB and there is a modem pool terminal maintenance BD offset but there isn't any timer pending.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will return the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's modem pool "mptmbd" BD offset.
<i>Bad Data</i> =	LCCB's timacc[] MPTM element.
<i>Good Data</i> =	PST3NULL - valid null value for the timer array index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 23 Error Code: INV

Error Code:	INV
Error Description:	Write protected huntgroup flag is set but the high 2w information in the DCDB write protected data and the high incoming values are both 0.
Possible Error Effect:	Prints a report and recovers the LCCB's.
Corrective Action Taken:	The audit will report the error and return AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	hihunt - either high incoming that is 0.
<i>Good Data</i> =	PH channel number.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	l3data information: DMPS_LCN pvc_h_lcn = High pvc assigned LCN range, DMPS_LCN l_lcn_in = Low incoming assigned LCN range, DMPS_LCN h_lcn_in = High incoming assigned LCN range, DMPS_LCN l_lcn_2w = Low 2 way assigned LCN range, DMPS_LCN h_lcn_2w = High 2 way assigned LCN range,

	DMPS_LCN l_lcn_out = Low outgoing assigned LCN range,
	DMPS_LCN h_lcn_out = High outgoing assigned LCN range,
	DMBOOL_mhglg = MLHG indicator.

Table 24 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An LCCB with an invalid cpstate field is queued on a DCCB. LCCBs queued on DCCBs should have the PSlccb[].cpstate field set to PSA_N0.
Possible Error Effect:	The Conditional Notification call associated with this LCCB entry will fail.
Corrective Action Taken:	The LCCB will be removed from the queue and recovered to the free linked list. Note that before removing the LCCB from the queue, the audit checks if the DCCB queue linked list is corrupted. If that is the case, the audit will call the queue recovery function to step through the PSlccb[] array and reconstruct the queue. Therefore, no errors are introduced by removing the LCCB from the queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	DCCB index.
<i>Bad Data</i> =	PSlccb[].cpstate.
<i>Good Data</i> =	PSA_N0.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 25 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has a non-null value in the timacc[] timer array used by X.25 access processor (XAC). On the free or billing linked list, there should not be any timers on for the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's index into the timacc[] array that is non-null.
<i>Bad Data</i> =	LCCB's tim value in the timacc[] array.
<i>Good Data</i> =	PST3NULL - valid null value for the timacc[] field.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 26 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The PH audit has found an LCCB on the free or billing linked list with the alcb_index field not null.
Possible Error Effect:	Prints a report and corrects the LCCB's alcb_index field.
Corrective Action Taken:	The LCCB alcb_index field is set to PSALCBNULL and the audit continues.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's owner (free linked or billing linked list).
<i>Bad Data</i> =	LCCB's alcb_index field.
<i>Good Data</i> =	PSALCBNULL - the valid null field for the alcb_index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 27 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG

Error Description:	The PH audit has found an LCCB on no linked lists (linked to an ALCB) with either the forward lccb index field or back lccb index field is not null. Since this is not on a linked list, these fields must be null.
Possible Error Effect:	Prints a report and corrects the faulty fields.
Corrective Action Taken:	The error report is printed and then the forward and backward lccb index fields are nulled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's linked list forward pointer (fdlccb ind).
<i>Bad Data =</i>	LCCB's linked list backward pointer (bklccb ind).
<i>Good Data =</i>	PSLCCBNULL - Valid null value of linkage fields.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 28 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	The PH audit has found an LCCB on no linked lists (linked to an ALCB) whose alcb index or lcn field is out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling functions which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's alcb index field.
<i>Bad Data =</i>	LCCB's logical channel number (lcn). The slot number of the LCCB's ALCB's lccb index array.
<i>Good Data =</i>	PSHMAXALCB - Maximum value of alcb index field.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 29 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The PH audit has found an LCCB on the free linked list with the logical channel number (lcn) field not zero.
Possible Error Effect:	Prints a report and zeroes the LCCB's lcn field.
Corrective Action Taken:	The LCCB lcn field is zeroed out and the audit continues.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LCCB's lcn field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 30 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The PH audit has found an LCCB on no linked lists (linked to an ALCB) and fails the point to point back check between LCCB and the ALCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The error report is printed and then AUFALSE is returned to recover the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's alcb index field.
<i>Bad Data =</i>	The LCCB that the LCCB's ALCB is pointing to.
<i>Good Data =</i>	LCCB's logical channel number (lcn). The slot number of the LCCB's ALCB's lccb index array.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 31 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit found a member error in the linked list of the specified audit.
Possible Error Effect:	Prints an error report and corrects the linked list.
Corrective Action Taken:	Return AUFALSE to calling function which rebuilds the linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Linked list null value.
<i>Logical Index</i> =	AUcount, the current element number.
<i>Dump</i> =	AULNK.

Table 32 Error Code: NMTIML

Error Code:	NMTIML
Error Description:	No marked timer was located. The audit found an LCCB linked to an ALCB for a VC with a protocol state (protstate) that requires a timer pending. Both the timacc[] array and the return from PST3query should indicate a timer and they don't.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's protocol state (protstate).
<i>Bad Data</i> =	LCCB's timacc[] VC element (timer tag).
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 33 Error Code: ORCOUNTS

Error Code:	ORCOUNTS
Error Description:	The audit has found an LCCB where the number of non-null elements in the holdbuffer array does not match the "hb_qd" count recorded in the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LCCB's "hb_qd" count (IPs count of buffers that are being held).
<i>Good Data</i> =	Audits count of the non-null values in the hold buffer array.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 34 Error Code: OWNER

Error Code:	OWNER
Error Description:	The PSlccb[].qdccb_own field is out of range. This field should be set to a valid DCCB index in which the LCCB is queued.
Possible Error Effect:	The condition notification call associated with this LCCB entry will fail.
Corrective Action Taken:	The audit recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PSlccb[].qdccb_own.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 35 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The PH audit found an LCCB on the free linked or billing linked list with the

	forward or backward LCCB pointers of the linked list out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The error report is printed and AUFALSE is returned to the calling function which the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSLCCBNULL - the valid null field for the forward or backward pointers.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	AULNK.

Table 36 Error Code: PPAUD_CH

Error Code:	PPAUD_CH
Error Description:	PVC audit error on channel type. The audit found an LCCB linked to an ALCB of a PVC type but the channel type is bad.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The call is torn down and associated resources (such as, the LCCB are returned to the appropriate idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's logical channel number (LCN) indicating that this is a PVC call.
<i>Bad Data</i> =	LCCB's channel type - "chantype".
<i>Good Data</i> =	PSPVC_TYPE - valid value for permanent virtual call (PVC) type for the LCCB's channel type.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 37 Error Code: QUAR

Error Code:	QUAR
Error Description:	The audit found an LCCB with its quarantine bit set.
Possible Error Effect:	Prints out a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LCCB's quarantine bit value (quar_bit).
<i>Good Data</i> =	DBNO - valid value for the quarantine bit indicating no problems.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 38 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The audit found an LCCB where the number of non-null elements in the holdbuffer array plus the retransmission buffer does not match the value of the LCCB's rtb_qd (retransmission count) and LCCB's shold_qd (hold buffer count).
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "rtb_qd" (retransmission count).
<i>Bad Data</i> =	LCCB's "shold_qd" count (IPs count of buffers that are being held).
<i>Good Data</i> =	Audits count of the non-null values in the retransmission buffer and hold buffer.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 39 Error Code: R_CSTATE

Error Code:	R_CSTATE
Error Description:	Range check on CP state. The audit found an LCCB linked to an ALCB for a VC

	call with an out of range "cpstate" value.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LCCB's "cpstate" state value.
<i>Good Data =</i>	PSA F6 - Maximum "cpstate" value.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 40 Error Code: RETRYTIME

Error Code:	RETRYTIME
Error Description:	A timer was not found when retrying to send or hold a buffer. The audit found an LCCB where there is more than one buffer marked in the retransmission array or the holdbuffer array which indicates there is at least one BD attached to the LCCB but there isn't any timer pending for this LCCB. There must be a timer for any BD on an LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Audit's count of hold buffer BDs.
<i>Bad Data =</i>	Audit's count of retransmission BDs.
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 41 Error Code: SPTIME

Error Code:	SPTIME
Error Description:	Suspect PVC timer. The audit has found an LCCB linked to an ALCB for a PVC call with a "cpstate" that requires a timer pending. No timer is indicated, so the LCCB will be recovered.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's cpstate state value.
<i>Bad Data =</i>	autimer - local value indicating that there is a timer tag in tim[] array. This should be DBYES.
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 42 Error Code: STATE

Error Code:	STATE
Error Description:	The audit found an LCCB linked to an ALCB with an invalid state field.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LCCB's protocol state (protstate).
<i>Good Data =</i>	PSMAX_VCSTATE - Maximum protstate value.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 43 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The queued incoming field (PSlccb[].qd_incom) of an LCCB is not PSNULLOFF. This field should only be used when the internal circuit state (PSlccb[].cpstate) is PSA_N0.
Possible Error Effect:	The condition notification call associated with this entry will fail.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PSlccb[].qd_incom.
<i>Bad Data =</i>	PSlccb[].cpstate.
<i>Good Data =</i>	PSNULLOFF.
<i>Logical Index =</i>	LCCB's logical channel number (PSlccb[].lcn).
<i>Dump =</i>	NA.

Table 44 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The audit found an LCCB linked to an ALCB of a non-virtual call type that has both the XAC protocol state field and the IP "cpstate" field indicating states where no timers are pending. This LCCB configuration must have a timer and one of the two state fields must be in a state requiring a timer.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and return AUFALSE which recovers LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's protocol state "protstate" (for XAC).
<i>Bad Data =</i>	LCCB's cpstate (for IP).
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 45 Error Code: STUCK

Error Code:	STUCK
Error Description:	The audit found an LCCB that is currently waiting for PVC data from the SMP for update of the completed PVC information (pvcw4data is PSENABLE), however there are no timers pending. Since the "wait for data" state is considered a transient state by audits, there must be a timer to trigger some resultant action but the audits did not find one pending.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's logical channel number (lcn).
<i>Bad Data =</i>	LCCB's protocol state "protstate" (for XAC).
<i>Good Data =</i>	LCCB's ALCBs PH channel number ("al_phcn").
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 46 Error Code: TID

Error Code:	TID
Error Description:	Timer index is invalid. The audit found an LCCB linked to an ALCB. There is a BD offset in "qdsetup_index" but no timer is pending.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's "qdsetup" BD offset.

<i>Bad Data</i> =	LCCB's "protstate".
<i>Good Data</i> =	LCCB's timacc[PST3VC].
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	(size PSTINDEX) return from PSt3query indicating that there is a timer or not.

Table 47 Error Code: TIDBAD

Error Code:	TIDBAD
Error Description:	Timer index bad. The audit found an LCCB linked to an ALCB with a timer tag value in the tim[] array; however, PSt3query has no record of any actual timers pending for this LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "cpstate".
<i>Bad Data</i> =	LCCB's "protstate".
<i>Good Data</i> =	LCCB's ALCB's PH channel number ("al_phcn").
<i>Logical Index</i> =	Logical PH number.
<i>Dump</i> =	LCCB's tim array - "tim[PSTIMTAGS]".

Table 48 Error Code: TIME_EXC

Error Code:	TIME_EXC
Error Description:	A timer is executing. The audit found an LCCB on the free linked list or billing list of LCCBs that has a recorded timer when PSt3_query is called to inquire information for the timers for this LCCB. For LCCBs on a billing or free linked list, there should not be any timers on for the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The return from PSt3_query() timer inquiry function.
<i>Good Data</i> =	PSNOTIMER - valid null value for the PSt3_query function for no timer.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 49 Error Code: TIMEORDER

Error Code:	TIMEORDER
Error Description:	The audit found an LCCB linked to an ALCB with a timer tag value in the PVC timer slot of the timacc[] array; however, the protstates is not S1 or S2, or the pvcw4data field is DBNO and should be DBYES.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Timer tag of the PVC slot of timacc[] array.
<i>Bad Data</i> =	LCCB's protocol state "protstate".
<i>Good Data</i> =	LCCB's pvcw4data that should be DBYES.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 50 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	An LCCB with no active timer is queued on a DCCB.
Possible Error Effect:	The Condition Notification call associated with this entry will fail.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	DCCB index.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 51 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The audit found an LCCB on the free linked list or billing list of LCCBs that has a non-null value in the tim[] timer array used by interprocessor communications (IP). On the free or billing linked list, there should not be any timers on for the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's index into the tim[] array that is non-null.
<i>Bad Data</i> =	LCCB's tim value in the tim[] array.
<i>Good Data</i> =	PST3NULL - valid null value for the tim[] field.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 52 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The audit found an LCCB linked to an ALCB. There is at least one BD in either ictl_bfr, intrpt_bfr, or rctl_bfr but there is no timer pending. Whenever a BD is linked to an LCCB, there must be timer; therefore the LCCB will be recovered.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "ictl_bfr" BD offset.
<i>Bad Data</i> =	LCCB's "intrpt_bfr" BD offset.
<i>Good Data</i> =	PSNULLOFF - valid null BD offset.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	(PSOFFSET size) - LCCB's "rctl_bfr" BD offset.

Table 53 Error Code: TPID

Error Code:	TPID
Error Description:	A timer pending invalid. The audit has found an LCCB linked to an ALCB for a VC call with a "cpstate" that requires a timer pending. No timer is indicated, so the LCCB will be recovered.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "cpstate" state value.
<i>Bad Data</i> =	autimer - local value indicating that there is a timer tag in tim[] array. This should be DBYES.
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 54 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The audit found an LCCB whose modem pool call is presently being torn down.

	Audits will time this transient state for at least 5 minutes. No audit sees this condition 5 minutes later so the LCCB will be torn down.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PH channel number.
<i>Bad Data</i> =	LCCB's modem pool call touch bit that audit sets and will interrogate 5 minutes later to see if it is set and needs to tear the LCCB down.
<i>Good Data</i> =	Logical PH number.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 55 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit found an LCCB linked to an ALCB of a VC type but the channel type is found bad.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The call is torn down and associated resources (such as, the LCCB are returned to the appropriate idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's logical channel number.
<i>Bad Data</i> =	LCCB's channel type - "chantype".
<i>Good Data</i> =	PSVC_TYPE - valid value for virtual call (VC) type for the LCCB's channel type.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 56 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, audit ID that was running at the time of the interrupt.
<i>Dump</i> =	NA.

LCCB-B

Software Release: 5E12 only

1. LOGICAL CHANNEL CONTROL BLOCK (LCCB) AUDIT

The LCCB audit is part of the switch maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The LCCB audit uses the packet switching logical channel control block array (PSlccb) and the access link control block array (PSalcb).

3. AUDIT CHECKS

The audit makes linkage checks within the PSlccb array and with PSalcb. It makes semantic checks to ensure the correctness of data in PSlccb. The audit driver is drps.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: AUD_CH

Error Code:	AUD_CH
Error Description:	Audit error found for the channel. The audit has found an LCCB linked to an ALCB that has an out of range PH channel value.
Possible Error Effect:	Prints a report and corrects the field.
Corrective Action Taken:	The audit will report the error and correct the LCCB field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's alcb_index that it is linked to.
<i>Bad Data =</i>	LCCB's ALCB's PH channel number.
<i>Good Data =</i>	PSMAXCHN - one more than the maximum value the PH channel number can be.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The PBG aggregate LCN count (PSpbgstat[].aggr_lcn) is inconsistent with the number of LCNs (LCCBs) associated with this PSpbgstat[] entry. The number of LCCBs that have the PSlccb[].loc_lcn_act and/or the PSlccb[].rem_lcn_act fields set to DBYES, should agree with the count in PSpbgstat[].aggr_lcn. The index to the PSpbgstat[] array is the PBG number found in PSlccb[].rem_pbg and/or PSlccb[].loc_pbg fields.
Possible Error Effect:	Incorrect overflow treatment could be applied. If PSpbgstat[].aggr_lcn is greater than the real count, calls could be blocked or charged premium when they shouldn't be. If PSpbgstat[].aggr_lcn is smaller than the real count, calls could be allowed or not charged premium when they should be.
Corrective Action Taken:	The audit will correct the PSpbgstat[].aggr_lcn field to the number of LCCBs associated with this entry.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the PSpbgstat[] entry.
<i>Bad Data</i> =	PSpbgstat[].aggr_lcn.
<i>Good Data</i> =	The number of LCCBs associated with this PSpbgstat[] entry.
<i>Logical Index</i> =	Index into PSpbgstat[] (PBG number).
<i>Dump</i> =	NA.

Table 3 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	The audit found an LCCB that has the retransmission buffer or holdbuffer arrays that have more than one "hole" in the array. These array must be continuous with only one starting element.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Count of "start/stops" in retransmission array.
<i>Bad Data</i> =	Count of "start/stops" in hold buffer array.
<i>Good Data</i> =	AUTWO - Maximum plus one of the number of starts.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 4 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The virtual call throughput class for the sending (PSlccb[].vc_tpc[PSCED]) and/or receiving (PSlccb[].vc_tpc[PSCING]) directions of an LCCB is/are out of range (PSTPC75 > PSlccb[].vc_tpc[] > PSTPC19200).
Possible Error Effect:	Calls could be blocked or charged premium when they shouldn't be.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSlccb[].vc_tpc[PSCED].
<i>Bad Data</i> =	PSlccb[].vc_tpc[PSCING].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 5 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The PH audit found an LCCB with an invalid owner field.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The error report is printed and AUFALSE returned to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LCCB's owner field.
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 6 Error Code: BADSUPVSTAT

Error Code:	BADSUPVSTAT
Error Description:	Bad suspected pvc status. The audit has found an LCCB whose receiving window size or sending window size is 0. If either of these is 0, no packets can be transmitted or received properly. Recover the LCCB and request a routine port conditioning of the PVC.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which

	recovers the LCCB to the free linked list. Since this is a PVC, a routine port conditioning is requested to reset up the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's receiving window size "iwsze_rcv".
<i>Bad Data</i> =	LCCB's sending window size "iwsze_send".
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 7 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	The PBG aggregate throughput count (PSpbgstat[].aggr_tpc) is inconsistent with the throughput class of LCCBs associated with this PSpbgstat[] entry. The throughput class of the LCCBs that have the PSlccb[].loc_tpc_act and/or the PSlccb[].rem_tpc_act fields set to DBYES, should agree with the count in PSpbgstat[].aggr_tpc. The index to the PSpbgstat[] array is the PBG number found in PSlccb[].rem_pbg and/or PSlccb[].loc_pbg fields. Only the larger throughput classes between the sending (PSlccb[].vc_tpc[PSCED]) and the receiving (PSlccb[].vc_tpc[PSCING]) directions is included in PSpbgstat[].aggr_tpc.
Possible Error Effect:	Incorrect overflow treatment could be applied. If PSpbgstat[].aggr_tpc is greater than the real count, calls could be blocked or charged premium when they shouldn't be. If PSpbgstat[].aggr_tpc is smaller than the real count, calls could be allowed or not charged premium when they should be.
Corrective Action Taken:	The audit will correct the PSpbgstat[].aggr_tpc field to the throughput class collected from the LCCBs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PSpbgstat[] array entry.
<i>Bad Data</i> =	PSpbgstat[].aggr_tpc.
<i>Good Data</i> =	The throughput class data collect from the LCCBs.
<i>Logical Index</i> =	The index into PSpbgstat[].
<i>Dump</i> =	NA.

Table 8 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The remote and/or the local PBG number of an LCCB is out of range (PSlccb[].rem_pbg and/or PSlccb[].loc_pbg > PSPBGMAX).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSlccb[].loc_pbg.
<i>Bad Data</i> =	PSlccb[].rem_pbg.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 9 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An LCCB, in use for an X.75 (protocol for inter-network packet switching between international gateway packet switches) or X.75' (protocol for inter-switch packet switching network within the same switch network, but by switches of different manufacturers), has the huntflag bit set (lccbptr->huntflag) while the logical channel number field, lcn, (lccbptr->lcn) is out of range.
Possible Error Effect:	If this LCCB was used correctly and the lcn field got corrupted, we will lose this call else it is a waste of resource as this LCCB should be on the free list.
Corrective Action Taken:	The huntflag field is disabled (set to PSDISABLED).

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's huntflag field (lccbptr->huntflag).
<i>Bad Data =</i>	LCCB's lcn field (lccbptr->lcn).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LCCB's cpstate field (lccbptr->cpstate).
<i>Dump 1 =</i>	The layer 3 data of type PSL3DATA associated with this LCCB.
<i>Dump 2 =</i>	LCCB's protstate field (lccbptr->protstate).
<i>Dump 3 =</i>	The al_phcn field of the ALCB associated with this LCCB (PSalcb[lccbptr->alcb_index].al_phcn).

Table 10 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has a non-null buffer descriptor (BD) field. On the free or billing linked list, there should not be any BDs attached to the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's qdsetup_index (BD field).
<i>Bad Data =</i>	LCCB's mptmbd (modem pool TM BD field).
<i>Good Data =</i>	PSNULLOFF - Valid null value for the BD fields.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	(PSOFFSET) the LCCB's rctl_bfr field (BD). (PSOFFSET) the LCCB's ictrl_bfr field (BD). (PSOFFSET) the LCCB's intrpt_bfr field (BD).

Table 11 Error Code: BDPCR

Error Code:	BDPCR
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has a non-null BD field in either the holding buffer or retransmission buffer arrays. On the free or billing linked list, there should not be any BDs attached to the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's holding buffer element (hold_buf[]).
<i>Bad Data =</i>	LCCB's retransmission buffer element (retran_buf[]).
<i>Good Data =</i>	Index into the buffer array that is non-null.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 12 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	An LCCB in use for an X.75' trunk has its billflg enabled.
Possible Error Effect:	A customer may be billed incorrectly.
Corrective Action Taken:	The field is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's billflg.
<i>Bad Data =</i>	The value in the LCCB's billflg (PSENABLED).
<i>Good Data =</i>	PSDISABLED.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 13 Error Code: BLINK

Error Code:	BLINK
Error Description:	BD linkage is corrupted. The audit has found an LCCB linked to an ALCB for a VC call with a protocol state (protstate) that must have a BD linked to the LCCB, but the qdsetup index BD field is null.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's protocol state (protstate).
<i>Bad Data =</i>	LCCB's qdsetup index (XACs BD).
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 14 Error Code: CHALLOC

Error Code:	CHALLOC
Error Description:	The audit found an LCCB linked to an ALCB that has an modem pool terminal maintenance (MPTM) or inter-switch (IT) PHC type but the channel type "chantype" is not a non-virtual type.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The call is torn down and associated resources (such as, the LCCB are returned to the appropriate idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's DCDB's PH channel type "phctype" - write protect data indicating IT or MPTM call type.
<i>Bad Data =</i>	LCCB's channel type - "chantype".
<i>Good Data =</i>	PSNVC TYPE - valid value for non-VC type for the LCCB's channel type.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	(Size PSOFFSET) LCCB's modem pool terminal maintenance BD value.

Table 15 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit has found an LCCB where the COUNT of retransmission buffers and the hold buffers is greater than PSBFRIND.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will return the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's count of retransmission buffers.
<i>Bad Data =</i>	LCCB's count of hold buffers.
<i>Good Data =</i>	PSBFRIND - maximum value for total buffer counts.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 16 Error Code: CURWFLG

Error Code:	CURWFLG
Error Description:	The audit has found an LCCB whose receiving window size or sending window size is out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to recover the LCCB. Since this is a PVC, a routine port conditioning will be requested to reset up the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's receiving window size "r_ wsize".
<i>Bad Data =</i>	LCCB's sending window size "s_ wsize".
<i>Good Data =</i>	LCCB index.

<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 17 Error Code: CURWFLG

Error Code:	CURWFLG
Error Description:	The audit found an LCCB whose receiving window size or sending window size is out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to recover the LCCB. Since this is a PVC, a routine port conditioning will be requested to reset up the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's receiving window size "r_ wsize".
<i>Bad Data</i> =	LCCB's sending window size "r_ wsize".
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 18 Error Code: DFLG

Error Code:	DFLG
Error Description:	The huntflag should be disabled. The audit has found an LCCB whose data control data block's (DCDB) write protected huntgroup flag is NOT set but the LCCB's flag is enabled.
Possible Error Effect:	Prints a report and corrects the LCCB's field.
Corrective Action Taken:	The audit will report the error and set the LCCB's huntflag to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's logical channel number.
<i>Bad Data</i> =	LCCB's "huntflag" value.
<i>Good Data</i> =	PSDISABLED - valid value the huntflag should have.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 19 Error Code: DISP

Error Code:	DISP
Error Description:	Disabled pointer. Write protected hunt group flag is set but the LCCB's logical channel number "lcn" is within the multi-line hunt group value.
Possible Error Effect:	Prints a report and corrects the LCCB's field.
Corrective Action Taken:	The audit will report the error and set the LCCB's hunt flag to PSENABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Highest LCN value of hunted range.
<i>Bad Data</i> =	LCCB's LCN field.
<i>Good Data</i> =	Lowest LCN of huntable range (HiPVC+1).
<i>Logical Index</i> =	LCCB's cpstate.
<i>Dump</i> =	LCCB's protstate.
	LCCB's PH channel number.

Table 20 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has either the huntflag, mcp_tear or mcp_touch bit enabled. On the free or billing linked list, there should not be any of these fields set.
Possible Error Effect:	Prints a report and recovers the LCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's huntflag.
<i>Bad Data</i> =	LCCB's modem pool call processing tear bit. Indicating that the MPCP call is

	being torn down.
<i>Good Data</i> =	LCCB's modem pool call processing touch bit. Audit had previously seen the mcp_tear bit set and is timing the transient state.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 21 Error Code: EPITIME

Error Code:	EPITIME
Error Description:	An LCCB is in cpstate PSA_NO, but has no timer assigned to it.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's cpstate field.
<i>Bad Data</i> =	tgret.
<i>Good Data</i> =	LCCB index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 22 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The specified headcell's first or last elements are out of range.
Possible Error Effect:	Prints an error report and continues.
Corrective Action Taken:	AUFALSE is returned and the headcell is nulled. The linked list owned by the headcell is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting range for the first/last headcell element.
<i>Bad Data</i> =	Ending range for the first/last headcell element.
<i>Good Data</i> =	Null value of the first and last.
<i>Logical Index</i> =	headcell index.
<i>Dump</i> =	headcell dump or PSRHEADCELL type.

Table 23 Error Code: IMPID

Error Code:	IMPID
Error Description:	Invalid modem pool timing index. The audit has found an LCCB linked to an ALCB and there is a modem pool terminal maintenance BD offset but there isn't any timer pending.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will return the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's modem pool "mptmbd" BD offset.
<i>Bad Data</i> =	LCCB's timacc[] MPTM element.
<i>Good Data</i> =	PST3NULL - valid null value for the timer array index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	The LCCB.

Table 24 Error Code: INV

Error Code:	INV
Error Description:	Write protected huntgroup flag is set but the high 2w information in the DCDB write protected data and the high incoming values are both 0.
Possible Error Effect:	Prints a report and recovers the LCCB's.
Corrective Action Taken:	The audit will report the error and return AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	hihunt - either high incoming that is 0.

<i>Good Data</i> =	PH channel number.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	l3data information: DMPS_LCN pvc_h_lcn = High pvc assigned LCN range, DMPS_LCN l_lcn_in = Low incoming assigned LCN range, DMPS_LCN h_lcn_in = High incoming assigned LCN range, DMPS_LCN l_lcn_2w = Low 2 way assigned LCN range, DMPS_LCN h_lcn_2w = High 2 way assigned LCN range, DMPS_LCN l_lcn_out = Low outgoing assigned LCN range, DMPS_LCN h_lcn_out = High outgoing assigned LCN range, DMBOOL mhgflg = MLHG indicator.

Table 25 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An LCCB with an invalid cpstate field is queued on a DCCB. LCCBs queued on DCCBs should have the PSlccb[].cpstate field set to PSA_N0.
Possible Error Effect:	The conditional notification call associated with this LCCB entry will fail.
Corrective Action Taken:	The LCCB will be removed from the queue and recovered to the free linked list. Note that before removing the LCCB from the queue, the audit checks if the DCCB queue linked list is corrupted. If that is the case, the audit will call the queue recovery function to step through the PSlccb[] array and reconstruct the queue. Therefore, no errors are introduced by removing the LCCB from the queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	DCCB index.
<i>Bad Data</i> =	PSlccb[].cpstate.
<i>Good Data</i> =	PSA_N0.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 26 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The audit has found an LCCB on the free linked list or billing list of LCCBs that has a non-null value in the timacc[] timer array used by X.25 Access processor (XAC). On the free or billing linked list, there should not be any timers on for the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's index into the timacc[] array that is non-null.
<i>Bad Data</i> =	LCCB's tim value in the timacc[] array.
<i>Good Data</i> =	PST3NULL - valid null value for the timacc[] field.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 27 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED

Error Description:	The PH audit has found an LCCB on the free or billing linked list with the alcb_index field not null.
Possible Error Effect:	Prints a report and corrects the LCCB's alcb_index field.
Corrective Action Taken:	The LCCB alcb_index field is set to PSALCBNULL and the audit continues.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's owner (free linked or billing linked list).
<i>Bad Data</i> =	LCCB's alcb_index field.
<i>Good Data</i> =	PSALCBNULL - the valid null field for the alcb_index.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 28 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The PH audit has found an LCCB on no linked lists (linked to an ALCB) with either the forward lccb index field or back lccb index field is not null. Since this is not on a linked list, these fields must be null.
Possible Error Effect:	Prints a report and corrects the faulty fields.
Corrective Action Taken:	The error report is printed and then the forward and backward lccb index fields are nulled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's linked list forward pointer (fdlccb_ind).
<i>Bad Data</i> =	LCCB's linked list backward pointer (bklccb_ind).
<i>Good Data</i> =	PSLCCBNULL - Valid null value of linkage fields.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 29 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	The PH audit has found an LCCB on no linked lists (linked to an ALCB) whose alcb_index or lcn field is out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE to the calling functions which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's alcb_index field.
<i>Bad Data</i> =	LCCB's logical channel number (lcn). The slot number of the LCCB's ALCB's lccb_index array.
<i>Good Data</i> =	PSHMAXALCB - Maximum value of alcb_index field.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 30 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	The PH audit has found an LCCB on the free linked list with the logical channel number (lcn) field not zero.
Possible Error Effect:	Prints a report and zeroes the LCCB's lcn field.
Corrective Action Taken:	The LCCB lcn field is zeroed out and the audit continues.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LCCB's lcn field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 31 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The PH audit has found an LCCB on no linked lists (linked to an ALCB) and fails

	the point to point back check between LCCB and the ALCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The error report is printed and then AUFALSE is returned to recover the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's alcb_index field.
<i>Bad Data =</i>	The LCCB that the LCCB's ALCB is pointing to.
<i>Good Data =</i>	LCCB's logical channel number (lcn). The slot number of the LCCB's ALCB's lccb_index array.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 32 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit found a member error in the linked list of the specified audit.
Possible Error Effect:	Prints an error report and corrects the linked list.
Corrective Action Taken:	Return AUFALSE to calling function which rebuilds the linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Linked list null value.
<i>Logical Index =</i>	AUcount, the current element number.
<i>Dump =</i>	AULNK.

Table 33 Error Code: NMTIML

Error Code:	NMTIML
Error Description:	No marked timer was located. The audit found an LCCB linked to an ALCB for a VC with a protocol state (protstate) that requires a timer pending. Both the timacc[] array and the return from PST3query should indicate a timer and they don't.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's protocol state (protstate).
<i>Bad Data =</i>	LCCB's timacc[] VC element (timer tag).
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 34 Error Code: ORCOUNTS

Error Code:	ORCOUNTS
Error Description:	The audit has found an LCCB where the number of non-null elements in the holdbuffer array does not match the "hb_qd" count recorded in the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LCCB's "hb_qd" count (IPs count of buffers that are being held).
<i>Good Data =</i>	Audits count of the non-null values in the hold buffer array.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 35 Error Code: OWNER

Error Code:	OWNER
Error Description:	The PSlccb[].qdccb_own field is out of range. This field should be set to a valid

	DCCB index in which the LCCB is queued.
Possible Error Effect:	The condition notification call associated with this LCCB entry will fail.
Corrective Action Taken:	The audit recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PSlccbf].qdccb_own.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 36 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The PH audit found an LCCB on the free linked or billing linked list with the forward or backward LCCB pointers of the linked list out of range.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The error report is printed and AUFALSE is returned to the calling function which the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSLCCBNUL - the valid null field for the forward or backward pointers.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	AULNK.

Table 37 Error Code: PPAUD_CH

Error Code:	PPAUD_CH
Error Description:	PVC audit error on channel type. The audit found an LCCB linked to an ALCB of a PVC type but the channel type is bad.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The call is torn down and associated resources (such as, the LCCB are returned to the appropriate idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's logical channel number (LCN) indicating that this is a PVC call.
<i>Bad Data</i> =	LCCB's channel type - "chantype".
<i>Good Data</i> =	PSPVC_TYPE - valid value for permanent virtual call (PVC) type for the LCCB's channel type.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 38 Error Code: QUAR

Error Code:	QUAR
Error Description:	The audit found an LCCB with its quarantine bit set.
Possible Error Effect:	Prints out a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE to the calling function which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LCCB's quarantine bit value (quar_bit).
<i>Good Data</i> =	DBNO - valid value for the quarantine bit indicating no problems.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 39 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The audit found an LCCB where the number of non-null elements in the holdbuffer array plus the retransmission buffer does not match the value of the LCCB's rtb_qd (retransmission count) and LCCB's shold_qd (hold buffer count).

Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's "rtb_qd" (retransmission count).
<i>Bad Data =</i>	LCCB's "shold_qd" count (IPs count of buffers that are being held).
<i>Good Data =</i>	Audits count of the non-null values in the retransmission buffer and hold buffer.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 40 Error Code: R_CSTATE

Error Code:	R_CSTATE
Error Description:	Range check on CP state. The audit found an LCCB linked to an ALCB for a VC call with an out of range "cpstate" value.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LCCB's "cpstate" state value.
<i>Good Data =</i>	PSA F6 - Maximum "cpstate" value.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 41 Error Code: RETRYTIME

Error Code:	RETRYTIME
Error Description:	A timer was not found when retrying to send or hold a buffer. The audit found an LCCB where there is more than one buffer marked in the retransmission array or the holdbuffer array which indicates there is at least one BD attached to the LCCB but there isn't any timer pending for this LCCB. There must be a timer for any BD on an LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Audit's count of hold buffer BDs.
<i>Bad Data =</i>	Audit's count of retransmission BDs.
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 42 Error Code: SPTIME

Error Code:	SPTIME
Error Description:	Suspect PVC timer. The audit has found an LCCB linked to an ALCB for a PVC call with a "cpstate" that requires a timer pending. No timer is indicated, so the LCCB will be recovered.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's cpstate state value.
<i>Bad Data =</i>	tqret - local value used internally for debugging.
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	The LCCB.

Table 43 Error Code: STATE

Error Code:	STATE
Error Description:	The audit found an LCCB linked to an ALCB with an invalid state field.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LCCB's protocol state (protstate).
<i>Good Data =</i>	PSMAX_VCSTATE - Maximum protstate value.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 44 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The queued incoming field (PSlccb[].qd_incom) of an LCCB is not PSNULLOFF. This field should only be used when the internal circuit state (PSlccb[].cpstate) is PSA_N0.
Possible Error Effect:	The condition notification call associated with this entry will fail.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PSlccb[].qd_incom.
<i>Bad Data =</i>	PSlccb[].cpstate.
<i>Good Data =</i>	PSNULLOFF.
<i>Logical Index =</i>	LCCB's logical channel number (PSlccb[].lcn).
<i>Dump =</i>	NA.

Table 45 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The audit found an LCCB linked to an ALCB of a non-virtual call type that has both the XAC protocol state field and the IP "cpstate" field indicating states where no timers are pending. This LCCB configuration must have a timer and one of the two state fields must be in a state requiring a timer.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and return AUFALSE which recovers LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's protocol state "protstate" (for XAC).
<i>Bad Data =</i>	LCCB's cpstate (for IP).
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 46 Error Code: STUCK

Error Code:	STUCK
Error Description:	The audit found an LCCB that is currently waiting for PVC data from the SMP for update of the completed PVC information (pvcw4data is PSENABLE), however there are no timers pending. Since the "wait for data" state is considered a transient state by audits, there must be a timer to trigger some resultant action but the audits did not find one pending.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's logical channel number (lcn).
<i>Bad Data =</i>	LCCB's protocol state "protstate" (for XAC).
<i>Good Data =</i>	LCCB's ALCBs PH channel number ("al_phcn").
<i>Logical Index =</i>	LCCB index.

<i>Dump</i> =	NA.

Table 47 Error Code: TID

Error Code:	TID
Error Description:	Timer index is invalid. The audit found an LCCB linked to an ALCB. There is a BD offset in "qdsetup_index" but no timer is pending.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "qdsetup" BD offset.
<i>Bad Data</i> =	LCCB's "protstate".
<i>Good Data</i> =	LCCB's timacc[PST3VC].
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	The LCCB.

Table 48 Error Code: TIDBAD

Error Code:	TIDBAD
Error Description:	Timer index bad. The audit found an LCCB linked to an ALCB with a timer tag value in the tim[] array however, PST3query has no record of any actual timers pending for this LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "cpstate".
<i>Bad Data</i> =	LCCB's "protstate".
<i>Good Data</i> =	LCCB's ALCB's PH channel number ("al_phcn").
<i>Logical Index</i> =	Logical PH number.
<i>Dump</i> =	LCCB's tim array - "tim[PSTIMTAGS]".

Table 49 Error Code: TIME_EXC

Error Code:	TIME_EXC
Error Description:	A timer is executing. The audit found an LCCB on the free linked list or billing list of LCCBs that has a recorded timer when PST3_query is called to inquire information for the timers for this LCCB. For LCCBs on a billing or free linked list, there should not be any timers on for the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	tqret - local value used internally for debugging.
<i>Good Data</i> =	PSTNULLTE - valid null value for the PSte_query function for no timer.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 50 Error Code: TIMEORDER

Error Code:	TIMEORDER
Error Description:	The audit found an LCCB linked to an ALCB with a timer tag value in the PVC timer slot of the timacc[] array however, the protstates is not S1 or S2, or the pvcw4data field is DBNO and should be DBYES.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Timer tag of the PVC slot of timacc[] array.
<i>Bad Data</i> =	LCCB's protocol state "protstate".
<i>Good Data</i> =	LCCB's pvcw4data that should be DBYES.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 51 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	An LCCB with no active timer is queued on a DCCB.
Possible Error Effect:	The condition notification call associated with this entry will fail.
Corrective Action Taken:	The LCCB is recovered to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	DCCB index.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 52 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The audit found an LCCB on the free linked list or billing list of LCCBs that has a non-null value in the tim[] timer array used by interprocessor communications (IP). On the free or billing linked list, there should not be any timers on for the LCCB.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's index into the tim[] array that is non-null.
<i>Bad Data</i> =	LCCB's tim value in the tim[] array.
<i>Good Data</i> =	PST3NULL - valid null value for the tim[] field.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	NA.

Table 53 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The audit found an LCCB linked to an ALCB. There is at least one BD in either ictl_bfr, intrpt_bfr, or rctl_bfr but there is no timer pending. Whenever a BD is linked to an LCCB, there must be timer therefore the LCCB will be recovered.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB's "ictl_bfr" BD offset.
<i>Bad Data</i> =	LCCB's "intrpt_bfr" BD offset.
<i>Good Data</i> =	PSNULLOFF - valid null BD offset.
<i>Logical Index</i> =	LCCB index.
<i>Dump</i> =	(PSOFFSET size) - LCCB's "rctl_bfr" BD offset.

Table 54 Error Code: TPID

Error Code:	TPID
Error Description:	A timer pending invalid. The audit has found an LCCB linked to an ALCB for a VC call with a "cpstate" that requires a timer pending. No timer is indicated, so the LCCB will be recovered.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's "cpstate" state value.
<i>Bad Data =</i>	tgret.
<i>Good Data =</i>	LCCB index.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 55 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The audit found an LCCB whose modem pool call is presently being torn down. Audits will time this transient state for at least 5 minutes. No audit sees this condition 5 minutes later so the LCCB will be torn down.
Possible Error Effect:	Prints a report and recovers the LCCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LCCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PH channel number.
<i>Bad Data =</i>	LCCB's modem pool call touch bit that audit sets and will interrogate 5 minutes later to see if it is set and needs to tear the LCCB down.
<i>Good Data =</i>	Logical PH number.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 56 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit found an LCCB linked to an ALCB of a VC type but the channel type is found bad.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The call is torn down and associated resources (such as, the LCCB are returned to the appropriate idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB's logical channel number.
<i>Bad Data =</i>	LCCB's channel type - "chantype".
<i>Good Data =</i>	PSVC TYPE - valid value for virtual call (VC) type for the LCCB's channel type.
<i>Logical Index =</i>	LCCB index.
<i>Dump =</i>	NA.

Table 57 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	AUrunning, audit ID that was running at the time of the interrupt.
<i>Dump =</i>	NA.

LCCB-C

Software Release: 5E13 only

1. LOGICAL CHANNEL CONTROL BLOCK (LCCB) AUDIT

Logical channel control blocks (LCCBs, entries in `PSlccb[ ]`) are used to represent layer 3 protocol data for a single user.

2. DATA STRUCTURES

For details on the fields contained in the structures mentioned in the error descriptions below, refer to the *5ESS®* Translations Data Manual (235-600-1xx) and the *5ESS®* Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	An LCCB's <i>huntflag</i> is set incorrectly based on the layer 3 data <i>mhgflag</i> and/or the <i>lcn</i> value.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets <code>LCCB.huntflag</code> to the opposite of it's current value (either <code>PSDISABLED</code> or <code>PSENABLED</code>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	A value of "1" indicates that the <i>huntflag</i> was <code>PSENABLED</code> but the layer 3 data <i>mhgflag</i> was <code>DBNO</code> . A value of "2" indicates that <i>huntflag</i> was <code>PSDISABLED</code> but the layer 3 data <i>mhgflag</i> was <code>DBYES</code> and the <i>lcn</i> value was in-range to be hunted and either was not a member of route selection or was routable.
<i>Bad Data =</i>	<code>LCCB.huntflag</code> .
<i>Good Data =</i>	The layer 3 data <i>mhgflag</i> value.
<i>Logical Index =</i>	The LCCB index (index into <code>PSlccb[]</code>).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB (indexed by <code>LCCB.alcb_index</code>).
<i>Dump 3 =</i>	The DCDB (indexed by <code>ALCB.al_phcn</code>).
<i>Dump 4 =</i>	The DCCB (indexed by <code>ALCB.al_phcn</code>).
<i>Dump 5 =</i>	The layer 3 data structure.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An LCCB is not configured properly based on the list it is linked to. For the idle and billing list (<i>owning_ll</i> = <code>PSFREELCCB</code> or <code>PSBILLLCCB</code>) this means that unused fields are not set to the expected default values (for example, the LCCB owns BDs or timers). For LCCBs linked to a DCCB (<i>owning_ll</i> = <code>PSDCCBQ</code>) the <i>lcn</i> or <i>alcb_index</i> fields are not null. In all other cases, the LCCB is not linked to an ALCB. Either <i>alcb_index</i> or <i>lcn</i> are out of range or <code>ALCB.lccb_index[lcn]</code> does not indicate the LCCB being audited.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in

	loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.owning_ll.
<i>Bad Data =</i>	LCCB.lcn.
<i>Good Data =</i>	LCCB.alcb_index.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB if LCCB.alcb_index is in range (< PSHMAXALCB).

Table 3 Error Code: BAD_FAILED

Error Code:	BAD_FAILED
Error Description:	An LCCB is internally inconsistent (for example, the current protocol state may require a timer but no timer is present).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	Varies based on the nature of the inconsistency. Some errors are corrected in-line, others require that the LCCB be reinitialized and linked to the idle list. The value in "bad data" can be used to determine which type of recovery was done.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.lcn.
<i>Bad Data =</i>	A value indicating the nature of the error.
<i>Good Data =</i>	LCCB.alcb_index.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3 =</i>	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4 =</i>	The DCCB (indexed by ALCB.al_phcn).

Table 4 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	An entry in PSpgbstat[] contains incorrect counts of LCNs (<i>aggr_lcn</i>) and/or TPCs (<i>aggr_tpc</i>) according to the actual LCCB usage.
Possible Error Effect:	Hunting based on packet business group (PBG) usage may not be done correctly.
Corrective Action Taken:	The audit corrects the bad counts.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into PSpgbstat[].
<i>Bad Data =</i>	The <i>aggr_lcn</i> value.
<i>Good Data =</i>	The <i>chk_lcn</i> value.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The PSpgbstat[] entry (which includes the good and bad TPC counts).

Table 5 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	The LCCB.billflg != PSDISABLED for an LCN associated with an X.75' trunk (DCDB.phctype = PST_X75P).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets LCCB.billflg to PSDISABLED.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	DCDB.phctype.
<i>Bad Data</i> =	LCCB.lcn.
<i>Good Data</i> =	LCCB.alcb_index.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 6 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	The list of BDs in the circular list LCCB.hold_buff[] should not have any holes in it but it does.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index of the hole in <i>hold_buff</i> [].
<i>Bad Data</i> =	LCCB.hold_top.
<i>Good Data</i> =	LCCB.hb_qd.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 7 Error Code: CALL_ID

Error Code:	CALL_ID
Error Description:	An LCCB's <i>lcn</i> value indicates that it is a PVC but the <i>chantype</i> value != PSPVC_TYPE.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.chantype.
<i>Good Data</i> =	Layer 3 data <i>pvc_h_lcn</i> .
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).
<i>Dump 5</i> =	The layer 3 data structure.

Table 8 Error Code: CONV_BAD

Error Code:	CONV_BAD
Error Description:	An LCCB is associated with an active PBG (<i>loc_tpc_act</i> and/or <i>rem_tpc_act</i> = DBYES) but the value in LCCB.vc_tpc[PSCED or PSCING] is invalid.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such

	as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.vc_tpc[PSCED]
<i>Bad Data =</i>	LCCB.vc_tpc[PSCING]
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump =</i>	The LCCB.

Table 9 Error Code: INDEX

Error Code:	INDEX
Error Description:	Either at least one of an LCCB's <i>loc_pbg</i> or <i>rem_pbg</i> are out of range (> PSPBGMAX) or the PBG value is 0 but the corresponding TPC active bit (<i>loc_tpc_act</i> or <i>rem_tpc_act</i>) is set to DBYES.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.loc_pbg.
<i>Bad Data =</i>	LCCB.rem_pbg.
<i>Good Data =</i>	PSPBGMAX.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump =</i>	The LCCB.

Table 10 Error Code: INDRANGE_BAD

Error Code:	INDRANGE_BAD
Error Description:	An LCCB is linked to an ALCB (using <i>alcb_index</i> and <i>lcn</i>) but the ALCB contains invalid indexes to write protected data such as the DCDB and ALDB (ALCB.al_phcn and ALCB.aldb_index).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.lcn.
<i>Bad Data =</i>	ALCB.al_phcn.
<i>Good Data =</i>	LCCB.alcb_index.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The associated ALCB.

Table 11 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	An LCCB headcell (element of PSlccbhd[]) is not empty (at least one of <i>first</i> or <i>last</i> is not PSLCCBNUL) but at least one of <i>first</i> or <i>last</i> is not a valid LCCB index ($\geq$ PSHMAXLCCB).
Possible Error Effect:	Varies depending on the list. If it is the idle list, LCCBs cannot be allocated (leading to layer 3 setup failures). If it is the billing list, packet billing data may be lost.
Corrective Action Taken:	The audit rebuilds the list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The headcell's <i>first</i> index.

<i>Bad Data</i> =	The headcell's <i>last</i> index.
<i>Good Data</i> =	The D-channel value associated with this list (as would appear in LCCB.qdccb_own if applicable).
<i>Logical Index</i> =	Index into the headcell array.
<i>Dump</i> =	NA.

Table 12 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An LCCB is marked as huntable (LCCB.huntflag = PSENABLED) on X.75 or X.75 prime trunk but the LCCB.lcn indicates that it is not huntable.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets LCCB.huntflag to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.huntflag.
<i>Good Data</i> =	DCDB.phctype.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).
<i>Dump 5</i> =	The layer 3 data structure.

Table 13 Error Code: LIST

Error Code:	LIST
Error Description:	Either not all of the entries in LCCB.retran_buff[] starting at LCCB.retran_top and going for LCCB.rtb_qd+LCCB.shold_qd are set or some entries outside that range are set.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index of the bad value in LCCB.retran_buff[].
<i>Bad Data</i> =	LCCB.retran_top.
<i>Good Data</i> =	LCCB.rtb_qd+LCCB.shold_qd.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 14 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	An LCCB headcell contains the index of an LCCB that is not owned by that headcell or whose <i>bklccb_ind</i> or <i>fdlccb_ind</i> field is not consistent with the LCCB being at the end of the list.
Possible Error Effect:	Varies depending on the list. If it is the idle list, LCCBs cannot be allocated (leading to layer 3 setup failures). If it is the billing list, packet billing data may be lost.
Corrective Action Taken:	The audit rebuilds the list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The headcell's <i>first</i> index.
<i>Bad Data</i> =	The headcell's <i>last</i> index.

<i>Good Data</i> =	The D-channel value associated with this list (as would appear in LCCB.qdccb_own if applicable).
<i>Logical Index</i> =	Index into the headcell array.
<i>Dump 1</i> =	The LCCB indexed by the headcell's <i>first</i> value.
<i>Dump 2</i> =	The LCCB indexed by the headcell's <i>last</i> value.

Table 15 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	An LCCB is not properly linked in a list or linked to no list. Either <i>fdlccb_ind</i> or <i>bklccb_ind</i> is out of range ($> \text{PSHMAXLCCB}$ and $\neq \text{PSLCCBNULL}$) or is null but the LCCB is not at the end of the list according to the headcell or indexes an LCCB which either does not point back (using <i>bklccb_ind</i> or <i>fdlccb_ind</i> respectively) or which contains different owner information (in <i>owning_ll</i> or <i>qdccb_own</i>).
Possible Error Effect:	If the owning list is the LCCB idle list, LCCBs cannot be allocated. Otherwise, a set of LCCBs cannot be allocated which could lead to layer 3 setup failures.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.owning_ll.
<i>Bad Data</i> =	LCCB headcell <i>first</i> LCCB index.
<i>Good Data</i> =	LCCB headcell <i>last</i> LCCB index.
<i>Logical Index</i> =	The LCCB index (index into <i>PSlccb[]</i>).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The LCCB indexed by <i>fdlccb_ind</i> if it's in range ($< \text{PSHMAXLCCB}$).
<i>Dump 3</i> =	The LCCB indexed by <i>bklccb_ind</i> if it's in range ($< \text{PSHMAXLCCB}$).
<i>Dump 4</i> =	The LCCB headcell's <i>first</i> LCCB if it's in range ($< \text{PSHMAXLCCB}$).
<i>Dump 5</i> =	The LCCB headcell's <i>last</i> LCCB if it's in range ($< \text{PSHMAXLCCB}$).

Table 16 Error Code: LS_RANGE_BAD

Error Code:	LS_RANGE_BAD
Error Description:	Either LCCB.retran_top is out of range ($\geq \text{PSBFRIND}$) or LCCB.rtb_qd+LCCB.shold_qd is $> \text{PSBFRIND}$.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.retran_top.
<i>Bad Data</i> =	LCCB.rtb_qd.
<i>Good Data</i> =	LCCB.shold_qd.
<i>Logical Index</i> =	The LCCB index (index into <i>PSlccb[]</i>).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 17 Error Code: OWNER

Error Code:	OWNER
Error Description:	An LCCB contains an invalid value in LCCB.owning_ll or checks specific to the owning linked list value failed. Specifically, if the <i>owning_ll</i> value is PSNOLCCBLST, one or more of the associated linkage fields (<i>fdlccb_ind</i> , <i>bklccb_ind</i> or <i>qdccb_own</i>) are not set to the appropriate default value. In the case of value PSCCBQ, the <i>qdccb_own</i> field is out of range ($\geq \text{PSMAXCHN}$).

Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.owning_ll.
<i>Bad Data =</i>	LCCB.fdlccb_ind.
<i>Good Data =</i>	LCCB.bklccb_ind.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	If the headcell index is in range, the headcell.

Table 18 Error Code: QUAR

Error Code:	QUAR
Error Description:	An LCCB's quarantine field has been set (LCCB.quar_bit = DBYES) indicating that the data in it is invalid.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.owning_ll.
<i>Bad Data =</i>	LCCB.alcb_index.
<i>Good Data =</i>	LCCB.lcn.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump =</i>	The LCCB.

Table 19 Error Code: QUE

Error Code:	QUE
Error Description:	One or both of LCCB.hold_top and LCCB.hb_qd are of out of range ($\geq$ PSBFRIND or $>$ PSBFRIND respectively).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.hold_top.
<i>Bad Data =</i>	LCCB.hb_qd.
<i>Good Data =</i>	PSBFRIND.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3 =</i>	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4 =</i>	The DCCB (indexed by ALCB.al_phcn).

Table 20 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	An LCCB contains the index of a timer (in LCCB.tim[] or LCCB.timacc[]) but the timer is not linked back.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets the timer index field to PSTNULLTE.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	A value indicating the subarray of timer elements in the LCCB.
<i>Bad Data</i> =	A value indicating the nature of the bad linkage.
<i>Good Data</i> =	Index into the timer subarray.
<i>Logical Index</i> =	Index of the timer element.
<i>Dump</i> =	The LCCB.

Table 21 Error Code: TRANS_FAILED

Error Code:	TRANS_FAILED
Error Description:	An LCCB has been marked for modem pool teardown (LCCB.mcp_tear = DBYES) for more than 5 minutes.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.alcb_index.
<i>Good Data</i> =	ALCB.al_phcn.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 22 Error Code: TYPE

Error Code:	TYPE
Error Description:	Based on the PH channel type (from DCDB.phctype where the DCDB is indexed by ALCB.al_phcn), an LCCB has an invalid <i>chantype</i> or other data (such as <i>mptmbd</i> != PSNULLOFF for phctype = PSDP_MP or ALCB.aldb_index = PSALDBNULL for non-trunk, non-modem pool cases).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.chantype.
<i>Good Data</i> =	DCDB.phctype.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 23 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	If the LCCB headcell is in range (< PSNUMLCCBHD), the audit will rebuild that list of LCCBs. If the LCCB index is in range (< PSHMAXLCCB), the audit unlinks the LCCB

	from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSNUMLCCBHD.
<i>Bad Data</i> =	Index into LCCB headcell array (PSrlccbhd[]). May be out of range.
<i>Good Data</i> =	PSHMAXLCCB.
<i>Logical Index</i> =	The LCCB index. Note that this value may be out of range ($\geq$ PSHMAXLCCB).
<i>Dump 1</i> =	The LCCB headcell if the index is in range ($<$ PSNUMLCCBHD), otherwise NA.
<i>Dump 2</i> =	The LCCB if the LCCB index is in range ($<$ PSHMAXLCCB), otherwise NA.

LCCB-D

Software Release: 5E14 and later

1. LOGICAL CHANNEL CONTROL BLOCK (LCCB) AUDIT

Logical channel control blocks (LCCBs, entries in PSlccb[]) are used to represent layer 3 protocol data for a single user.

2. DATA STRUCTURES

For details on the fields contained in the structures mentioned in the error descriptions below, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	An LCCB's <i>huntflag</i> is set incorrectly based on the layer 3 data <i>mhgflg</i> and/or the <i>lcn</i> value.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets LCCB.huntflag to the opposite of it's current value (either PSDISABLED or PSENABLED).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	A value of "1" indicates that the <i>huntflag</i> was PSENABLED but the layer 3 data <i>mhgflg</i> was DBNO. A value of "2" indicates that <i>huntflag</i> was PSDISABLED but the layer 3 data <i>mhgflg</i> was DBYES and the <i>lcn</i> value was in-range to be hunted and either was not a member of route selection or was routable.
<i>Bad Data =</i>	LCCB.huntflag.
<i>Good Data =</i>	The layer 3 data <i>mhgflg</i> value.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3 =</i>	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4 =</i>	The DCCB (indexed by ALCB.al_phcn).
<i>Dump 5 =</i>	The layer 3 data structure.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An LCCB is not configured properly based on the list it is linked to. For the idle and billing list (<i>owning_ll</i> = PSFREEELCCB or PSBILLLCCB) this means that unused fields are not set to the expected default values (for example, the LCCB owns BDs or timers). For LCCBs linked to a DCCB (<i>owning_ll</i> = PSDCCBQ) the <i>lcn</i> or <i>alcb_index</i> fields are not null. In all other cases, the LCCB is not linked to an ALCB. Either <i>alcb_index</i> or <i>lcn</i> are out of range or ALCB.lccb_index[lcn] does not indicate the LCCB being audited.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in

	loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.owning_ll.
<i>Bad Data =</i>	LCCB.lcn.
<i>Good Data =</i>	LCCB.alcb_index.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB if LCCB.alcb_index is in range (< PSHMAXALCB).

Table 3 Error Code: BAD_FAILED

Error Code:	BAD_FAILED
Error Description:	An LCCB is internally inconsistent (for example, the current protocol state may require a timer but no timer is present).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	Varies based on the nature of the inconsistency. Some errors are corrected in-line, others require that the LCCB be reinitialized and linked to the idle list. The value in "bad data" can be used to determine which type of recovery was done.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.lcn.
<i>Bad Data =</i>	A value indicating the nature of the error.
<i>Good Data =</i>	LCCB.alcb_index.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3 =</i>	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4 =</i>	The DCCB (indexed by ALCB.al_phcn).

Table 4 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	An entry in PSpgbstat[] contains incorrect counts of LCNs (<i>aggr_lcn</i>) and/or TPCs (<i>aggr_tpc</i>) according to the actual LCCB usage.
Possible Error Effect:	Hunting based on packet business group (PBG) usage may not be done correctly.
Corrective Action Taken:	The audit corrects the bad counts.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into PSpgbstat[].
<i>Bad Data =</i>	The <i>aggr_lcn</i> value.
<i>Good Data =</i>	The <i>chk_lcn</i> value.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The PSpgbstat[] entry (which includes the good and bad TPC counts).

Table 5 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	The LCCB.billflg != PSDISABLED for an LCN associated with an X.75' trunk (DCDB.phctype = PST_X75P) or internal LCN for INET.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets LCCB.billflg to PSDISABLED.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	DCDB.phctype.
<i>Bad Data</i> =	LCCB.lcn.
<i>Good Data</i> =	LCCB.alcb_index.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 6 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	The list of BDs in the circular list LCCB.hold_buff[] should not have any holes in it but it does.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index of the hole in <i>hold_buff</i> [].
<i>Bad Data</i> =	LCCB.hold_top.
<i>Good Data</i> =	LCCB.hb_qd.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 7 Error Code: CALL_ID

Error Code:	CALL_ID
Error Description:	An LCCB's <i>lcn</i> value indicates that it is a PVC but the <i>chantype</i> value != PSPVC_TYPE.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.chantype.
<i>Good Data</i> =	Layer 3 data <i>pvc_h_lcn</i> .
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).
<i>Dump 5</i> =	The layer 3 data structure.

Table 8 Error Code: CONV_BAD

Error Code:	CONV_BAD
Error Description:	An LCCB is associated with an active PBG (<i>loc_tpc_act</i> and/or <i>rem_tpc_act</i> = DBYES) but the value in LCCB.vc_tpc[PSCED or PSCING] is invalid.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such

	as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.vc_tpc[PSCED]
<i>Bad Data =</i>	LCCB.vc_tpc[PSCING]
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump =</i>	The LCCB.

Table 9 Error Code: INDEX

Error Code:	INDEX
Error Description:	Either at least one of an LCCB's <i>loc_pbg</i> or <i>rem_pbg</i> are out of range (> PSPBGMAX) or the PBG value is 0 but the corresponding TPC active bit (<i>loc_tpc_act</i> or <i>rem_tpc_act</i>) is set to DBYES.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.loc_pbg.
<i>Bad Data =</i>	LCCB.rem_pbg.
<i>Good Data =</i>	PSPBGMAX.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump =</i>	The LCCB.

Table 10 Error Code: INDRANGE_BAD

Error Code:	INDRANGE_BAD
Error Description:	An LCCB is linked to an ALCB (using <i>alcb_index</i> and <i>lcn</i>) but the ALCB contains invalid indexes to write protected data such as the DCDB and ALDB (ALCB.al_phcn and ALCB.aldb_index).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LCCB.lcn.
<i>Bad Data =</i>	ALCB.al_phcn.
<i>Good Data =</i>	LCCB.alcb_index.
<i>Logical Index =</i>	The LCCB index (index into PSlccb[]).
<i>Dump 1 =</i>	The LCCB.
<i>Dump 2 =</i>	The associated ALCB.

Table 11 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	An LCCB headcell (element of PSlccbhd[]) is not empty (at least one of <i>first</i> or <i>last</i> is not PSLCCBNUL) but at least one of <i>first</i> or <i>last</i> is not a valid LCCB index ($\geq$ PSHMAXLCCB).
Possible Error Effect:	Varies depending on the list. If it is the idle list, LCCBs cannot be allocated (leading to layer 3 setup failures). If it is the billing list, packet billing data may be lost.
Corrective Action Taken:	The audit rebuilds the list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The headcell's <i>first</i> index.

<i>Bad Data</i> =	The headcell's <i>last</i> index.
<i>Good Data</i> =	The D-channel value associated with this list (as would appear in LCCB.qdccb_ own if applicable).
<i>Logical Index</i> =	Index into the headcell array.
<i>Dump</i> =	NA.

Table 12 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An LCCB is marked as huntable (LCCB.huntflag = PSENABLED) on X.75 or X.75 prime trunk but the LCCB.lcn indicates that it is not huntable.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets LCCB.huntflag to PSDISABLED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.huntflag.
<i>Good Data</i> =	DCDB.phctype.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).
<i>Dump 5</i> =	The layer 3 data structure.

Table 13 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	Multiple line hunt group (MLHG) flag is set to DBYES for internal INET LCN.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets PSL3DATA.mhgflg to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.huntflag.
<i>Good Data</i> =	PSL3DATA.mhgflg.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).
<i>Dump 5</i> =	The layer 3 data structure.

Table 14 Error Code: LIST

Error Code:	LIST
Error Description:	Either not all of the entries in LCCB.retran_buf[] starting at LCCB.retran_top and going for LCCB.rtb_qd+LCCB.shold_qd are set or some entries outside that range are set.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index of the bad value in LCCB.retran_buf[].
<i>Bad Data</i> =	LCCB.retran_top.
<i>Good Data</i> =	LCCB.rtb_qd+LCCB.shold_qd.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).

<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 15 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	An LCCB headcell contains the index of an LCCB that is not owned by that headcell or whose <i>bkcccb_ind</i> or <i>fdcccb_ind</i> field is not consistent with the LCCB being at the end of the list.
Possible Error Effect:	Varies depending on the list. If it is the idle list, LCCBs cannot be allocated (leading to layer 3 setup failures). If it is the billing list, packet billing data may be lost.
Corrective Action Taken:	The audit rebuilds the list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The headcell's <i>first</i> index.
<i>Bad Data</i> =	The headcell's <i>last</i> index.
<i>Good Data</i> =	The D-channel value associated with this list (as would appear in LCCB.qdcccb_own if applicable).
<i>Logical Index</i> =	Index into the headcell array.
<i>Dump 1</i> =	The LCCB indexed by the headcell's <i>first</i> value.
<i>Dump 2</i> =	The LCCB indexed by the headcell's <i>last</i> value.

Table 16 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	An LCCB is not properly linked in a list or linked to no list. Either <i>fdcccb_ind</i> or <i>bkcccb_ind</i> is out of range ($> \text{PSHMAXLCCB}$ and $\neq \text{PSLCCBNULL}$) or is null but the LCCB is not at the end of the list according to the headcell or indexes an LCCB which either does not point back (using <i>bkcccb_ind</i> or <i>fdcccb_ind</i> respectively) or which contains different owner information (in <i>owning_ll</i> or <i>qdcccb_own</i>).
Possible Error Effect:	If the owning list is the LCCB idle list, LCCBs cannot be allocated. Otherwise, a set of LCCBs cannot be allocated which could lead to layer 3 setup failures.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.owning_ll.
<i>Bad Data</i> =	LCCB headcell <i>first</i> LCCB index.
<i>Good Data</i> =	LCCB headcell <i>last</i> LCCB index.
<i>Logical Index</i> =	The LCCB index (index into <i>PSlcccb[]</i>).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The LCCB indexed by <i>fdcccb_ind</i> if it's in range ($< \text{PSHMAXLCCB}$).
<i>Dump 3</i> =	The LCCB indexed by <i>bkcccb_ind</i> if it's in range ($< \text{PSHMAXLCCB}$).
<i>Dump 4</i> =	The LCCB headcell's <i>first</i> LCCB if it's in range ($< \text{PSHMAXLCCB}$).
<i>Dump 5</i> =	The LCCB headcell's <i>last</i> LCCB if it's in range ($< \text{PSHMAXLCCB}$).

Table 17 Error Code: LS_RANGE_BAD

Error Code:	LS_RANGE_BAD
Error Description:	Either LCCB.retran_top is out of range ($\geq \text{PSBFRIND}$) or LCCB.rtb_qd+LCCB.shold_qd is $> \text{PSBFRIND}$.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	LCCB.retran_top.
<i>Bad Data</i> =	LCCB.rtb_qd.
<i>Good Data</i> =	LCCB.shold_qd.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 18 Error Code: OWNER

Error Code:	OWNER
Error Description:	An LCCB contains an invalid value in LCCB.owning_ll or checks specific to the owning linked list value failed. Specifically, if the <i>owning_ll</i> value is PSNOLCCBLST, one or more of the associated linkage fields (<i>fdlccb_ind</i> , <i>bklccb_ind</i> or <i>qdccb_own</i>) are not set to the appropriate default value. In the case of value PSCCBQ, the <i>qdccb_own</i> field is out of range ($\geq$ PSMAXCHN).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.owning_ll.
<i>Bad Data</i> =	LCCB.fdlccb_ind.
<i>Good Data</i> =	LCCB.bklccb_ind.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	If the headcell index is in range, the headcell.

Table 19 Error Code: QUAR

Error Code:	QUAR
Error Description:	An LCCB's quarantine field has been set (LCCB.quar_bit = DBYES) indicating that the data in it is invalid.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.owning_ll.
<i>Bad Data</i> =	LCCB.alcb_index.
<i>Good Data</i> =	LCCB.lcn.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump</i> =	The LCCB.

Table 20 Error Code: QUE

Error Code:	QUE
Error Description:	One or both of LCCB.hold_top and LCCB.hb_qd are out of range ($\geq$ PSBFRIND or $>$ PSBFRIND respectively).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	LCCB.hold_top.
<i>Bad Data</i> =	LCCB.hb_qd.
<i>Good Data</i> =	PSBFRIND.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 21 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	An LCCB contains the index of a timer (in LCCB.tim[] or LCCB.timacc[]) but the timer is not linked back.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit sets the timer index field to PSTNULLTE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A value indicating the subarray of timer elements in the LCCB.
<i>Bad Data</i> =	A value indicating the nature of the bad linkage.
<i>Good Data</i> =	Index into the timer subarray.
<i>Logical Index</i> =	Index of the timer element.
<i>Dump</i> =	The LCCB.

Table 22 Error Code: TRANS_FAILED

Error Code:	TRANS_FAILED
Error Description:	An LCCB has been marked for modem pool teardown (LCCB.mcp_tear = DBYES) for more than 5 minutes.
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.
<i>Bad Data</i> =	LCCB.alcb_index.
<i>Good Data</i> =	ALCB.al_phcn.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 23 Error Code: TYPE

Error Code:	TYPE
Error Description:	Based on the PH channel type (from DCDB.phctype where the DCDB is indexed by ALCB.al_phcn), an LCCB has an invalid <i>chantype</i> or other data (such as <i>mptmbd</i> != PSNULLOFF for phctype = PSDP_MP or ALCB.aldb_index = PSALDBNULL for non-trunk, non-modem pool cases).
Possible Error Effect:	The protocol stack associated with the LCCB is corrupted. This may result in loss of layer 3, dropped frames or other problems varying according to conditions.
Corrective Action Taken:	The audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LCCB.lcn.

<i>Bad Data</i> =	LCCB.chantype.
<i>Good Data</i> =	DCDB.phctype.
<i>Logical Index</i> =	The LCCB index (index into PSlccb[]).
<i>Dump 1</i> =	The LCCB.
<i>Dump 2</i> =	The ALCB (indexed by LCCB.alcb_index).
<i>Dump 3</i> =	The DCDB (indexed by ALCB.al_phcn).
<i>Dump 4</i> =	The DCCB (indexed by ALCB.al_phcn).

Table 24 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	If the LCCB headcell is in range (< PSNUMLCCBHD), the audit will rebuild that list of LCCBs. If the LCCB index is in range (< PSHMAXLCCB), the audit unlinks the LCCB from any list it is in and from any owning ALCB, sets all fields to default values and links it to the idle list. Associated resources such as timers may also be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSNUMLCCBHD.
<i>Bad Data</i> =	Index into LCCB headcell array (PSrlccbhd[]). May be out of range.
<i>Good Data</i> =	PSHMAXLCCB.
<i>Logical Index</i> =	The LCCB index. Note that this value may be out of range ($\geq$ PSHMAXLCCB).
<i>Dump 1</i> =	The LCCB headcell if the index is in range (< PSNUMLCCBHD), otherwise NA.
<i>Dump 2</i> =	The LCCB if the LCCB index is in range (< PSHMAXLCCB), otherwise NA.

LCGREP

Software Release: 5E10 and later

1. LINE CONCENTRATOR REPORT INHIBIT MASK (LCGREP) AUDIT

Peripheral control (PC) scanning detects input signals from the switching periphery in the switching module (SM). These input signals include line and trunk originations, digit collection, service request signals, and other supervision signals. Two types of scanning are audited: control interface (CI) scanning and summary scanning. Control interface scanning is described in the scan mask audit (SCNMSK). Summary scanning detects input signals from line concentrators (grids) connected to the line unit or line group controllers (LGCs) for integrated services line units (ISLUs). Refer to Figure 1. The line unit contains from four to eight grid circuits for LU1, 16 half-grids for LU2, and 20 for LU3. Each circuit controls a maximum of 64 customer lines (32 for LU2, and LU3). Scanning detects when a customer's line goes off-hook.

The line concentrator report inhibit mask audit (LCGREP) checks summary scanning and ensures that line originations are not being ignored. The line concentrator scan mask audit (LCSMSK) also checks summary scanning.

The audit detects errors in the line concentrator report inhibit mask array (Cllcgrep). Each bit in Cllcgrep corresponds to a group of eight lines in a grid. If a bit is set, a report for that group is entered on the report queue but has not been completely processed. This prevents queuing other reports for the same group. A single bit error in Cllcgrep may cause origination failures (no dial tone) for all eight lines in the affected group.

This audit is part of the peripheral control audits subgroup.

2. DATA STRUCTURES

The data structures verified by this audit are the line concentrator report inhibit mask array (Cllcgrep), the group service data block relation (rlGSDB), and the line concentrator touch bit array (Sllcgtch).

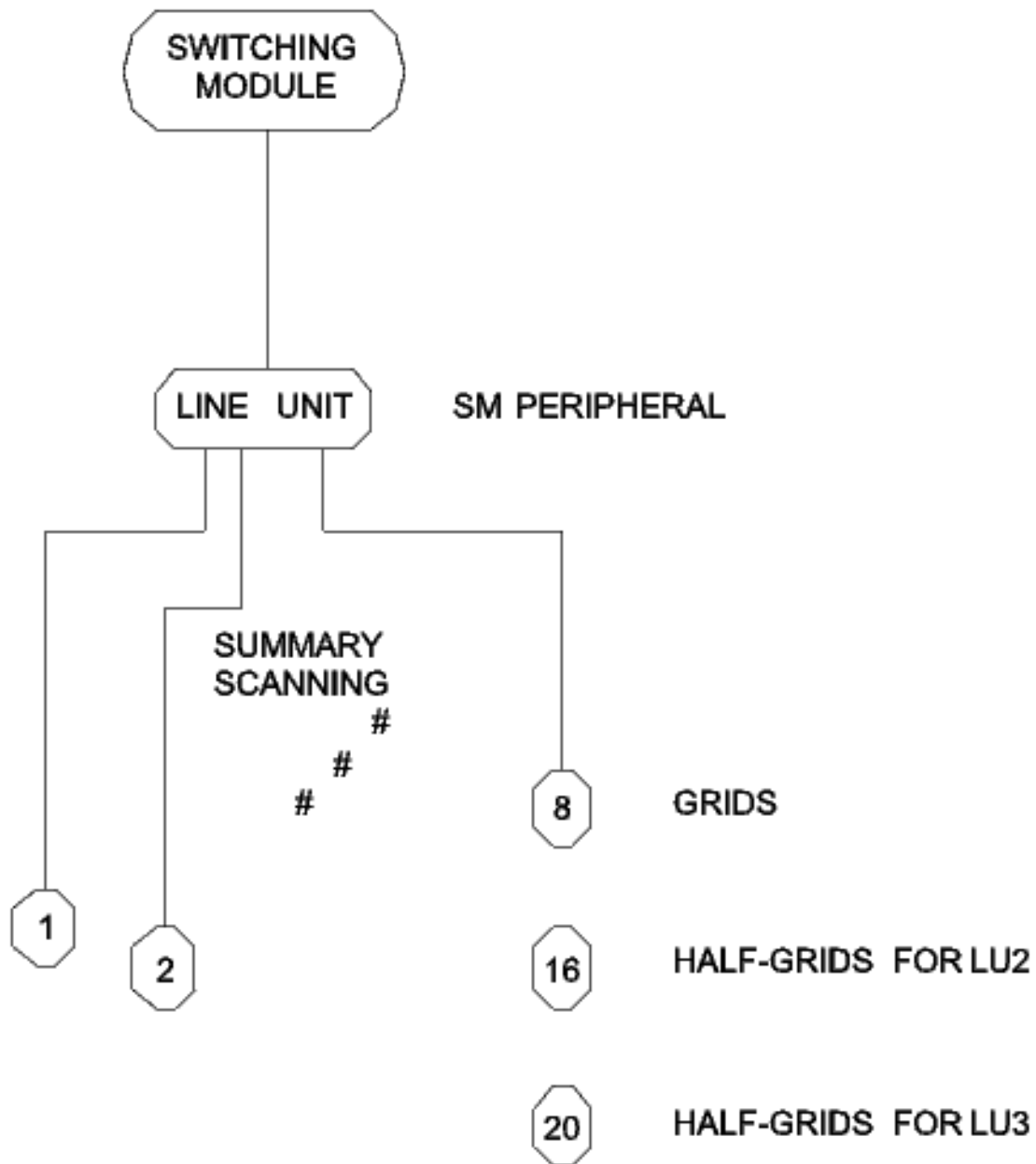


Figure 1 (LCGREP-1) Summary Scanning

3. AUDIT CHECKS

The LCGREP audit steps through the Cllcgrep array. Using line unit and grid information, the audit calculates a key and retrieves the rIGSDB tuple with that key. The information in the tuple, along with the information in the Sllcgtch array, is used to audit the Cllcgrep array.

- (a) The audit compares each bit in Cllcgrep against each bit in Sllcgtch for one of four conditions:
- Both bits are not set. This is the desired condition because it enables reporting.
 - The bit in Cllcgrep is not set, and the bit in Sllcgtch is set. This condition is an error.

- The bit in Cllcgrep is set, and the bit in Sllcgtch is not set. This condition is temporary and will be changed by setting the Sllcgtch bit.
- Both bits are set. In this case, the corresponding rIGSDB tuple is checked to see whether it is on an idle list. If the rIGSDB tuple is on an idle list, this condition is an error because reports have been inhibited for at least 30 seconds and there is no queued job to allow further reports.

- (b) The Cllcgrep and Sllcgtch arrays allow the audit to infer activity. If there is no activity for 30 seconds and no queued report, the audit concludes that errors occurred and zeroes the Cllcgrep and Sllcgtch bits.

The Cllcgrep array is not used by integrated services line unit (ISLU) and remote ISLU (RISLU) operational software. Consequently, any entries in the arrays corresponding to these unit types are not audited.

4. ERROR HANDLING

If any of the conditions are in error, bits in the Cllcgrep and Sllcgtch arrays are zeroed.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: INHHDW

Error Code:	INHHDW
Error Description:	Status of the group in hardware does not match the software status.
Possible Error Effect:	The tuple associated with this bit will be invalidly used. consequently cause unnecessary ROP output.
Corrective Action Taken:	The hardware will be overwritten to match the software.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Line unit number.
<i>Bad Data =</i>	Data from hardware read.
<i>Good Data =</i>	Value of corresponding Cllcgrep[] element.
<i>Logical Index =</i>	The value of our loop cursor.
<i>Dump =</i>	NA.

Table 2 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	The audited bit is not a valid bit yet it's "bitval" is not 0.
Possible Error Effect:	The tuple associated with this bit will be invalidly used. consequently cause unnecessary ROP output.
Corrective Action Taken:	The bit will be set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Indication of bit validity.
<i>Bad Data =</i>	The Cllcgrep[element] value.
<i>Good Data =</i>	The Cllcgrep[element] value after it is corrected.
<i>Logical Index =</i>	The value of AUCursor.
<i>Dump =</i>	NA.

Table 3 Error Code: RISTUK

Error Code:	RISTUK
Error Description:	Bits in Cllcgrep are stuck set, inhibiting origination reports.
Possible Error Effect:	Affected customers cannot originate.
Corrective Action Taken:	The erroneous bit(s) are cleared to allow originations to be reported.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the element in Cllcgreg.
<i>Bad Data</i> =	Value of bitval.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Number of the affected group.
<i>Dump</i> =	riGSDB tuple corresponding to the bit in error.

Table 4 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AULcgtim, used for transient timing may have occurred.
Possible Error Effect:	The will cause the LCGREP audit not to run on this invocation of the audit. The LCGREP audit should not run faster than every 30 seconds.
Corrective Action Taken:	AULcgtim will be updated with the current OSDS ITIME (OSnow) and the audit will not run until the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the transient timing variable (AULcgtim).
<i>Bad Data</i> =	The difference in time between current ITIME (OSnow) and the transient timing variable (AULcgtim).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Content of the current OSDS ITIME (OSnow).
<i>Dump 2</i> =	Content of the transient timing variable (AULcgtim).

Table 5 Error Code: TMUT

Error Code:	TMUT
Error Description:	The touch bits have been mutilated. Bits in Sllcgtch were found set with the corresponding bits in Cllcgreg found cleared.
Possible Error Effect:	The LCGREP audit will not operate correctly.
Corrective Action Taken:	The mutilated touch bits are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element in Sllcgtch.
<i>Bad Data</i> =	Value of touchbit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Number of the affected group.
<i>Dump</i> =	NA.

Table 6 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Line unit number (0-7).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	NA.

LCSMSK

Software Release: 5E10 and later

1. LINE CONCENTRATOR SCAN MASK AUDIT (LCSMSK)

Peripheral control (PC) scanning detects input signals from the switching periphery in the switching module (SM). These input signals include line and trunk originations, digit collection, service request signals, and other supervision signals. Two types of scanning are audited: control interface (CI) scanning and summary scanning. Control interface scanning is described in the scan mask audit (SCNMSK). Summary scanning detects input signals from line concentrators (grids) connected to the line unit or line group controllers (LGCs) for integrated services line units (ISLUs) and remote ISLUs (RISLU). Refer to Figure 1. The line unit contains from four to eight grids for LU1 (16 half-grids for LU2, 20 for LU3); each grid controls a maximum of 64 customer lines (32 for LU2 and LU3). Scanning detects when a customer's line goes off-hook. The line concentrator scan mask audit (LCSMSK) and the line concentrator report inhibit mask audit (LCGREP) check summary scanning.

The LCSMSK audit detects errors in the line concentrator scan mask array (Cllcsmask). Each bit in Cllcsmask corresponds to a circuit that controls 64 customer lines for LU1 (32 for LU2, LU3, ISLU and RISLU). When a bit in Cllcsmask enables scanning, the corresponding circuit is scanned for changes. A single bit error in Cllcsmask may cause call origination failures (no dial tone) for all lines terminating at the affected circuit.

This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The data structures verified by this audit are the line concentrator scan mask array (Cllcsmask), the switch maintenance equipment status table relation (RLSMEST), and the line concentrator scan mask copy (Sllcsmask).

The audit checks the ISLU/RISLU common controller (CC) duplex failure array (SMccdupfail) and the ISLU common data (CD) configuration array (PCiscdconfig).

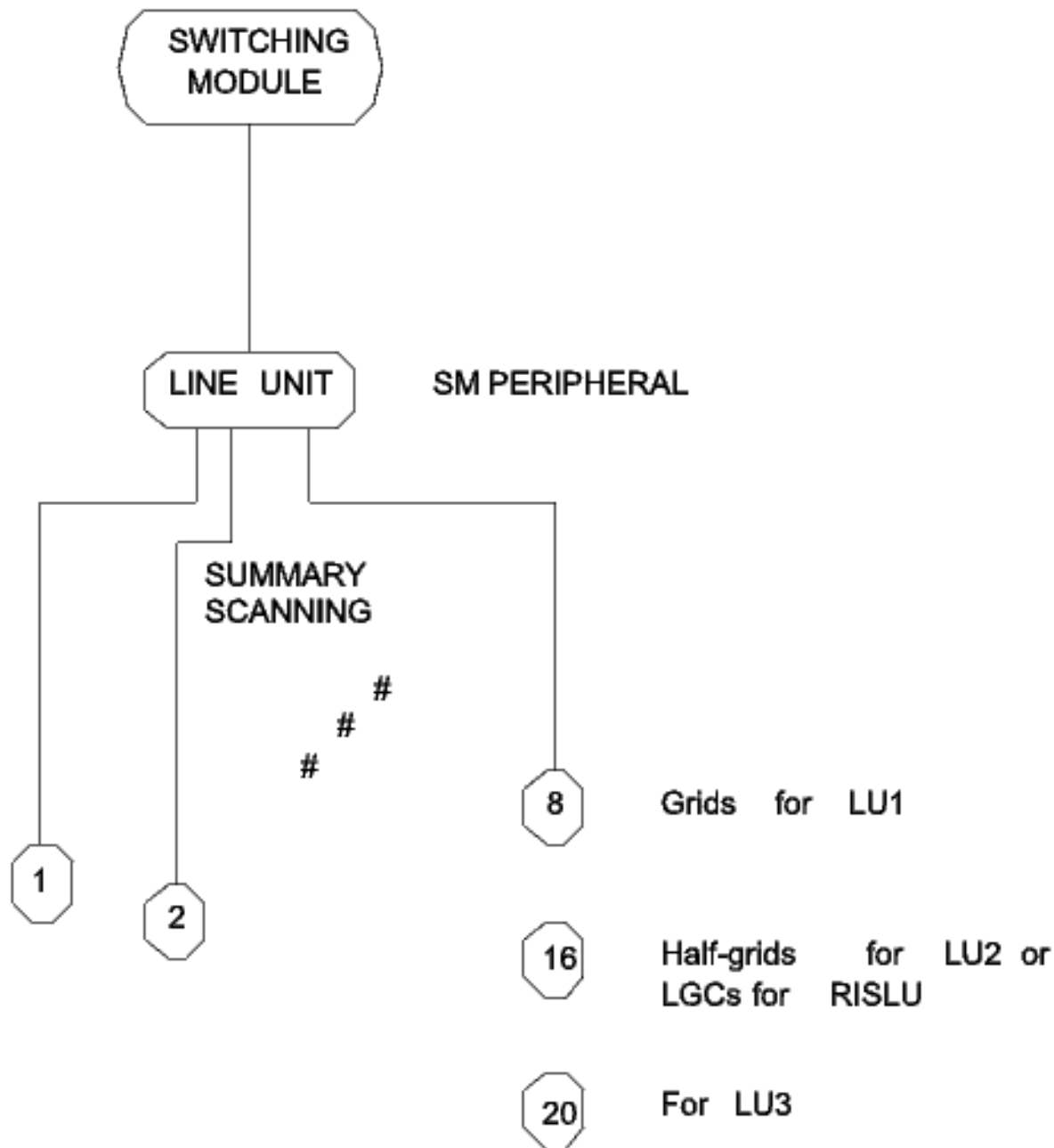


Figure 1 (LCSMSK-1) Summary Scanning

3. AUDIT CHECKS

LCSMSK audit calls the bit mask driver (DRBMSK) to step through the Cllcsmk array. The DRBMSK driver calculates the corresponding GRID/LGC circuit key, retrieves the rSMEST tuple with that key, and calls a semantic check function to perform the following checks for each bit in Cllcsmk:

- The audit uses the information in the rSMEST tuple to predict the state of the Cllcsmk bit.
- The audit verifies the predicted state of the Cllcsmk bit.
- The audit verifies each bit in Sllcsmk against Cllcsmk to find inconsistencies.

- (d) If the line unit is unequipped, the corresponding bits in Cllcsmsk[] are checked to make sure they are set to "inhibit".
- (e) For ISLU/RISLU, if the CC is duplex failed (SMccdupfail[]) or the CC is being restored (PCiscdconfig[]), the audit skips further checking on this line unit. Otherwise, the unified control interface (UCI) hardware register in the ISLU common controller (CC) is checked for agreement with the Cllcsmsk[] array and rISMEST for the ISLU LGCs in-service/OOS status.

4. ERROR HANDLING

If an inconsistency exists between Cllcsmsk and rISMEST, both Cllcsmsk and Sllcsmsk are changed to agree with rISMEST. If an inconsistency exists between Cllcsmsk and Sllcsmsk, Sllcsmsk is changed to agree with Cllcsmsk.

If the UCI IS/OOS bit disagrees with the rISMEST and/or Cllcsmsk[] for ISLU/RISLU LGC circuits, the audit will write the UCI hardware bit to agree with rISMEST and update the Cllcsmsk[] bit. If the Cllcsmsk[] bits for an unequipped LU are not marked "inhibit," the audit will update them.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The LGC UCI hardware register bit indicates that the LGC in question is active when the rISMEST tuple for the LGC indicates OOS or ACTIVE/CAMPON.
Possible Error Effect:	Operational and maintenance code will access this LGC improperly and report errors.
Corrective Action Taken:	The LGC UCI bit is written to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LGC UCI register.
<i>Bad Data</i> =	Data in LGC UCI register.
<i>Good Data</i> =	Corrected data in LGC UCI register.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	Copy of corresponding rISMEST tuple.

Table 2 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The rISMEST tuple for an RISLU/ISLU CC circuit calculated for this equipped RISLU/ISLU was not found (no match from DB read).
Possible Error Effect:	The system has lost resources, data problems.
Corrective Action Taken:	The CKTDATA audit is scheduled in elevated mode (related audit) to insert the missing rISMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of Cllcsmsk[].
<i>Bad Data</i> =	Circuit name of RISLU/ISLU CC.
<i>Good Data</i> =	Line unit number.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	NA.

Table 3 Error Code: SALW (1)

Error Code:	SALW (1)
Error Description:	A bit in Cllcsmsk[] says that scanning is allowed for a grid when the maintenance information in the corresponding rISMEST tuple says scanning

	should not be allowed.
Possible Error Effect:	PC foreground is scanning a grid that it shouldn't. This causes extra work for the system and may result in CI interrupts. These in turn could cause system initializations.
Corrective Action Taken:	The incorrect bit in Clcsmask[] is reset to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clcsmask[] containing error.
<i>Bad Data</i> =	Value of element of Clcsmask containing error.
<i>Good Data</i> =	Value of element of Clcsmask with the single bit in error corrected.
<i>Logical Index</i> =	Number of bit in Clcsmask[].
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 4 Error Code: SALW (2)

Error Code:	SALW (2)
Error Description:	For an ISLU LGC, the LGC UCI hardware bit does not indicate active (1) for an active LGC so marked in rSMEST.
Possible Error Effect:	The hardware equipage status for the LGC is wrong and will cause operational and maintenance failures.
Corrective Action Taken:	The LGC UCI hardware bit is written to active (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LGC UCI hardware register.
<i>Bad Data</i> =	Data in LGC UCI hardware register.
<i>Good Data</i> =	Corrected data in LGC UCI hardware register.
<i>Logical Index</i> =	Number of bit in Clcsmask[].
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 5 Error Code: SICPY

Error Code:	SICPY
Error Description:	The corresponding bits in Slcsmask[] and Clcsmask[] are unequal, but the bit in Clcsmask[] apparently has the right value. The SI copy of Clcsmask[] is incorrect.
Possible Error Effect:	The integrity monitor will assert periodically (and frequently).
Corrective Action Taken:	The SI copy is corrected to correspond with Clcsmask[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of Slcsmask[].
<i>Bad Data</i> =	Value of element of Slcsmask[] containing error.
<i>Good Data</i> =	Value of element of Clcsmask with the single bit in error corrected.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	NA.

Table 6 Error Code: SINH (1)

Error Code:	SINH (1)
Error Description:	A bit in Clcsmask[] says that scanning is inhibited for a grid when the maintenance information in the corresponding rSMEST tuple says scanning should be allowed.
Possible Error Effect:	PC foreground isn't scanning a grid that it should. This causes call processing failures.
Corrective Action Taken:	The incorrect bit in Clcsmask[] is reset to allow scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clcsmask[] containing error.
<i>Bad Data</i> =	Value of element of Clcsmask[] containing error.
<i>Good Data</i> =	Value of element of Clcsmask with the single bit in error corrected.
<i>Logical Index</i> =	Number of bit in Clcsmask[].
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 7 Error Code: SINH (2)

Error Code:	SINH (2)

Error Description:	For an unequipped LGC on an equipped ISLU, the corresponding LGC UCI hardware bit should be zero and is not.
Possible Error Effect:	Operational and maintenance software may try to access this unequipped LGC by accident.
Corrective Action Taken:	The corresponding LGC UCI bit is written to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of LGC UCI register.
<i>Bad Data</i> =	Data in LGC UCI register.
<i>Good Data</i> =	Corrected data in LGC UCI register.
<i>Logical Index</i> =	Number of bit in Cllcsmsk[].
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 8 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for an ISLU CD circuit calculated for this equipped ISLU was not found (no match from DB read).
Possible Error Effect:	The system has lost resources, data problems.
Corrective Action Taken:	The CKTDATA audit is scheduled in elevated mode (related audit) to insert the missing rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of Cllcsmsk[].
<i>Bad Data</i> =	Circuit name of ISLU CD.
<i>Good Data</i> =	Line unit number.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	NA.

Table 9 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The Cllcsmsk[] bit for an unequipped line unit is not zero (inhibit).
Possible Error Effect:	Operational and maintenance code may try to access the circuit corresponding to this bit by accident.
Corrective Action Taken:	The Cllcsmsk bit is set to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of Cllcsmsk[].
<i>Bad Data</i> =	Value of element of Cllcsmsk[] containing error.
<i>Good Data</i> =	Corrected value of element of Cllcsmsk.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

19. LD - LG

LDATBLF

Software Release: 5E10 and later

1. BUSY LAMP FIELD (LDATBLF) AUDIT

Integrated services digital network (ISDN) attendant (ISAT) features provides business customers subscribing to ISDN service with one or more switch-based features used to serve calls requiring assistance. The only hardware required for ISAT features is the ISDN attendant console, itself. All control functions are done by the switch and by the console through the D-channel.

The direct station selection (DSS) busy lamp field (BLF) attendant feature provides customers with a display of the status of a maximum of 100 monitored directory numbers (DNs) simultaneously. The status of each monitored DN is indicated by a lamp on the hundreds group controller (HGC). When a lamp is lit, the DN is busy.

The status of monitored DN's is stored in an array of bits in the RLLDAT_BLF relation. The LDATBLF audit runs in all switching modules (SMs) in segmented mode only.

2. DATA STRUCTURES

The RLLDAT_BLF relation is used in three separate ISAT monitored call scenarios: ISAT monitored multi-line hunt groups (MLHGs), ISAT monitored PODS/analog DN's, and ISAT monitored MCA/shared DN's. The data structures involved in these three scenarios are shown in Figures 1, 2, and 3 respectively.

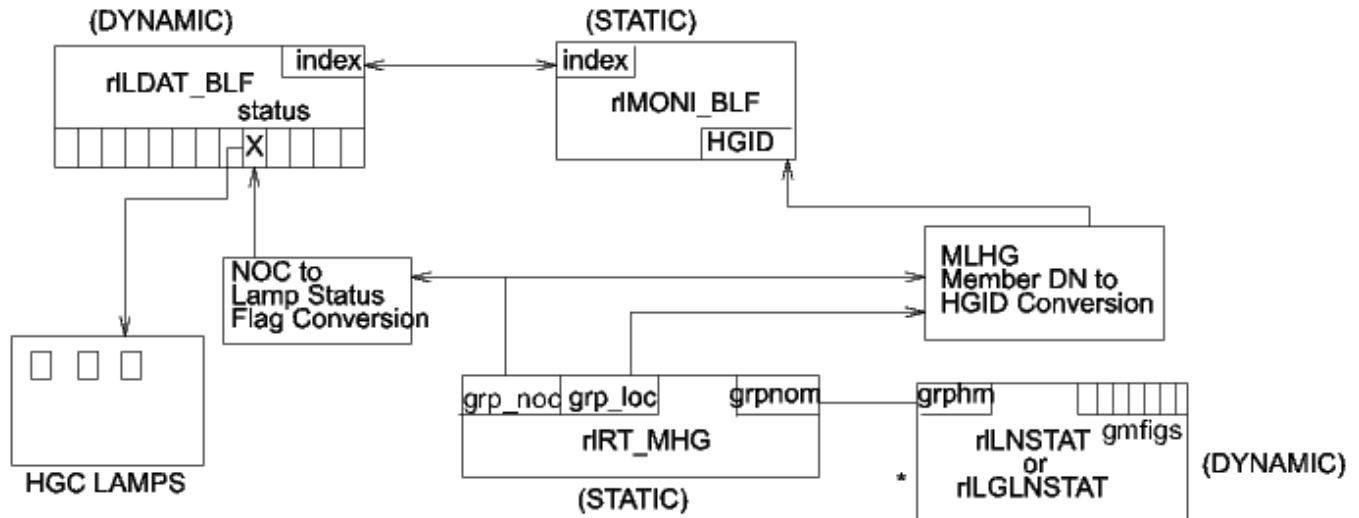
In all three scenarios, there exists a one-to-one relationship between the rLLDAT_BLF dynamic tuple and an rLMONI_BLF static tuple. The RLMONI_BLF relation contains the hundreds group identifier (hgid) associated with the lamps the rLLDAT_BLF tuple monitors.

For the ISAT monitored MLHG scenario, the RLRT_MHG static relation is used to contain the MLHG which is being monitored (grpnum) as well as the group DN (grp_noc and grp_loc) which maps to the Hundreds Group Identifier in RLMONI_BLF. The actual members status of the MLHG are stored in either RLLNSTAT (small groups) or RLLGLNSTAT (large groups) dynamic relations in a bit map (gmflg[]). These relations have an implied linkage with RLRT_MHG.

The PODS and Analog DN scenario representations are very similar. In both cases, the DN is stored in the static relation RLFC_LINE (noc and digits) which maps to the RLMONI_BLF hundreds group identifier. In addition, the port stored in RLFC_LINE (gport) has an implied linkage to RLPORTLA. If this is a PODS monitored call, the RLPORTLA will be a D-channel, and will have an implied linkage to RLDSLEQUIP where rLPORTLA.port.member equals rLDSLEQUIP.dsl.

For the MCA/Shared DN scenario, the monitored DN will be stored in the RLIS_LINE static relation (noc and loc) and will map to the hundreds group identifier stored in RLMONI_BLF. The RLIS_LINE logical termination index (ltm_idx) is the key for a corresponding rLTMSTAT dynamic tuple which contains the status of all call appearances associated with the DN. In addition, RLIS_LINE contains the logical port for the DN (lport) which is also stored in the logical port to physical port static relation RLLPT_PT. The RLLPT_PT relation contains the physical port (gport) by which the rLPORTLA tuple associated with this DN may be accessed. Lastly, for the port and party in RLLPT_PT, there may exist a corresponding RLFC_LINE tuple provided that the monitored DN is a primary DN. If the DN is a secondary only DN (a secondary DN for which there exists no primary DN), there will be no corresponding RLFC_LINE static tuple.

In all three scenarios, the noc from either RLRT_MHG, RLFC_LINE, or RLIS_LINE is used to obtain the correct RLLDAT_BLF status bit position which corresponds to the lamp on the HGC for this monitored DN.



* For small groups, this will be an rILNSTAT tuple. For large groups, this will be an rLGLNSTAT tuple.

Figure 1 (LDATBLF-1) ISAT Monitored MHLG Scenario

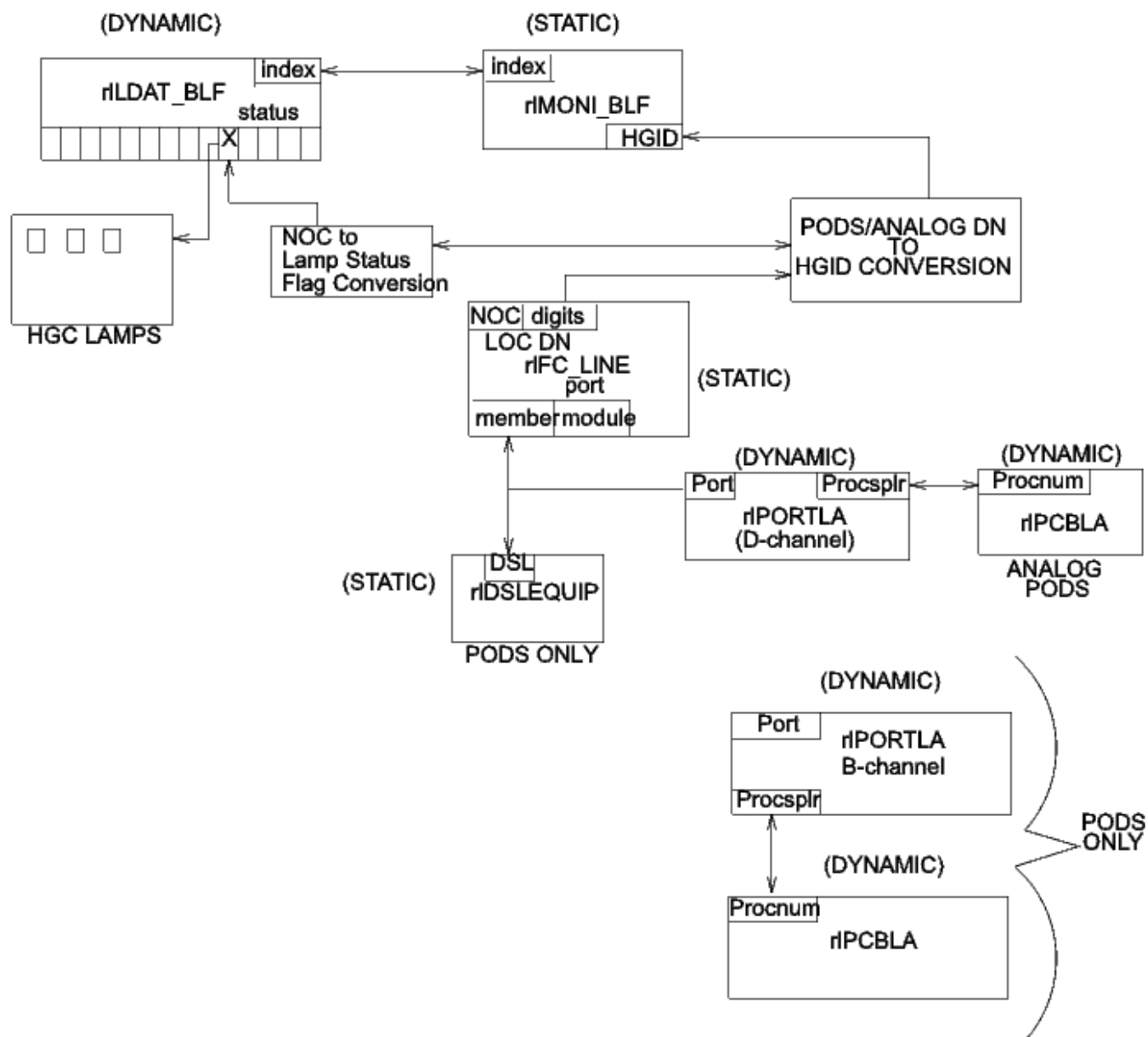


Figure 2 (LDATBLF-2) PODS and Analog ISAT Monitored Scenario

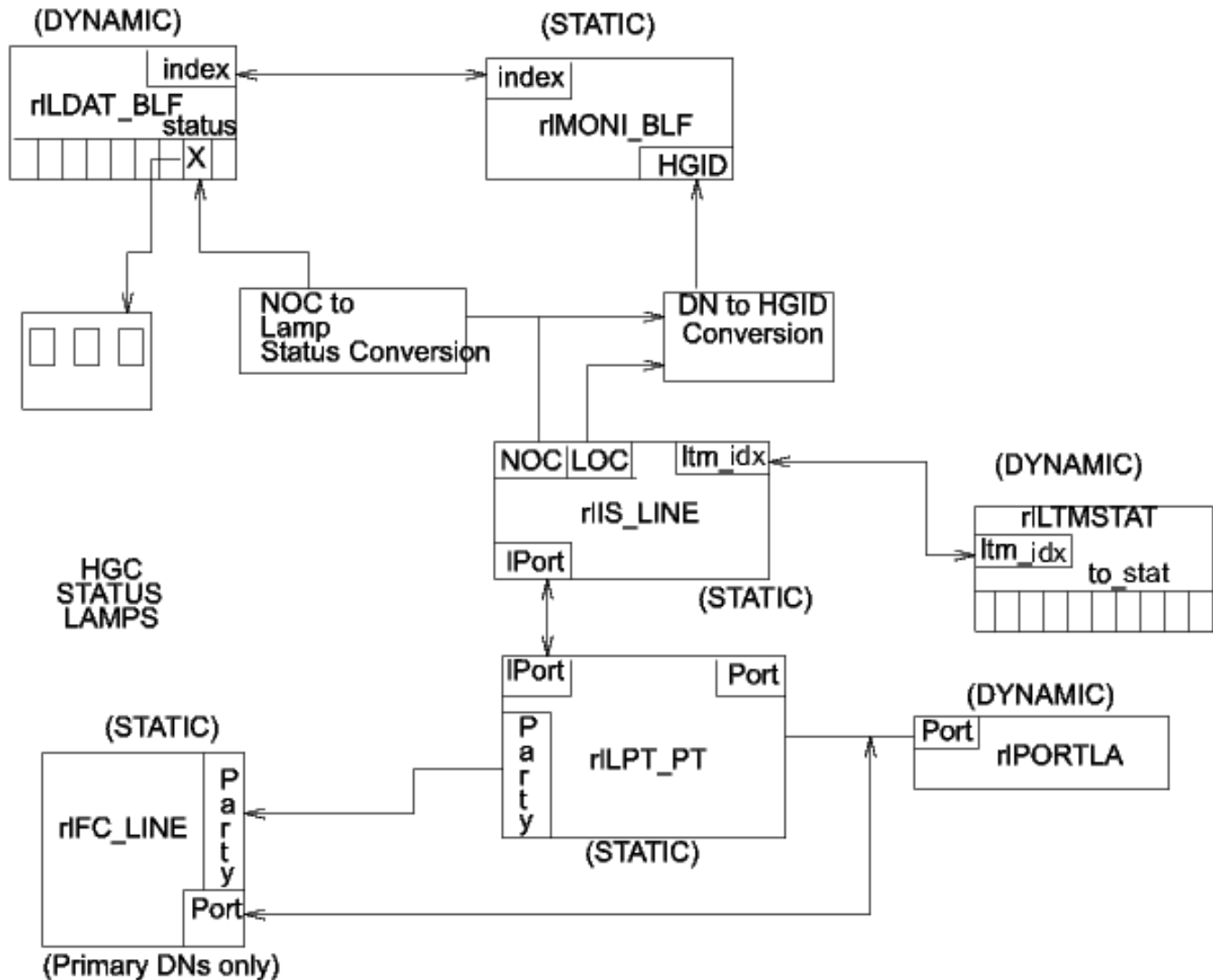


Figure 3 (LDATBLF-3) MCA/Shared DN ISAT Monitored DN Scenario

3. AUDIT CHECKS

The LDATBLF audit uses the DRSTATIC driver, and for each rLSTAT_BLF tuple, the audit makes the following checks: If the LDATBLF audit is entered:

KEY = Check for corruption of the rLSTAT_BLF tuple's key.

LOSTTUP = Check for missing rLSTAT_BLF tuple(s).

If this rLSTAT_BLF status bit is associated with an ISAT monitored MLHG:

BADIDLE = Check the idlecount field for the rLSTAT/rLGLNSTAT tuple.

CL = For this rLRT_MHG tuple, read the corresponding rIMONI_BLF tuple.

NOTIDLE = Check the lamp state for the DN associated with a monitored MLHG.

PPKEY = Check for a missing rILNSTAT or rILGLNSTAT tuple for this monitored MLHG.

RANGE = Check hunt type for this small or large monitored MLHG.

TYPE = Check the hunt type valid for this small or large monitored MLHG.

If this is a PODS or ANALOG DN:

LGKEY = Obtain the rIORTLA tuple for this monitored Analog or PODS DN.

NOPCBLA = This analog or PODS port is owned by a process; try to obtain the corresponding rIPCBLA tuple.

If this is a PODS call:

HTYPE = For this tuple in RLFC_LINE read the corresponding tuple in RLMONI_BLF which has a key (hgid) corresponding to the locdn (noc and digits) obtained from RLFC_LINE.

INV_STATE = For a PODS B-channel rIORTLA tuple obtain the corresponding D-channel rIORTLA tuple.

STATUS = Check the lamp state for this monitored Analog or PODS port.

If this is an MCA/Shared DN call:

NOORTLA = Access the rIORTLA tuple for this ISDN monitored port.

TRKG = For this tuple in RLIS_LINE read the corresponding tuple in RLMONI_BLF.

If the port for this DN is out of service:

TLAMP = Check the lamp state for this OOS monitored ISDN port is correct.

If the port for this DN is not out of service:

DYN_STAT = Obtain the rILTMSTAT tuple containing the call appearances for this ISDN monitored port for this ltm_idx in RLIS_LINE.

LAMP = Check the lamp state for this in-service monitored ISDN port is correct.

PORT = For this tuple in RLIS_LINE obtain the corresponding tuple in RLPT_PT.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The value of the idle count (idlecount) in the rILNSTAT tuple (if bad data equals 1) or rILGLNSTAT tuple (if bad data equals 2) does not agree with the actual number of idle members of this MLHG.

Possible Error Effect:	The lamp status for this monitored multi-line hunt group may be incorrect on the hundreds group controller associated with this attendant console.
Corrective Action Taken:	The correct value of the idle count will be inserted in the rLNSTAT or rLGLNSTAT tuple.
Manual Action Required:	None, unless the rIRT_MHG.htype is RTH_ISAT or RTH_MUPH. These are invalid hunt types for monitored MLHGs, since ISAT and MUPH MLHGs are not monitorable. Use ODBE to change the hunt type for this group to a valid monitorable hunt type.
Dump Description:	
<i>Error Address =</i>	The address of the rLNSTAT tuple (if bad data equals 1) or rLGLNSTAT tuple (if bad data equals 2).
<i>Bad Data =</i>	Valid values are: 1 = The dynamic tuple is rLNSTAT. 2 = The dynamic tuple is rLGLNSTAT.
<i>Good Data =</i>	The correct value of the idle count based on the contents of the gmflg[] array status map.
<i>Logical Index =</i>	The key of the rLNSTAT or rLGLNSTAT tuple.
<i>Dump =</i>	The contents of the rLNSTAT tuple (if bad data equals 1) or rLGLNSTAT tuple (if bad data equals 2) which contained the wrong idlecount.

Table 2 Error Code: CL

Error Code:	CL
Error Description:	There is a Multiline Hunt Group that is marked as monitored in the RLRT_MHG static relation (monitor = DBYES), however, there is not a corresponding tuple in the RLMONI_BLF static relation.
Possible Error Effect:	This multi-line hunt group cannot be monitored at the attendant's console, and the lamp associated with this monitored MLHG cannot be audited.
Corrective Action Taken:	None.
Manual Action Required:	There are two solutions available: 1 = The multi-line hunt group must be marked not monitored (rIRT_MHG.monitor set to DBNO), or 2 = The corresponding tuple of the static relation rMONI_BLF must be inserted. The group number for the monitored MLHG may be obtained from the key (rIRT_MHG.grpnum) field of this audit dump. If the group should not be monitored then, using ODBE, set the monitor bit (rIRT_MHG.monitor) in the rIRT_MHG tuple to DBNO. If the group is monitored, then the rMONI_BLF tuple must be inserted into the database. To insert the tuple, the key (rMONI_BLF.hgid) must be calculated from the Normalized Office Code (NOC) and the Local Directory Number (LOC DN), in binary, from the rIRT_MHG.grp_noc and rIRT_MHG.grp_loc fields respectively. The LOC and NOC can be obtained from the audit dump information. The RLMONI_BLF key (rMONI_BLF.hgid) can be calculated as follows: $rMONI\_BLF.hgid = ((rIRT\_MHG.grp\_noc \ll 8) \mid ((rIRT\_MHG.grp\_loc / 1000) \ll 4) \mid ((rIRT\_MHG.grp\_loc / 100) \% 10)) .$

	Once the key of the rIMONI_BLF tuple has been calculated for this monitored MLHG, the tuple may be inserted into the database using ODBE and the population rules for this relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the multi-line hunt group.
<i>Dump</i> =	The contents of the rLRT_MHG tuple.

Table 3 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	While trying to audit the status of a lamp corresponding to a DN associated with an ISDN-related call, the rLTMSTAT dynamic tuple, for a given rIIS_LINE static tuple, could not be accessed. There is a relationship such that, for every rIIS_LINE static tuple, a corresponding rLTMSTAT dynamic tuple should exist, which contains all call appearances associated with the ISDN monitored port represented in the rIIS_LINE tuple.
Possible Error Effect:	There is a static-to-dynamic data inconsistency, such that the sub-addresses in an rLTMSTAT dynamic tuple cannot be accessed by call processing in order to monitor the status of the hundreds group controller's lamp associated with this DN. In addition, the LDATBLF audit cannot determine what the correct state of the lamp should be in order to audit the lamp's state, and further auditing of this DN must be aborted for now.
Corrective Action Taken:	The LTMSTAT audit is scheduled as a related audit in order to insert the missing rLTMSTAT tuple, as necessary.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address where the rLTMSTAT tuple was expected to be.
<i>Bad Data</i> =	The key of the rLTMSTAT tuple.
<i>Good Data</i> =	The key of the associated rIIS_LINE static tuple.
<i>Logical Index</i> =	The key of the associated rIIS_LINE static tuple.
<i>Dump 1</i> =	The rLTMSTAT tuple data that was obtained.
<i>Dump 2</i> =	The corresponding rIIS_LINE static tuple, for which an associated rLTMSTAT dynamic tuple should exist.

Table 4 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	There is a DN that is marked as monitored in the RLFC_LINE static relation (busymon = DBYES). However, there is not a corresponding rIMONI_BLF static tuple with Hundreds Group Identifier (hgid) for the DN in RLFC_LINE (locdn and digits) to provide the index for accessing the rLDATE_BLF dynamic tuple.
Possible Error Effect:	This DN cannot be monitored at the hundreds group controller, and the lamp associated with this DN cannot be audited.
Corrective Action Taken:	None.
Manual Action Required:	Two solutions are available: 1 = The corresponding rRFC_LINE tuple must be marked not monitorable (rRFC_LINE.busymon = DBNO), or 2 = The corresponding tuple of the RLMONI_BLF static relation must be inserted. The port for the monitored DN may be obtained from the key (rRFC_LINE.port) field of this audit dump. If the group should not be monitored then, using ODBE, set the monitor bit (rRFC_LINE.busymon) in the rRFC_LINE tuple to DBNO.

	If the group is monitored, then the rIMONI_BLF tuple must be inserted into the database. To insert the tuple, the key (rIMONI_BLF.hgid) must be calculated from the normalized office code (NOC) and the local directory number (LOC DN), in binary, from the rIFC_LINE.noc and rIFC_LINE.digits fields respectively. The LOC and NOC can be obtained from the audit dump information. The RLMONI_BLF key (rIMONI_BLF.hgid) can be calculated as follows: Using ODBE, read the rINOCODE tuple with key (rINOCODE.nxx) equal to the NOC for this DN (rIFC_LINE.noc). This tuple will provide the actual NOC for use in obtaining the key for RLMONI_BLF. This actual NOC is specified by rINOCODE.noc where rINOCODE.nxx = rIFC_LINE.noc. Using the NOC obtained from the rINOCODE.noc field, calculate the RLMONI_BLF key as follows: <pre> rIMONI_BLF.hgid = ((rINOCODE.noc << 8) ((rIFC_LINE.digits / 1000) << 4) ((rIFC_LINE.digits / 100) % 10)). </pre> Once the key of the rIMONI_BLF tuple has been calculated for this monitored DN, the tuple may be inserted into the database using ODBE and the population rules for this relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIFC_LINE tuple.
<i>Dump</i> =	The contents of the rIFC_LINE tuple.

Table 5 Error Code: KEY

Error Code:	KEY
Error Description:	This rILDAT_BLF tuple's key has become corrupted.
Possible Error Effect:	Use of this tuple might propagate errors, since the wrong bits for lamps associated with monitored DNs are being marked.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rILDAT_BLF tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	The correct value of the key.
<i>Dump</i> =	The contents of the rILDAT_BLF tuple with the wrong key.

Table 6 Error Code: LAMP

Error Code:	LAMP
Error Description:	The lamp associated with this ISDN monitored port (as expressed in rILDAT_BLF.status[]) was found to be in the wrong state. For multiple call appearance (MCA) monitored calls, all sub-indexes in the rILTMSTAT tuple (bu_stat[]) must be marked busy for the lamp status bit in the rILDAT_BLF tuple to be correctly marked busy (1 for LAMP ON). Otherwise, the bit should be idle (0 for LAMP OFF).
Possible Error Effect:	The status of the lamp associated with this DN may be incorrect.
Corrective Action Taken:	The bit in the rILDAT_BLF tuple that corresponds to this lamp's DN will be set to the correct state and the hardware state of the lamp refreshed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The incorrect status of the bit.

<i>Good Data</i> =	The value to which it is corrected.
<i>Logical Index</i> =	The key of the rLDDAT_BLF tuple.
<i>Dump 1</i> =	The rLDDAT_BLF tuple.
<i>Dump 2</i> =	The rLIS_LINE static tuple.
<i>Dump 3</i> =	The associated rLTMSTAT tuple.
<i>Dump 4</i> =	The associated rLPORTLA tuple.

Table 7 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	The rLPORTLA dynamic tuple for this Analog or PODS DN could not be accessed to determine the port's state. For the port in RLFC_LINE (port) there should be a corresponding rLPORTLA tuple with rLPORTLA.port = rLFC_LINE.port.
Possible Error Effect:	A data inconsistency with the RLPORTLA and/or RLDDAT_BLF relations exists such that the DN associated with the port cannot be monitored, nor its lamp status audited.
Corrective Action Taken:	The LDDATBLF audit will skip auditing this DN for now, and the PORTLA audit will be scheduled as related audit to try to correct the inconsistency.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of the (wrong) key in the rLPORTLA tuple.
<i>Good Data</i> =	0 for an analog port, and 1 for a PODS port.
<i>Logical Index</i> =	The correct value of the key.
<i>Dump 1</i> =	The contents of the rLPORTLA tuple.
<i>Dump 2</i> =	The contents of the corresponding rLFC_LINE tuple.
	In case of a PODS port (GOOD DATA is 1), the rLPORTLA tuple corresponds to B-channel.

Table 8 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A tuple in the RLDDAT_BLF relation, which corresponds to an rLMONI_BLF static tuple, is missing. For every dynamic tuple in RLDDAT_BLF there must exist a corresponding tuple in RLMONI_BLF where rLMONI_BLF.index = rLDDAT_BLF.index.
Possible Error Effect:	The DNs in the hundreds group specified by the rLMONI_BLF tuple, which are in this SM, cannot be monitored.
Corrective Action Taken:	An attempt will be made to insert the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	GLSUCCESS if the insertion of the rLDDAT_BLF tuple was successful.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The contents of the rLMONI_BLF static tuple.

Table 9 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	According to data in the rLPORTLA tuple for a PODS or analog port, this port is owned by a call process. However the associated rLPCBLA tuple for this process could not be accessed. In this instance, rLPORTLA.proc_cplr.st_headk - DBSFPCBAL = rLPCBLA.procnum where procnum = the process number of a live call process.
Possible Error Effect:	There is no process associated with a port in use. Since the dynamic data associated with a process and port is in an illegal state, no further auditing by LDDATBLF can take place for this rLDDAT_BLF tuple until the data is recovered to known idle states. Also, this port cannot be used by any other process unless recovered to an idle state.

Corrective Action Taken:	Related audits PCBLA and PORTLA are scheduled to recover data to know consistent states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Valid values are: 1 = PODS related ports. 2 = ANALOG related ports.
<i>Bad Data =</i>	The key of the rIPCBLA tuple that was found.
<i>Good Data =</i>	The key of the rIPCBLA (process) that was expected.
<i>Logical Index =</i>	The key of the rIORTLA tuple (port name) for which there was no corresponding rIPCBLA tuple (process).
<i>Dump 1 =</i>	The rIORTLA dynamic tuple (D-channel for PODS).
<i>Dump 2 =</i>	The corresponding rIFC_LINE static tuple.
<i>Dump 3 =</i>	The corresponding rIPCBLA tuple (if found by the audit) whose process number did not match the value stored in the rIORTLA tuple.

Table 10 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	No rIORTLA tuple can be found for a corresponding tuple in the RLLPT_PT static relation. For ISDN monitorable ports, it should exist that rIORTLA.port = rIORTLA.port.
Possible Error Effect:	The audit needs the status of the port from the rIORTLA tuple to complete its checking of rILDAT_BLF data. Also, an inconsistency exists between the RLPORTLA and RLLPT_PT relations which may cause call processing failures and asserts.
Corrective Action Taken:	The PORTLA audit will be scheduled to verify the existence of the rIORTLA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rIORTLA port field.
<i>Good Data =</i>	port field from rIORTLA tuple.
<i>Logical Index =</i>	rIIS_LINE lead_in port field.
<i>Dump =</i>	The contents of the rIIS_LINE tuple.

Table 11 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The bit in the rILDAT_BLF tuple (status[]) that corresponds to the status of this MLHG was set incorrectly. For monitored MLHGs, the lamp should be on, when all members in the group (DNs) are busy (idlecount in RLLNSTAT or RLLGLNSTAT set to 0). Otherwise, the lamp should be off.
Possible Error Effect:	The status of the lamp on the hundreds group controller which monitors this MLHG will be incorrect.
Corrective Action Taken:	The lamp status bit in the rILDAT_BLF tuple that corresponds to this MLHG will be corrected to reflect the state of the group (ON = all members busy, otherwise, OFF if at least one member is idle.)
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rILDAT_BLF tuple having the wrong bit.
<i>Bad Data =</i>	Valid values are: 1 = The dynamic tuple is rILNSTAT. 2 = The dynamic tuple is rILGLNSTAT.
<i>Good Data =</i>	The correct status of the bit.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The contents of the rILDAT_BLF tuple. The contents of the rILNSTAT tuple (if

bad data equals 1) or rLGLNSTAT tuple (if bad data equals 2).

Table 12 Error Code: PORT

Error Code:	PORT
Error Description:	There is a rLIS_LINE static tuple with an lport attribute that does not match any static tuple in the RLLPT_PT relation. It should exist that for every tuple in RLIS_LINE there should exist a corresponding tuple in RLLPT_PT, where rLPT_PT.lport = rLIS_LINE.lead_lport.
Possible Error Effect:	This is an ODD inconsistency which may result in call processing failures and asserts. If the port associated with this data is monitored, the lamp state can not be maintained by call processing, nor audited by this audit. This error will fire each time the LDATBLF audit runs until the data inconsistency is fixed.
Corrective Action Taken:	None.
Manual Action Required:	Two solutions are available: 1 = The corresponding tuple of the static relation rLPT_PT must be inserted. 2 = The corresponding rLIS_LINE tuple lead_lport field should be modified to agree with an lport field in the corresponding rLPT_PT tuple. If insertion of the corresponding RLLPT_PT entry is desired, the following rules apply: rLIS_LINE.lead_port = rLPT_PT.lport = rIGROUP_PORT.lport = rIPORT_GROUP.lport, rLPT_PT.lpt_type = LPT_ISDN, rLPT_PT.port = rIPORTLA.port. These fields may be examined and manipulated using ODBE.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	lport field of buffer used to read RLLPT_PT.
<i>Good Data</i> =	return code from DB primitive used to read RLLPT_PT.
<i>Logical Index</i> =	lead_port attribute of the rLIS_LINE tuple.
<i>Dump</i> =	The contents of the rLIS_LINE tuple.

Table 13 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	The tuple in the dynamic hunting data relation RLLNSTAT or RLLGLNSTAT has the wrong key (grpnum). It should exist that for every rLGLNSTAT/rLNSTAT tuple there exists a corresponding tuple in RLRT_MHG where rLRT_MHG.grpnum = rLXLNSTAT.grpnum. - If the bad data equals 1, the rLNSTAT tuple has wrong key. - If the bad data equals 2, the rLGLNSTAT tuple has wrong key.
Possible Error Effect:	A static-to-dynamic inconsistency exists between RLRT_MHG and RLLNSTAT/RLLGLNSTAT relations such that no routing may take place on this group, nor, if monitored, no lamp status update can be made.
Corrective Action Taken:	The rLNSTAT and rLGLNSTAT audits will be scheduled as related audits to insert the rLNSTAT/rLGLNSTAT dynamic tuples, if missing, and correct the key, if it was corrupted.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Valid values are: 1 = The RLLNSTAT relation. 2 = The RLLGLNSTAT relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the corresponding rIRT_MHG tuple.
<i>Dump</i> =	The contents of the rIRT_MHG tuple.

Table 14 Error Code: RANGE

Error Code:	RANGE
Error Description:	There is an MLHG that is marked as monitored (monitor = DBYES). However, the hunt type (rIRT_MHG.htype) of this hunt group is not included in the cases that this audit considers valid. ISDN Attendant "ISAT" (RTH_ISAT) and multi-position hunt group "MUPH" (RTH_MUPH) MLHGs are not monitorable. This error will for this MLHG every time the LDATBLF audit runs until fixed.
Possible Error Effect:	The lamp for this MLHG can not be monitored by a lamp at the hundreds group controller by call processing, nor audited, because an ODD inconsistency exists where a non-monitorable hunt type is marked as monitored.
Corrective Action Taken:	None.
Manual Action Required:	The MLHG should be marked not monitored for ISAT and MUPH hunt types (rIRT_MHG.htype must not be RTH_ISAT or RTH_MUPH). Either change the hunt type in the rIRT_MHG static tuple to a legal monitorable hunt type, or change the group so that it is not monitorable (rIRT_MHG.monitor = DBNO).
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the MLHG.
<i>Dump</i> =	The contents of the rIRT_MHG tuple.

Table 15 Error Code: STATUS

Error Code:	STATUS
Error Description:	The bit in the rLDAT_BLF tuple that corresponds to the DN for this Analog or PODS port (status[]) is incorrect. These are the rules: (1) If the port is in-service/idle, (rPORTLA.bsc_state = TMIS and rPORTLA.proc_cplr.st_headk = 0) the status bit should be idle (0) and the lamp OFF. (2) If the port is in-service/busy, (rPORTLA.bsc_state = TMIS and rPORTLA.proc_cplr.st_headk != 0), and the process owning the port is required to mark the BLF lamp to busy (rPCBLA.priority = RTFC), the status bit should be busy (1) and the lamp ON. (3) If, for PODS, the B-channel is OOS, (rPORTLA.bsc_state = TMOOS) and the D-channel is OOS/BLOCKED, (rPORTLA.list = TMLTDCHOOS) the status bit should be idle (0) and the lamp OFF. (4) If, for PODS, the B-channel is OOS, and the D-channel is In-service, this is treated as a transient case, and not audited by the LDATBLF audit. If things stay transient for too long, recovery is provided by the PORTLA audit. When this occurs, the PORTLA audit fires TRANCK and BTRANCK errors. This case is listed here just for clarification. (5) If, for PODS, the B-channel is OOS, and the D-channel is OOS, but

	not on the OOS/BLOCKED list, the status bit should be busy (1) and the lamp ON. (6) For PODS, if a B-channel is OOS BLOCKED on a multipoint DSL which is busy lamp field (BLF) busy/monitorable, the BLF feature is not supported by call processing. For this case, the associated status bit should be idle (0) and the lamp OFF. (7) For any PODS/analog instance where the process owning the port is not a call process (testing/maintenance processes, for example) the BLF feature is not supported. For this case, the status bit should be idle (0) and the lamp OFF. (8) If for analog, the port is OOS, the status bit should be busy (1) and the lamp ON.
Possible Error Effect:	The status of the dynamic data for the light monitoring this DN on the attendant console's hundreds group controller will be incorrect. The lamp on the customers attendant console no longer correctly reflects the status of this DN.
Corrective Action Taken:	The bit in the rLDATE_BLF tuple that corresponds to this DN will be set to the correct state, and the hardware state of the lamp refreshed.
Manual Action Required:	If this error continues to roll out of an SM for the same RLDATE_BLF key, check the RLFC_LINE ltm_idx fields in the ODD. They should be populated with all unique values (no duplicates) except when the field is 0 (meaning that the DN is not monitored). Also, the number of non-zero entries should correspond to the number of attendant consoles on the SM. In addition, the RLMONI_BLF index fields should all be unique for a given SM.
Dump Description:	
<i>Error Address =</i>	Address of the rLDATE_BLF tuple.
<i>Bad Data =</i>	The status of the LDAT bit for this DN.
<i>Good Data =</i>	If the port was busy when the error occurred, the program ID of the system process which created the terminal process owning the port (OSbpcb[rIPCBLA.procnum].progi) will be dumped. Otherwise, the state the LDAT bit was corrected to will be dumped.
<i>Logical Index =</i>	The key of the rLDATE_BLF tuple.
<i>Dump 1 =</i>	The contents of the rLDATE_BLF tuple.
<i>Dump 2 =</i>	The contents of the corresponding rLFC_LINE tuple.
	Note: The following two dumps allow you to determine what type of call data was found incorrect. If only the D-channel rPORTLA tuple is dumped, this means analog. If both the D and B rPORTLAs are dumped, this means PODS.
<i>Dump 3 =</i>	The rPORTLA tuple (D-channel for PODS, or ANALOG.)
<i>Dump 4 =</i>	The B-channel rPORTLA tuple (PODS only.)
	If a process owned this monitored port when the error was found, the following data will be dumped:
<i>Dump 5 =</i>	The rIPCBLA tuple where (D-Channel) rPORTLA.proc_cplr.st_headk - DBSFPCBLA = (key) rIPCBLA.procnum.

Table 16 Error Code: TLAMP

Error Code:	TLAMP
Error Description:	The bit in the rLDATE_BLF tuple that corresponds to this DN associated with an OOS ISDN port is incorrect. For lamps associated with DNs for MCA monitored ports, if the port is OOS, but not BLOCKED, the status bit should be busy (1) and the lamp ON. If the port is OOS BLOCKED, the status bit should be idle (0) and the lamp OFF.
Possible Error Effect:	The status of this DN on the hundreds group controller will be incorrect until the

	audit runs. This means that the user may unknowingly miss incoming calls.
Corrective Action Taken:	The bit in the rLDAT_BLF tuple that corresponds to this DN will be set to busy and the lamp ON if the port is OOS but not BLOCKED. Otherwise, the bit should be idle and the lamp OFF if the port is OOS/BLOCKED.
Manual Action Required:	If the busymon fields in the rLIS_LINE and rLFC_LINE tuples do not match, they should be corrected so that they both are DBYES or DBNO. These tuples are dumped when this error fires, and their keys are available from their respective dumps.
Dump Description:	
<i>Error Address</i> =	Not an address; the value of the list attribute of the corresponding rIORTLA tuple, indicates the status of the port.
<i>Bad Data</i> =	Status of the rLDAT_BLF bit: 0 = DBNO for not busy. 1 = DBYES for busy.
<i>Good Data</i> =	The value to which it is corrected.
<i>Logical Index</i> =	The key of the rLDAT_BLF tuple.
<i>Dump 1</i> =	The contents of the rLDAT_BLF tuple before it is corrected.
<i>Dump 2</i> =	The corresponding rLIS_LINE tuple.
<i>Dump 3</i> =	The corresponding rIORTLA tuple.
<i>Dump 4</i> =	If the corresponding rLFC_LINE static tuple was obtained, it will also be dumped.

Table 17 Error Code: TRKG

Error Code:	TRKG
Error Description:	There is a DN that is marked as monitored in the rLIS_LINE tuple. However, there is not a corresponding rIMONI_BLF tuple to provide the index for accessing the rLDAT_BLF tuple.
Possible Error Effect:	An ODD inconsistency between RLFC_LINE and RLMONI_BLF relations exists. A DN is not monitored, yet it is marked as monitored in RLIS_LINE.
Corrective Action Taken:	None.
Manual Action Required:	The corresponding rLIS_LINE tuple (which is dumped when this error fires) must be marked not monitorable (rLIS_LINE.busymon = DBNO). Then, using recent change, create a monitored DN, if desired.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The contents of the rLIS_LINE tuple.

Table 18 Error Code: TYPE

Error Code:	TYPE
Error Description:	There is a MLHG that is marked as monitored (monitor = DBYES). However, the hunt type (rIRT_MHG.htype) of this hunt group is not included in the cases that call processing (RTA) considers valid. ISAT (RTH_ISAT) and MUPH (RTH_MUPH) MLHGs are not monitorable. This error will for this MLHG every time the LDATBLF audit runs until fixed.
Possible Error Effect:	The lamp for this MLHG can not be monitored by a lamp at the hundreds group controller by call processing, nor audited, because an ODD inconsistency exists where a non-monitorable hunt type is marked as monitored.
Corrective Action Taken:	None.
Manual Action Required:	The MHLG should be marked not monitored for ISAT and MUPH hunt types (rIRT_MHG.htype must not be RTH_ISAT or RTH_MUPH). Either change the hunt type in the rIRT_MHG static tuple (which is dumped when this error fires, and whose key is the group number of the monitored MLHG) to a legal monitorable hunt type, or change the group so that it is not monitorable (rIRT_MHG.monitor = DBNO).

Dump Description:	
<i>Error Address</i> =	NA
<i>Bad Data</i> =	NA
<i>Good Data</i> =	Valid values are: 1 = Associated with an rLNSTAT (small MLHG) tuple. 2 = Associated with an rLGLNSTAT (large MLHG) tuple.
<i>Logical Index</i> =	The key of the MHLG.
<i>Dump</i> =	The contents of the rIRT_MHG tuple.

LGLNSTAT

Software Release: 5E10 and later

1. LARGE LINE STATUS (LGLNSTAT) AUDIT

To process a call, the RTA routing process in the switching module (SM) and in a remote switching module (RSM) must determine which ports are available for terminating calls. Information about the types of hunted ports is kept on bit maps. A bit map is included in each tuple in the large line status relation (RLLGLNSTAT), which contains information about large multiline hunt groups (between 64 and 2015 lines per group) of hunted lines. The RLLGLNSTAT relation resides in the SM, but it may also reside in an RSM. During stand-alone operation the tuples for other ports that the RSM contains are inserted into RLLGNSTAT; when the RSM leaves stand-alone operation, these tuples are deleted.

The large line status audit (LGLNSTAT) checks the dynamic relation rLGLNSTAT for extra tuples. It also performs a check on the status bits for each group. For information about smaller groups of hunted lines refer to the line status audit (LSTAT). For information about groups of hunted trunks, refer to the trunk status audit (TKSTAT) and the large trunk status audit (LGTKSTAT).

This audit is part of the Routing and Terminal Allocation audits subgroup.

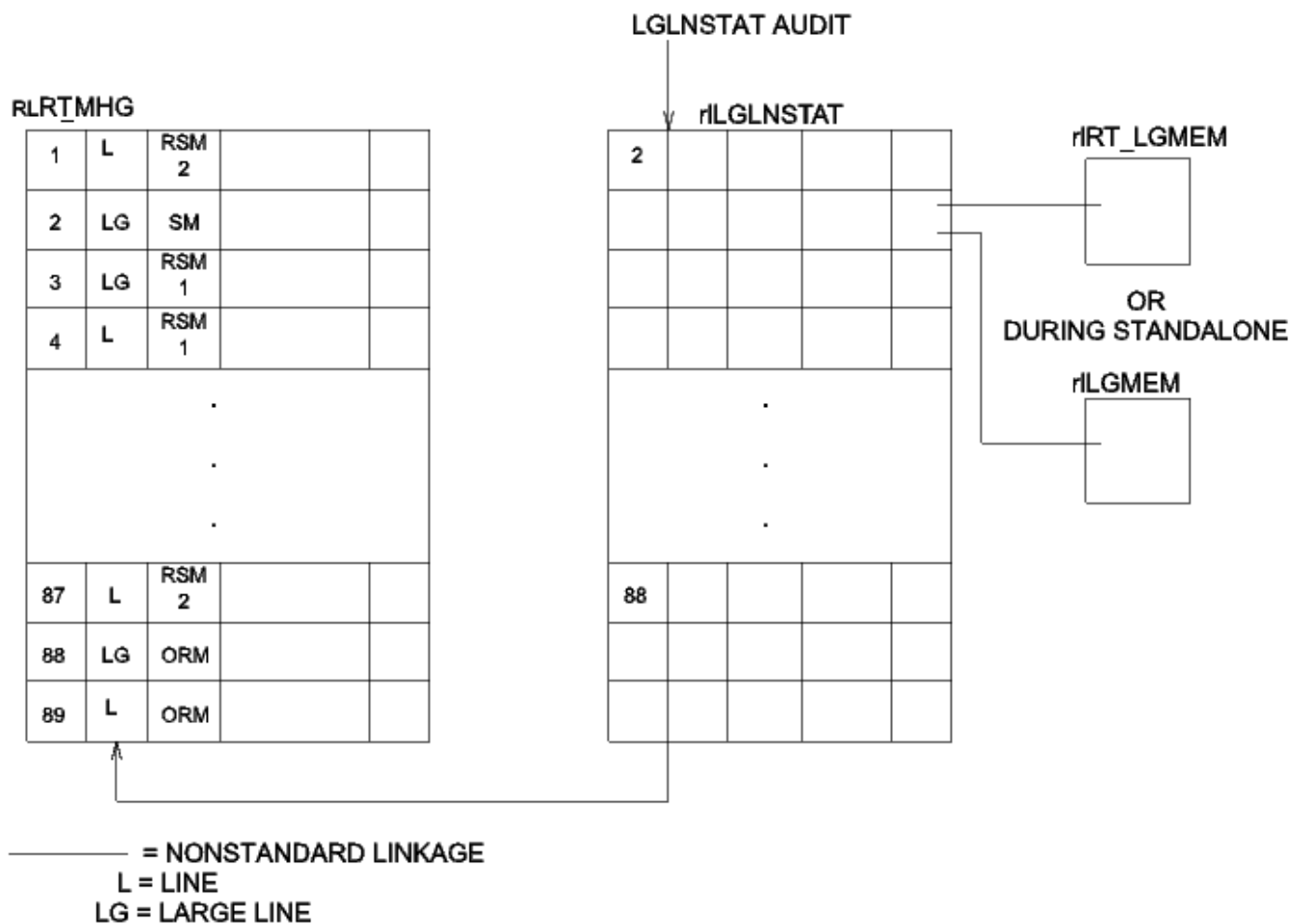
2. DATA STRUCTURES

The data structures checked by the LGLNSTAT audit are the multi-line hunt group information relation (RLRT_MHG) and the large line status relation (RLLGLNSTAT). Figure 1 illustrates the two relations.

The new paragraph RLRT_MHG relation is a static (write-protected) relation that contains information about all multiline hunt groups. This information includes a key (grpnum), the hunt type, the hunting SM number, and the type of group (local or global) it is. Both RLRT_LGMEM and RLLGMEM are relations that contain information about each MLHG member port equipage. The RLRT_LGMEM relation resides on the designated hunting module, SM/ORM/RSM. The RLLGMEM relation is created only on the RSM/ORM with standalone capability. The large line status audit steps through RLLGLNSTAT and verifies it against the information contained in the RLRT_MHG and either the RLRT_LGMEM or RLLGMEM.

SM & RSM*

AM



*IN AN RSM ENVIRONMENT, THE rLGLNSTAT RESIDES
IN BOTH THE HUNTING SM AND THE RSM.

Figure 1 (LGLNSTAT-1) Large Line Status Audit

3. AUDIT CHECKS

The LGLNSTAT audit uses the static driver (DRSTATIC) to step through the RLLGLNSTAT relation and sequentially retrieve tuples. The DRSTATIC driver invokes a semantic check function that performs the following checks:

- (a) The audit verifies each rLGLNSTAT tuple against the corresponding rRT_MHG tuple by using the key (group number) of rLGLNSTAT to read rRT_MHG relation. The audit looks specifically for extra rLGLNSTAT tuples.
- (b) The audit performs a Boolean check between the following bit arrays: port available flag (PORTAVAIL), stop hunt flag (STOPHUNT), make busy flag (MAKEBUSY) and the group member flag (GMFLG). A boolean check uses the boolean operators AND, NOT, and OR. The check is as follows:

If the PORTAVAIL flag is 1 AND NOT the MAKEBUSY flag

OR

The STOPHUNT flag is 1

Then the GMFLAG should be 1

- (c) The audit checks the hunt type. Only RTH_LGREG, RTH_LGCIR, RTH_LGUCD, PTH_PSLGUCD, and RTH_PPLMHG are valid.
- (d) The audit ensures that the module number is correct for the tuple.
- (e) The audit verifies that busy/idle statuses are busy for large MLHG members that are not equipped with ports.

4. ERROR HANDLING

Excess tuples, tuples not in the appropriate module, and tuples with the wrong hunt types, are deleted from the database. Boolean checks, that pass, are further checked for the group member flag for 1. If it is not 1, the group member flag is set to 1. Status bit checks, that fail for non-equipped members, are connected by the LGLNSTAT audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	While auditing an rLGLNSTAT tuple, a DAGETLGMEM did not find the associate rLGMEM tuple. Since during Stand Alone RLLGMEM relation is audited against static RLGROUP_PORT/RLPSGP_PRT relation, audit should believe dynamic rLGMEM tuple and delete dynamic rLGLNSTAT tuple.
Possible Error Effect:	Incorrect use of data resources for multi-line hunt groups.
Corrective Action Taken:	The extra rLGLNSTAT tuple deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLGMEM tuple.
<i>Bad Data =</i>	The key of the extra rLGLNSTAT tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The busy/idle status in a tuple of the relation RLLGLNSTAT is inconsistent with the corresponding member equipage tuple bit map. The nonequipped member should be marked busy. If the KEY field of the dump is 1 (SM Stand Alone) then an rLGLNSTAT tuple is inconsistent with an rLGMEM tuple, or if it is 0 then the rLGLNSTAT tuple is inconsistent with an rIRT_LGMEM tuple.
Possible Error Effect:	Failure in terminating to an invalidly marked idle members.
Corrective Action Taken:	The audit set the nonequipped members to correct status, busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLGLNSTAT tuple having the wrong busy/idle status.
<i>Bad Data =</i>	The array containing the members busy/idle status.
<i>Good Data =</i>	The array containing the members equipage information.
<i>Logical Index =</i>	This is not a key. Verification of busy/idle status is with rLGMEM tuple if this dump is 1, or else with rIRT_LGMEM tuple if this dump is 0.
<i>Dump =</i>	NA.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	NA.

Table 4 Error Code: GMFLG

Error Code:	GMFLG
Error Description:	The Boolean check between the status flag bits (portavail, makebusy, and stophunt) has passed, but a gmflg status bit(s) was found to be zero (not set) when it should be set to 1. There are five possible combinations for this error to occur.
Possible Error Effect:	If this group member is picked by RTA for routing, the erroneous status data will propagate further inconsistencies.
Corrective Action Taken:	The gmflg status bit(s) is set to 1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLGLNSTAT tuple.
<i>Bad Data</i> =	Erroneous word in the gmflg bit array.
<i>Good Data</i> =	Corrected word in the gmflg bit array.
<i>Logical Index</i> =	Group member of this tuple.
<i>Dump</i> =	NA.

Table 5 Error Code: KEY

Error Code:	KEY
Error Description:	The tuple of the relation rLGLNSTAT has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLGLNSTAT tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 6 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	The group is of the hunt type RTH_PSLGUCD but the makebusy[] or stophunt[] fields are not zero.
Possible Error Effect:	The group may not be hunted correctly.
Corrective Action Taken:	The fields are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLGLNSTAT tuple.
<i>Bad Data</i> =	The erroneous makebusy[] value.
<i>Good Data</i> =	The erroneous stophunt[] value.
<i>Logical Index</i> =	grpnum of the tuple.
<i>Dump</i> =	NA.

Table 7 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP

Error Description:	The audit has found an inconsistency between the static relations RLRT_LGMEM and RLRT_MHG. Current SM is the hunting SM of the multi-line hunt group (MLHG) as indicated by the field in rIRT_MHG tuple and therefore the key of this tuple is used to access the MLHG member equipage tuple in the RLRT_LGMEM relation but no matching tuple is found and the audit cannot continue to do checks.
Possible Error Effect:	Audit continues to the next tuple of the RLLGLNTAT relation.
Corrective Action Taken:	Audit report an error.
Manual Action Required:	Verify tuples rIRT_MHG and rIRT_LGMEM in the ODD, using the rIRT_MHG tuple key found in the "KEY" field of the audit report. Corresponding tuple rIRT_LGMEM with the same key should exist on the same SM. Correct any inconsistencies found.
Dump Description:	
<i>Error Address =</i>	The address of the rIRT_MHG tuple.
<i>Bad Data =</i>	The key of rIRT_LGMEM tuple returned from data base.
<i>Good Data =</i>	The hunt module of multi-line hunt group.
<i>Logical Index =</i>	The key of rIRT_MHG tuple used to access rIRT_LGMEM tuple.
<i>Dump =</i>	The contents of the rIRT_MHG tuple.

Table 8 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the AM, only tuples from global hunt groups should be present. In an RSM, only tuples from local hunt groups on that RSM should be present except during stand-alone operation when tuples from global hunt groups are allowed.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLGLNSTAT tuple.
<i>Bad Data =</i>	Module number for local group; 0 if global.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	grpnum of this tuple.
<i>Dump =</i>	NA.

Table 9 Error Code: RELQKEY

Error Code:	RELQKEY
Error Description:	The tuple of the relation rLGMEM has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLGMEM tuple having the wrong key.
<i>Bad Data =</i>	The value of the wrong key.
<i>Good Data =</i>	The correct value of the key.
<i>Logical Index =</i>	This is not a key but the hunt module of the multi-line hunt group.
<i>Dump =</i>	NA.

Table 10 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in rLGLNSTAT (that is RTH_LGREG, RTH_LGCIR, RTH_LGUCD, RTH_PLREG, or RTH_PPLMHG) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLGLNSTAT tuple.
<i>Bad Data =</i>	Hunt type of the group.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	grpnum of this tuple.

<i>Dump</i> =	NA.

LGMEM

Software Release: 5E10 and later

1. MULTILINE HUNT LARGE GROUP MEMBER STATUS (LGMEM) AUDIT

To process a call during the stand-alone (SA) condition, the routing and terminal allocation (RTA) routing process in the remote switching module (RSM) and in the optically remote module (ORM) must determine which ports are available for terminating calls. Information about a large (up to 2016 members) multi-line hunt group (MLHG) is kept on bit maps, with each bit representing a member of the MLHG. The bit map is included in each tuple in the MLHG status relation (RLLGMEM). When a member is populated for a MLHG, the corresponding bit must be set to "1" to indicate the member is active. If the member is not active, the bit must be set to "0". The relation RLLGMEM is created during the SA condition in a RSM/ORM for all global MLHG members for which that switching module (SM) is the hunting SM and for all global members that have ports in that SM.

The MLHG member status (LGMEM) audit checks tuples in the dynamic relation RLLGMEM for extra tuples. It also performs a check on the status bit for each group. The LGMEM audit verifies with the static relations RLGROUPOUT/RLPSGRP_PRT that each MLHG member set to "1" is a valid active member.

This audit is scheduled only during the SA mode.

2. DATA STRUCTURES

The data structures checked by the LGMEM audit are the group and member number to port relation (RLGROUPOUT), and the packet switching multiline/trunk group and member number to packet switching port relation (RLPSGRP_PRT). The RLGROUPOUT and RLPSGRP_PRT relations are static (write-protected) relations that contain information about all global MLHG ports. For every member of the global MLHG, a rLGROUPOUT/rLPSGRP_PRT tuple must exist in the SM which owns the port, in the designated hunting SM, and in the designated routing SM. The rLPSGRP_PRT is populated only for RTH_PSUCD and RTH_PSLGUCD hunt types (packet switching ports), while rLGROUPOUT is populated for all other MLHG hunt types.

3. AUDIT CHECKS

The LGMEM audit uses the static driver (DRSTATIC) to step through the RLLGMEM relation and sequentially retrieve tuples. The DRSTATIC driver invokes a semantic check function that verifies each rLGMEM tuple against the rLGROUPOUT/rLPSGRP_PRT tuples by using the key (group number) of rLGMEM to read the static relations. The audit looks specifically for members that are marked as active in the RLLGMEM relation, but for which there is no corresponding tuple in the static relation.

4. ERROR HANDLING

Excess tuples, tuples for which all members are not active, are deleted from the relation. Bits that are set to "1" (active), for which there are no MLHG members associated with it, are corrected to "0" (not active). If a tuple of the member status relation contains an invalid data in its key, the key is corrected.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An ASSERT will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	rLGMEM tuple.

Table 2 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	Audit cannot find static rLRTMHG tuple using dynamic rLGMEM key.
Possible Error Effect:	An error will report if this tuple is not found.
Corrective Action Taken:	The dynamic rLGMEM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLRT MHG tuple.
<i>Bad Data</i> =	The key of rLGMEM tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 3 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	The corresponding member in the dynamic relation RLLGMEM was incorrectly initialized to valid ("1").
Possible Error Effect:	Initialization was wrong and calls will fail to that member.
Corrective Action Taken:	The member will be set to invalid ("0").
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLGMEM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	0 (invalid member).
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rLGMEM tuple.

Table 4 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	A DAGETLGMEM macro returns a pointer whose key does not match the tuple buffer key of rLGMEM tuple.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLGMEM tuple having the wrong key.
<i>Bad Data</i> =	The key value of pointer to tuple.
<i>Good Data</i> =	The key value of tuple buffer pointer.
<i>Logical Index</i> =	The correct value of the key.
<i>Dump</i> =	NA.

Table 5 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The status of SM indicate normal and this dynamic tuple rLGMEM should not exist in this situation.
Possible Error Effect:	Incorrect busy/idle status may exist for this multi-line hunt group.
Corrective Action Taken:	The dynamic rLGMEM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLGMEM tuple buffer.
<i>Bad Data</i> =	The key of rLGMEM tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 6 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type as indicated in rIRT_MHG tuple for the group number in RLLGMEM relation.
Possible Error Effect:	An error will report if this group's hunt type is not of the type in rIMEM tuple (that is, RTH_LGREG, RTH_LGCIR, RTH_LGUCD, RTH_PSLGUCD, RTH_PPLMHG, or RTH_MUPH).
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLGMEM tuple.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of rLGMEM tuple used to access rIRT_MHG tuple.
<i>Dump</i> =	rIRTMHG tuple

LGSFSTAT

Software Release: 5E10 and later

1. LARGE SIMULATED FACILITY STATUS (LGSFSTAT) AUDIT

Simulated facility groups (SFGs) provide customers with the appearance of dedicated trunks without requiring the local switching office to designate specific hardware for each customer. Simulated facility groups thus increase the efficiency of the network while maintaining all the customer's features and needs. The simulated groups allow customers with private branch exchange (PBX) or centrex features status lights for each group; when the group is completely busy, the busy lamp status flag lights. Another feature, the make-busy key, allows the customers to give the appearance of a busy trunk group; this is done, for example, when a conference call needs to be placed at a specific time.

The large simulated facility status audit (LGSFSTAT) verifies the existence and consistency of the data in large (more than 96 trunks) simulated facility groups. For information about small trunk groups (1-96 trunks), refer to the simulated facility status audit (SFSTAT).

The LGSFSTAT audit is part of the routing and terminal allocation (RTA) audits subgroup. It runs in the communications module processor (CMP) in the initialization and segmented modes.

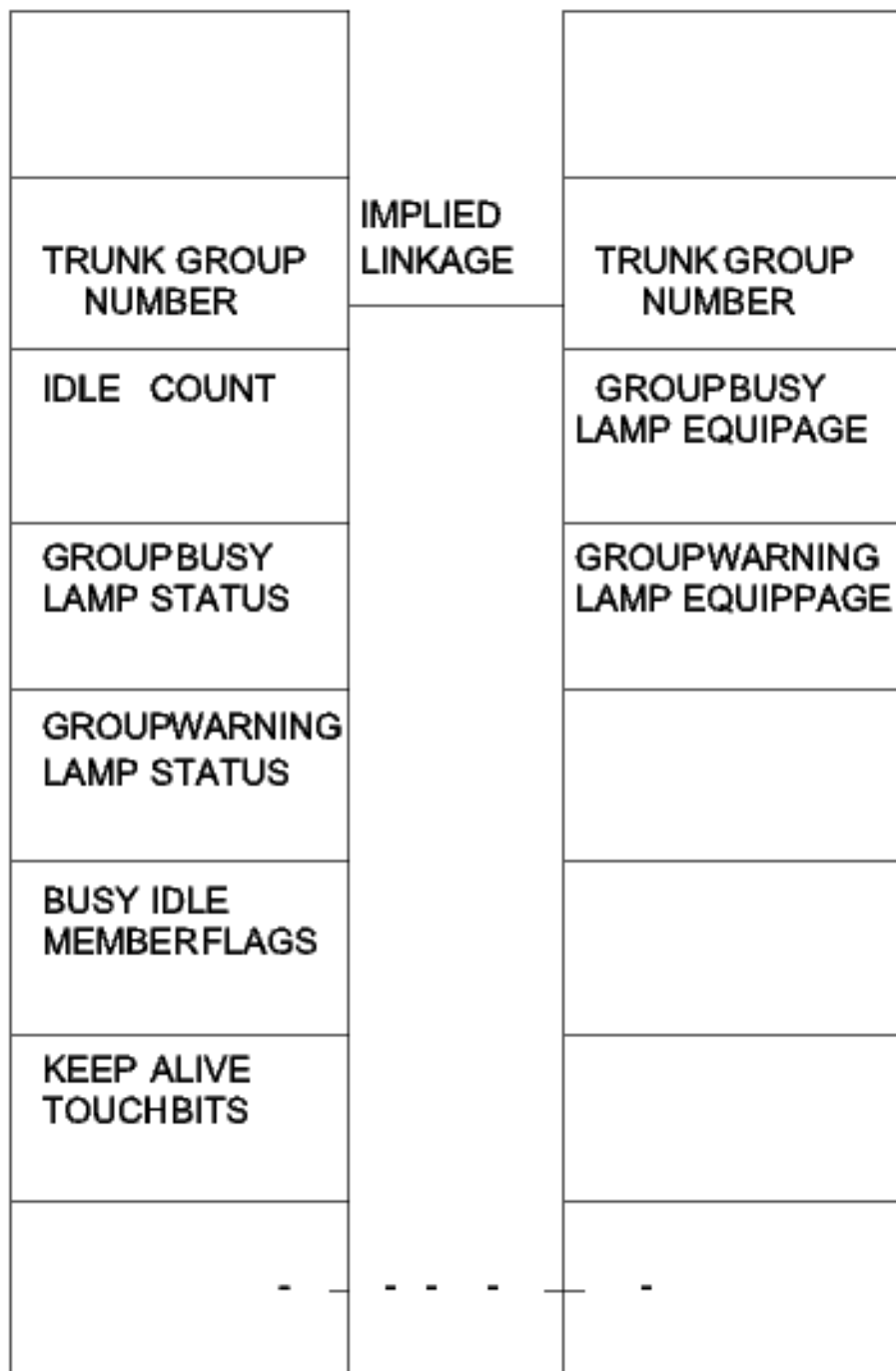
2. DATA STRUCTURES

This audit checks several fields in the large simulated facility group relation (RLLGSFSTAT), including the key, idle count, and group busy lamp fields. The RLLGSFSTAT relation has one tuple per simulated facility group. Each tuple contains one bit, a busy/idle flag, per member. The audit verifies the information in RLLGSFSTAT with information in the static simulated facility group relation (RLRT_SFG). The data structures of the LGSFSTAT audit are shown in Figure 1 .

The LGSFSTAT audit checks several fields in the dynamic relation RLLGSFSTAT and the static relation RLRT_SFG. These fields support the integrated services digital network (ISDNs) busy and warning lamps. The data structures of the LGSFSTAT audit are shown in Figure 2 .

rLGSFSTAT

rIRT_SFG



- - -# = IMPLIED LINKAGE

Figure 1 (LGSFSTAT-1) Data Structures of the LGSFSTAT Audit

rLGSFSTAT

TRUNK GROUP NUMBER
IDLE COUNT
GROUPBUSY LAMP STATUS
B&I FLAGS
ISDN WARNING LAMP STATUS
- - - - -

rIRT_SFG

TRUNK GROUP NUMBER
GROUPBUSY LAMP EQUIPAGE
ISDN TRUNK GROUP BUSY/ WARNING LAMP
ISDN ATTENDANT TRUNK GROUP WARNING THRESHOLD
- - - - -

#- - - -# = NONSTANDARD LINKAGE

Figure 2 (LGSFSTAT-2) Data Structures of the LGSFSTAT Audit

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) to step through the RLLGSFSTAT relation and perform a dynamic-to-static existence check based on the key (trunk group number) fields of rLGSFSTAT and rIRT_SFG to ensure that a tuple in RLRT_SFG exists for each tuple in RLLGSFSTAT. If the office is equipped with the busy lamp, the audit performs the following semantic checks.

- During the existence check, the audit counts the number of idle bits. When the semantic check function is invoked, the audit compares its idle bit count with the automatic count made by call processing. Unlike the trunk status and its TKSTAT, LGTKSTAT, and TRKOWNER, this check of the idle counts is unconditional whether or not an analog console is associated with this simulated facilities group.
- If there is an analog console associated with this simulated facilities group, the audit checks that the status of the make-busy key, the idle count and the busy lamp are consistent. If the make-busy key is pressed, the lamp should be on, regardless of the idle count.

- (c) If there is an ISDN attendant console associated with the simulated facilities group, the audit checks that the busy lamp and make-busy key of this console are consistent with the idle count. If the ISDN attendant console is equipped with a warning lamp, this lamp's state is checked against the idle count and the preset value in the simulated facilities group information relation `rIRT_TRKG`. If the make-busy key is pressed, the busy and warning lamps should be on regardless of the idle count.

4. ERROR HANDLING

If the dynamic-to-static existence check fails because the audit finds no static tuple corresponding to the tuple in `RLLGSFSTAT`, the audit deletes the tuple in `RLLGSFSTAT`.

If the semantic check fails, the audit initializes the `rLGSFSTAT` tuple. The audit corrects counts busy lamp errors.

The audit corrects ISDN attendant console busy and warning lamp for inconsistencies if a simulated facilities group is equipped with such lamps.

When running in keep alive mode, stuck bits are detected and idled, the idle count [count of available simulated facility (SF) members left in this group] is updated (incremented) to account for all bits that were found stuck, and subsequently idled. Any analog busy lamps, and/or ISDN attendant busy or warning lamps are refreshed, as needed, to correspond to the new count of available SF members in the group.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	A count of the idle group members did not equal the stored idle count.
Possible Error Effect:	Measurements would be in error for this group. Possible mishandling of the group busy lamp, if equipped.
Corrective Action Taken:	The idle count was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	RLLGSFSTAT tuple address.
<i>Bad Data =</i>	Idle count field.
<i>Good Data =</i>	Correct count of idle members.
<i>Logical Index =</i>	Group number.
<i>Dump =</i>	RLLGSFSTAT tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Group number not in static relation <code>RLRT_SFG</code> .
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Group number not in static relation <code>RLRT_SFG</code> .
<i>Dump =</i>	lgsfstat tuple that was deleted.

Table 3 Error Code: ISDNTGBWL

Error Code:	ISDNTGBWL
Error Description:	The <code>isdntgbwl</code> field, which indicates if this SFG is associated with an ISDN attendant console busy and possible warning lamp, is a dynamic duplicate of the

	isdntgbwl field stored in static data (rIRT_SFG), but did not match the corresponding field stored in the RLRT_SFG static relation.
Possible Error Effect:	Call processing will be unable to determine if an ISDN attendant console is associated with this SFG, and will be unable to perform manipulations on the busy lamp, warning lamp, and make-busy key for this console.
Corrective Action Taken:	The dynamic representation of this data (rISFSTAT) was corrected to match its static counterpart in rIRT_SFG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLGSFSTAT tuple.
<i>Bad Data =</i>	The isdntgbwl field found in the rLGSFSTAT tuple.
<i>Good Data =</i>	The isdntgbwl field found in the rIRT_SFG static tuple.
<i>Logical Index =</i>	The key (SF group number) of the rLGSFSTAT tuple.
<i>Dump =</i>	The rLGSFSTAT tuple.

Table 4 Error Code: LAMP

Error Code:	LAMP
Error Description:	This simulated facility group's analog and/or ISDN attendant console group busy lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of RLLGSFSTAT tuple.
<i>Bad Data =</i>	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	SF group number.
<i>Dump =</i>	RLLGSFSTAT tuple.

Table 5 Error Code: RANGE

Error Code:	RANGE
Error Description:	The idle count (rLGSFSTAT.idle_count) which is the count of idle SFG members could not be audited because the hunt type in the RLRT_SFG static relation (rIRT_SFG.hunt_type) was not marked properly (where rIRT_SFG.grpnum = rLGSFSTAT.grpnum). For small SFGs, the hunt type must be marked RTH_LGSFG. This error will fire until fixed by the CRAFT.
Possible Error Effect:	The count of idle members may become corrupt causing customers to get reorder treatment and call failures.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, read the RLRT_SFG relation with key set to the key which dumped as "good data" with this error code message. Modify the hunt_type field to RTH_LGSFG.
Dump Description:	
<i>Error Address =</i>	The address of the rLGSFSTAT tuple.
<i>Bad Data =</i>	The hunt_type field found in the rIRT_SFG static tuple.
<i>Good Data =</i>	The group number (rIRT_SFG.grpnum) of the rIRT_SFG static tuple.
<i>Logical Index =</i>	The key (SF group number) of the rLGSFSTAT tuple (rLGSFSTAT.grpnum).
<i>Dump =</i>	Nothing dumped.

Table 6 Error Code: SFGBL

Error Code:	SFGBL
Error Description:	The sfgbl field, which indicates if this SFG is associated with an analog console busy lamp, is a dynamic duplicate of the sfgbl field stored in static data (rIRT_SFG), but did not match the corresponding field stored in the RLRT_SFG static relation.
Possible Error Effect:	Call processing will be unable to determine if an analog console is associated with this SFG, and will be unable to perform manipulations on the busy lamp and make-busy key for this console.
Corrective Action Taken:	The dynamic representation of this data (rISFSTAT) was corrected to match its static counterpart in rIRT_SFG.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLGSFSTAT tuple.
<i>Bad Data</i> =	The sfqbl field found in the rLGSFSTAT tuple.
<i>Good Data</i> =	The sfqbl field found in the rIRT_SFG static tuple.
<i>Logical Index</i> =	The key (SF group number) of the rLGSFSTAT tuple.
<i>Dump</i> =	The rLGSFSTAT tuple.

Table 7 Error Code: TLAMP

Error Code:	TLAMP
Error Description:	This simulated facility group's threshold warning lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected, and the hardware state of the lamp was refreshed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of RLLGSFSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	SF group number.
<i>Dump</i> =	NA.

Table 8 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in RLLGSFSTAT (that is, RTH_LGSFG) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Hunt type of group as shown in RLRT_SFG.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	LGSFSTAT tuple that was deleted.

Table 9 Error Code: WARNTRESH

Error Code:	WARNTRESH
Error Description:	The warnthresh field indicates the customer selected value for the number of idle members a customer has still available when an ISDN attendant console warning lamp (is equipped) will light. This is a dynamic duplicate of the warnthresh field stored in the RLRT_SFG static relation.
Possible Error Effect:	Call processing will be unable to determine if an ISDN attendant console warning lamp should be ON or OFF, and the lamp may become stuck in an incorrect state, giving the customer incorrect SF group status.
Corrective Action Taken:	The dynamic representation of this data (rSFSTAT) was corrected to match its static counterpart in rIRT_SFG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLGSFSTAT tuple.
<i>Bad Data</i> =	The warnthresh field found in the rLGSFSTAT tuple.
<i>Good Data</i> =	The warnthresh field found in the rIRT_SFG static tuple.
<i>Logical Index</i> =	The key (SF group number) of the rLGSFSTAT tuple.
<i>Dump</i> =	The rLGSFSTAT tuple.

LGTKSTAT

Software Release: 5E10 and later

1. LARGE TRUNK STATUS (LGTKSTAT) AUDIT

The large trunk status audit (LGTKSTAT) checks the dynamic relation RLLGTKSTAT for extra tuples. To process a call, the RTA routing process in the communications module processor (CMP) and in a switching module (SM) must determine which ports are available for terminating calls. In the CMP, the RLLGTKSTAT relation contains tuples for all global ports that exist in more than one SM. During normal operation, when this relation resides in an SM it contains only tuples for local port groups that exist in that SM. During stand-alone operation, these tuples are deleted.

For information about smaller trunk groups, refer to the trunk status audit (TKSTAT). For information about groups of hunted lines, refer to the line status audit (LNSTAT) and the large line status audit (LGLNSTAT).

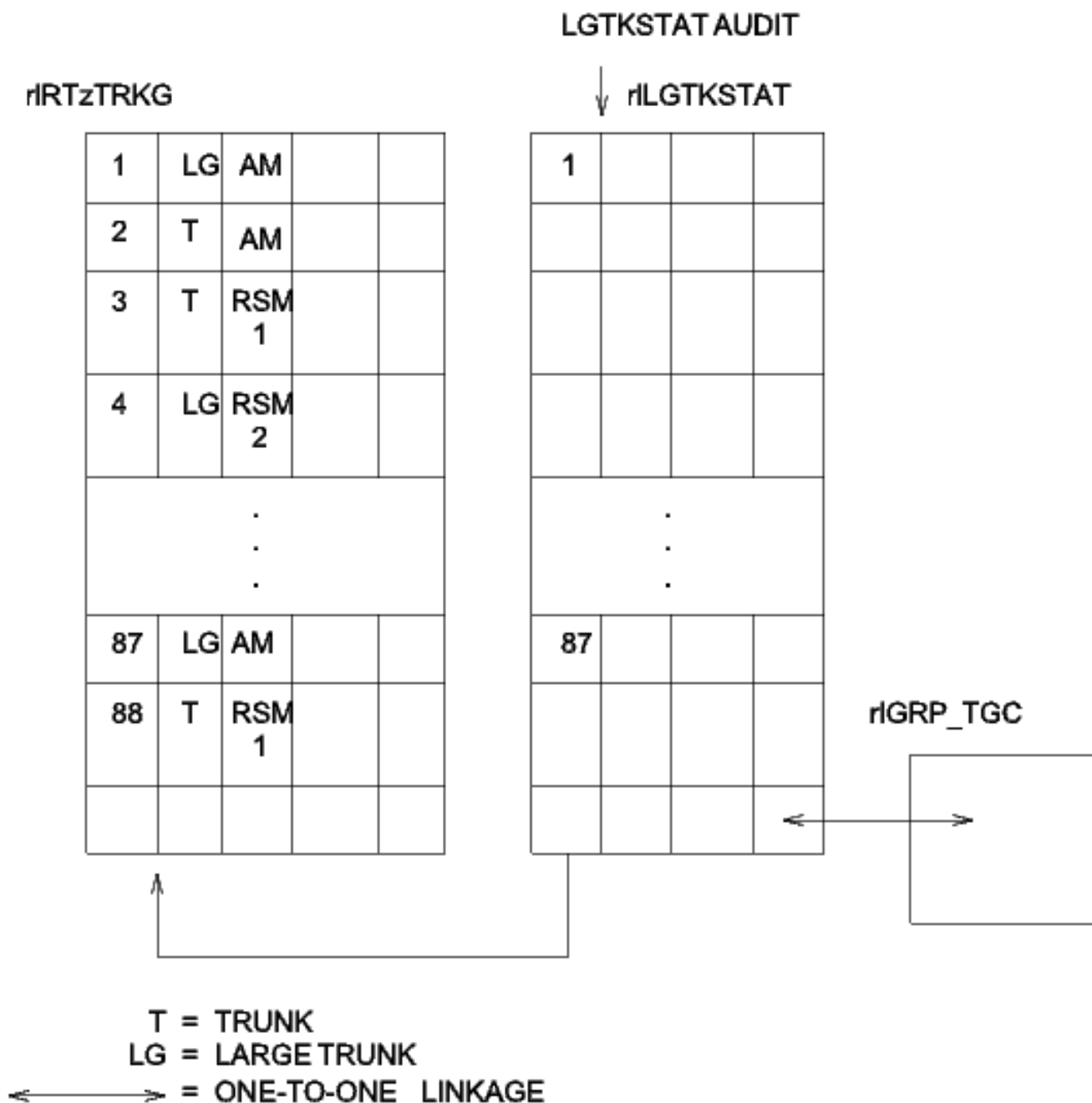
This audit is part of the routing and terminal allocation audits subgroup.

2. DATA STRUCTURES

The data structures checked by the LGTKSTAT audit are the large trunk status relation (RLLGTKSTAT) and the trunk group information relation (RLRT_TRKG). Figure 1 illustrates the two relations. The RLRT_TRKG relation is a static (write-protected) relation that contains information about all hunted trunks. This information includes a key (group number), the type of trunk group (two-way or incoming, and so forth), the module (CMP or SM) that controls it, and the hunt type. The LGTKSTAT audit steps through the RLLGTKSTAT relation and verifies it against the information contained in the RLRT_TRKG relation. The RLRT_TRKG relation is also used in the trunk status audit (TKSTAT), the trunk group information audit (RTTRKG), the trunk queue audit (TKQUE), and the trunk owner audit (TKOWNER).

SM & RSM*

AM



*In an RSM environment, the RLLGTKSTAT resides in both the AM and RSM.
 But in the RSM, the linkage between RLGTKSTAT and RLGRP_TGC does not exist.

Figure 1 (LGTKSTAT-1) Large Trunk Status Audit

3. AUDIT CHECKS

The LGTKSTAT audit calls the static driver (DRSTATIC) to check the correspondence between the RLLGTKSTAT and RLRT_TRKG relations and then performs the following checks:

- The audit verifies each rLGTKSTAT tuple against the corresponding rIRT_TRKG tuple by using the key (group number) of the rLGTKSTAT tuple to read the RLRT_TRKG relation. The audit looks specifically for extra rLGTKSTAT tuples.
- The audit checks the group type (RTTOWAY) and hunt type (RTH_LGBWD, RTH_LGFWD, RTH_LGUCD, RTH_LGGUCD, RTH_LGBGUCD, RTH_PLBLIN, or RTH_PLFLIN) of the rIRT_TRKG tuple.
- In the CMP, the audit also checks the link to the group trunk group controls relation (RLGRP_TGC).
- The audit ensures that the module number is correct for the tuple.
- If there is an analog console busy lamp associated with a trunk group, the audit compares its idle bit count with the automatic count made by call processing. In addition to the idle count (idle number of trunks), it checks that the status of the make busy key and busy lamp are consistent. If the make-busy key is processed, the lamp should be on regardless of the idle count.

If there is an ISDN attendant console busy lamp associated with a trunk group, the audit checks that the busy lamp and make-busy key of this console are consistent with the idle count (idle number of trunks). If the ISDN attendant console is equipped with a warning lamp, this lamp's state is checked against the idle count and the preset value in the trunk group information relation RLRT_TRKG. If the make-busy key is pressed, the busy and warning lamps should be on regardless of the idle-count.

4. ERROR HANDLING

Excess tuples, tuples not in the appropriate module, and tuples with wrong hunt types are deleted from the relation. If a linkage check fails, the audit zeroes the tuple and the tuple is reconstructed. The audit corrects counts and lamp errors.

The audit corrects ISDN attendant console busy lamp and warning lamp inconsistencies if a trunk group is equipped with such lamps.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	In the process of calculating the idle count (number of idle trunk members in the group as expressed by the number of bit positions in rLGTKSTAT.gmflg[] which are marked as "1"), the global trunk group tuple buffer AUtrkgbuf has been found with an invalid hunt type for this group (AUtrkgbuf.htype). This tuple buffer is used to store a local copy of the corresponding rIRT_TRKG static tuple associated with the rLGTKSTAT dynamic tuple under audit (where $rIRT\_TRKG.grpnum = rLGTKSTAT.grpnum$).
Possible Error Effect:	The idle count, console busy lamp and console warning lamp data (rLGTKSTAT.idle_count, rLGTKSTAT.tgbl_on, and rLGTKSTAT.tgwl_on) cannot be audited for correctness. Should these fields become corrupted, the customer would not receive proper trunk group status from the status lamps on

	the console monitoring this group.
Corrective Action Taken:	Audit processing is aborted for this group, the RTTRKG audit is scheduled, and in the process, the contents of the AUtrkgbuf global buffer are reinitialized.
Manual Action Required:	If this error continues to fire, it indicates that an ODD inconsistency exists. Using ODBE, examine the corresponding rIRT_TRKG static tuple for this group (where the key rIRT_TRKG.grpnum = rLGTKSTAT.grpnum). The key for the rIRT_TRKG static tuple is dumped as the logical key when this audit error fires. Once the rIRT_TRKG static tuple has been obtained, examine the hunt type (rIRT_TRKG.htype). It should reflect one of the legal LARGE LINEAR hunting algorithm hunt types. The legal LARGE LINEAR hunt types are listed in the RLRT_TRKG relation population rules, and are identified as RTH_LGXXX, where XXX denotes the particular hunting method (FW for forward, BW for backward, etc.) Using ODBE, correct the hunt type in RLRT_TRKG to match one of the LARGE LINEAR hunt types listed in the population rules.
Dump Description:	
<i>Error Address</i> =	The rITKSTAT tuple address.
<i>Bad Data</i> =	The hunt type that was found in the global trunk group tuple buffer (AUtrkgbuf.htype).
<i>Good Data</i> =	The trunk group number (key) that was found in the global trunk group tuple buffer (AUtrkgbuf.grpnum) for which there should exist a corresponding rIRT_TRKG static tuple, and rLGTKSTAT dynamic tuple.
<i>Logical Index</i> =	Group number (key) found in the rLGTKSTAT dynamic tuple (rLGTKSTAT.grpnum).
<i>Dump</i> =	The contents of the rLGTKSTAT dynamic tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	A count of the idle group members did not equal the stored idle count.
Possible Error Effect:	Measurements would be in error for this group. Possible mishandling of the group busy and warning lamps(emif equipped).
Corrective Action Taken:	The idle count was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rLGTKSTAT tuple address.
<i>Bad Data</i> =	idle_count field.
<i>Good Data</i> =	Correct count of idle members.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	rLGTKSTAT tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	rLGTKSTAT tuple.

Table 4 Error Code: LAMP

Error Code:	LAMP
Error Description:	This trunk group's analog and/or ISDN attendant console group busy lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of RLLGTKSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 = OFF; 1 = ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	rLGTKSTAT tuple.

Table 5 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the CMP, only tuples from global trunk groups should be present. In an SM, only tuples from local trunk groups on that SM should be present except during stand-alone operation when tuples from global trunk groups are allowed.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLGTKSTAT tuple.
<i>Bad Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	NA.

Table 6 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in rLGTKSTAT (that is, RTH_LGFWD, RTH_LGBWD, RTH_LGUCD, RTH_LGGUCD, RTH_LGBGUCD, RTH_PLFLIN, or RTH_PLBLIN) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLGTKSTAT tuple.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	NA.

Table 7 Error Code: TLAMP

Error Code:	TLAMP
Error Description:	This trunk group's ISDN attendant console threshold warning lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of RLLGTKSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	RLLGTKSTAT tuple.

20. LH - LP

LINE

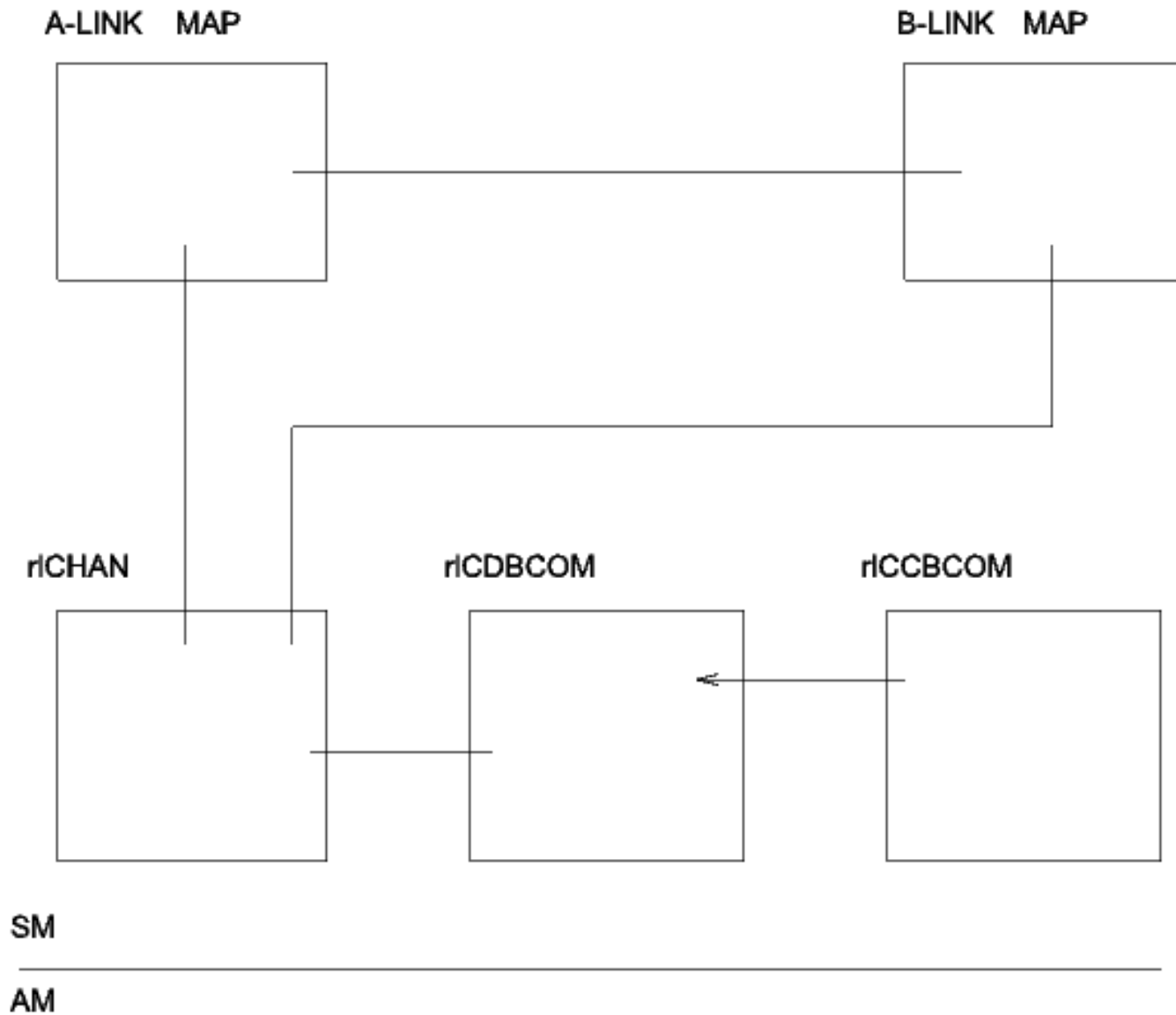
Software Release: 5E10 and later

1. LINE CONCENTRATOR (LINE) AUDIT

The line unit consists of a line concentrator and customer interface hardware that includes channel circuits, high-level service circuits, and control circuits. The line unit can terminate as many as 512 customer lines. It is divided into two service groups, each of which can accommodate 32 active calls. The line concentration hunts through a path to select a channel for a call when a customer originates. Through a series of A-links and B-links, the line concentrator provides concentration ratios of 10:1, 8:1, 6:1, or 4:1 customer lines to channel circuits, depending on the office configuration. There is one B-link for each channel circuit; in a fully equipped line unit, as many as eight B-links may terminate on one A-link. Only one channel circuit may be connected to the A-link at a time, however. The line concentrator (LINE) audit ensures the consistency of the software controlling the A-links, B-links, and channel circuits. The LINE audit is part of the peripheral control audits subgroup. It operates at execution level B in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structures verified by the LINE audit are the A-link allocation map, the B-link allocation map, the channel circuit relation (RLCHAN), the circuit data block common area relation (RLCDBCOM), and the channel control block common area relation (RLCCBCOM). The channel circuit relation defines the mapping between channel circuits and crosspoints. It tells whether each channel is busy or idle; if the channel is busy, RLCHAN tells which crosspoints form the path. The circuit data block common area (RLCDBCOM) relation defines the allocation of the circuit to processes, idle lists, and out-of-service lists. Figure 1 illustrates these data structures and linkages among them.



NOTE: A-LINK MAPS TO EIGHT B-LINKS.
 B-LINK CORRESPONDS TO CHANNEL CIRCUIT.
 EACH CHANNEL CIRCUIT HAS AN rCHAN TUPLE
 AND AN rCDBCOM TUPLE.
 rCCBCOM HAS HEAD CELLS FOR rCDBCOM
 TUPLES.

—————> = One-to-many linkage
 ————— = Nonstandard linkage

Figure 1 (LINE-1) Line Concentrator Linkages

3. AUDIT CHECKS

After determining the type of line units the switching module has (model LU1, LU2, or LU3), the audit performs two phases of semantic checks: the A-link phase and the B-link phase.

- A-link phase - The audit checks the peripheral control line unit service group array to determine whether the line unit is equipped. If it is not equipped, the audit checks the next line unit.
- (a) The audit obtains each entry in the A-link allocation map and checks the traffic busy/idle status of the A-link.
 - (b) The audit computes the eight possible B-link names that may connect on the A-link. It checks the maintenance status of each B-link. If the maintenance status is busy, the audit ignores it and proceeds to the next B-link.
 - (c) For each B-link, the audit obtains the channel number for the B-link within the specified service group. The audit calculates the circuit name and checks the appropriate rICHAN tuple to ensure that the channel traffic status and B-link status are the same (there is a one-to-one correspondence between B-links and channel circuits).
 - (d) The audit then determines whether the channel terminates on the proper A-link. If it does, the audit checks the busy/idle status of the rICHAN tuple and stores the count of the crosspoints that are busy. If the channel does not terminate on the proper A-link, the audit proceeds to the next channel.
 - (e) The audit checks the A-link traffic status against the busy count. A busy A-link should have a busy count of one. An idle A-link should have a busy count of zero.
- B-link phase - The audit checks the peripheral control line unit service group array to determine whether the line unit is equipped. If it is not equipped, the audit checks the next line unit.
- (a) If the line unit is equipped, the audit obtains each entry in the B-link allocation map and checks the traffic and maintenance busy/idle statuses of the B-link.
 - (b) The audit calculates the channel name for each B-link for accessing rCDBCOM and obtains the channel key for accessing rICHAN.
 - (c) The audit accesses the rCDBCOM tuple for the channel and checks the status of the channel circuit. If the channel circuit is unequipped or out of service, the rICHAN tuple should be vacant or busy, respectively, and the B-link should be marked busy. If the channel circuit is in service, the audit checks the consistency among rCDBCOM, rICHAN, and the B-link map based on the traffic busy/idle status in rCDBCOM.

4. ERROR HANDLING

The A-link phase of the LINE audit handles the following errors. If the A-link status does not agree with the busy count, an error is reported and the A-link status is set to idle. If the A-link status is idle and the busy count is greater than zero, the B-link status is set to idle and the corresponding rCDBCOM tuple is quarantined.

The B-link phase of the LINE audit handles the following errors. If a channel circuit is marked unequipped or out of service (as in the case for growth) in an rCDBCOM tuple and the B-link status or rICHAN tuple is not vacant or is

marked idle, respectively, an error is reported and the B-link allocation map or rICHAN tuple is set busy. If the busy/idle status of the B-link in the rICHAN tuple does not agree with the B-link allocation map status or the B-link is in an invalid state (for example, maintenance busy and traffic idle), an error is reported and the B-link status and the rICHAN tuple are set to idle. If the rICHAN tuple indicated that the status was busy, the corresponding rCDBCOM tuple is quarantined to ensure error confinement and recovery by the circuit data audit (CKTDATA).

The LINE audit schedules the CKTDATA and CHAN audits as related audits when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALINK

Error Code:	ALINK
Error Description:	The traffic busy/idle status of an A-link is inconsistent with the traffic busy/idle status of the corresponding channel circuits.
Corrective Action Taken:	Idle the A-link map traffic status bit. Idle the corresponding B-link. Quarantine the rCDBCOM tuple of the channel circuit corresponding to the B-link.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of an rICHAN tuple which could connect to the A-link in a LU1/LU2/LU3 line unit.
<i>Bad Data</i> =	A-link status.
<i>Good Data</i> =	Count of busy first stage crosspoints.
<i>Logical Index</i> =	Logical key of the rICHAN tuple mentioned under ERROR ADDRESS. All of the following dumps are printed in one dump call. Within this large dump, the mini dumps are separated by the hex string FACE.
<i>Dump 1</i> =	Line unit number.
<i>Dump 2</i> =	A-link map for the line unit.
<i>Dump 3</i> =	B-link map for the line unit.
<i>Dump 4</i> =	rICHAN tuples which can connect to the A-link in a LU1/LU2/LU3 line unit.

Table 2 Error Code: ASSOC_CDB

Error Code:	ASSOC_CDB
Error Description:	An rCDBCOM tuple associated with a B-link appears to be lost or has been corrupted.
Possible Error Effect:	The status of the B-link cannot be verified. Possible lost call processing resource.
Corrective Action Taken:	Schedule the CKTDATA audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rCDBCOM tuple.
<i>Bad Data</i> =	Key of rCDBCOM tuple at DA supplied address.
<i>Good Data</i> =	Calculated rCDBCOM tuple key.
<i>Logical Index</i> =	Calculated channel circuit name.
<i>Dump</i> =	rCDBCOM tuple.

Table 3 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	An rICHAN tuple associated with a B-link appears to be lost or has been corrupted.
Possible Error Effect:	The status of the B-link cannot be verified. Possible lost call processing resource.
Corrective Action Taken:	Schedule the CHAN audit as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHAN tuple.
<i>Bad Data</i> =	Key of rICHAN tuple at DA supplied address.
<i>Good Data</i> =	Calculated rICHAN tuple key.

<i>Logical Index</i> =	Calculated channel circuit name.
<i>Dump</i> =	rICHAN tuple.

Table 4 Error Code: BLINK

Error Code:	BLINK
Error Description:	The busy/idle status of a B-link is inconsistent with the busy/idle status of the corresponding channel circuit (rICHAN) or circuit data (rCDBCOM) status. Another possible cause is that the B-link was found to be in the illegal state "MTCE busy - TRAFFIC idle".
Possible Error Effect:	Available B-links may not be used by call processing due to the incorrect map.
Corrective Action Taken:	Idle the B-link map status bit, and quarantine the rCDBCOM tuple if not on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rICHAN tuple.
<i>Bad Data</i> =	B-link traffic status.
<i>Good Data</i> =	rICHAN status bit.
<i>Logical Index</i> =	Logical key of the rICHAN tuple. Each dump is separated by a marker (FACE).
<i>Dump 1</i> =	Line unit.
<i>Dump 2</i> =	B-link map.
<i>Dump 3</i> =	rICHAN tuple.
<i>Dump 4</i> =	rCDBCOM tuple.

Table 5 Error Code: OOS_BLINK

Error Code:	OOS_BLINK
Error Description:	A channel circuit is out-of-service or unequipped, but the B-link map is idle.
Possible Error Effect:	An out-of-service channel may be allocated to a call.
Corrective Action Taken:	Busy the B-link map.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCDBCOM tuple.
<i>Bad Data</i> =	B-link status.
<i>Good Data</i> =	AUBLNKBUSY (1).
<i>Logical Index</i> =	Logical key of rICHAN tuple. Each dump is separated by a marker (FACE).
<i>Dump 1</i> =	Line unit.
<i>Dump 2</i> =	PCblunit[] for this line unit.
<i>Dump 3</i> =	Dump of the rCDBCOM tuple.

Table 6 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Line unit number (0-7).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcursor.
<i>Dump</i> =	NA.

LKBDST

Software Release: 5E10 and later

1. LINK AND BAND STATUS (LKBDST) AUDIT

The LKBDST audit is run automatically several times a day (depending on the number of equipped nodes). It can also be run manually, as described under the AUD:LKBDST listing in the Input Messages manual (235-600-700). The purpose of the audit is to detect and correct any discrepancies between basic link and band status data stored on the central processor and the nodes.

2. DATA STRUCTURES

This information is maintained by both the central processor and the nodes. It includes:

- link state
- node state
- internal link and mate link status (indicating whether or not the link or its mate are carrying traffic)
- route switch status (reflecting band status)
- other node parameters and indicators

If the link is of type CCS7 and has virtual links assigned, the data checked includes only ECIS6 data (that is, point code routing data, CCS7 node parameters, and CCS7 link status data are not checked). If the node is a private branch exchange (PBX) type, the data includes per node data and per link data.

3. AUDIT CHECKS

In automatic mode, each equipped node (other than RPC nodes) is audited in sequence until all nodes in the office have been checked. Each time LKBDST executes, one node is audited. Upon the next invocation, the audit continues with the next node. A node is skipped if it is out-of-service. In manual mode, either a specific node or all nodes can be audited.

The central processor sends a query to the node being audited. The node responds by sending the basic link and band status data to the central processor to be compared with its version of the data. From one to three messages are required.

The data audited is both correctable and non-correctable. Correctable data includes:

- link states
- route switch status

If a correctable error is found, the central processor sends another message to the node containing the correct data. Any one error causes all correctable data to be updated. If the node is a PBX type, all D-channel (DCHN) and special access data channel (SADC) links may be restarted.

Non-correctable data includes:

- link status
- mate link status

- node states

If anon-correctable error is found, the node is restarted (that is, it is diagnosed and downloaded).

The audit usually completes 15 minutes to two hours after it is started, depending on the number of nodes in the office and on how busy it is at the time.

NOTE: If the audit is executing manually on all nodes, the audit aborts if more than two nodes must be restarted. This is to avoid removing a large number of nodes from service at one time.

4. ERROR HANDLING

Output messages are found in the Output Messages manual (235-600-750). If the audit is running manually or if the detailed (DETL) option was selected, up to 16 detailed reports are provided. If more than 16 errors are found, a modified form of the raw data report shows the number of errors found as well as the number corrected.

LLCB-A

Software Release: 5E10 - 5E11

1. LOGICAL LINK CONTROL BLOCK (LLCB) AUDIT

The LLCB audit is part of the switch maintenance audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The LLCB audit uses the packet switching logical link control block array (PSllcb), the D-channel control block (PSdccb), and the internal frame descriptor array (PSifd).

3. AUDIT CHECKS

The audit makes linkage checks within the PSllcb array and with PSdccb and PSifd. It makes semantic checks to ensure the correctness of data in PSllcb.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ACPTRBAD

Error Code:	ACPTRBAD
Error Description:	The audit found a bad ALCB pointer. A LAPD LLCB is on the timing or timeout list, but its ALCB index is out of range.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPD LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's alcb_index field.
<i>Good Data =</i>	PSALCBNULL, the alcb_index value is greater than this.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 2 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The audit found a LAPB or LAPD LLCB with IFDs that have been monitored by the audits for at least 2 audit cycles and have not moved, yet the LLCB is not in peer busy state (the "peer-busy" bit is not set). This is a potential stuck condition that may put the PH in overload where the IFDs holding BDs are not moving therefore tying up these resources.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB and all related data structures to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's timer type "ttype".
<i>Bad Data =</i>	LLCB's frame age counter "frmag_cnt".
<i>Good Data =</i>	LLCB's channel number "dchan".
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	LLCB being audited.

Table 3 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The audit found a RRL or LAN LLCB whose owner field is invalid.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAN or RRL LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's owner field.
<i>Good Data =</i>	PSLLNOLST - the highest valid owner value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 4 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	Flow Control LLCBs are those LLCBs that fall with the range of PSSLSIX-PSSLLEIX, and whose type field has a value of PSL_FLWCTRL. For an FCCB LLCB, the phctype field did not match the same field found in DCDB accessed through the dchan field of this LLCB which serves as an index to PSdcbd[] array.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will correct the phctype field of this LLCB to be equal to the phctype field value found in DCDB accessed through the dchan field of this LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's dchan field.
<i>Bad Data =</i>	LLCB's phctype field.
<i>Good Data =</i>	The value of phctype field found in DCDB accessed through the dchan field of this LLCB.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 5 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The OCstrt_time[] array holds time stamps for busy channels and is used for the LAPB and LAPD LLCBs. When audit counted all the non-null entried in the OCstrt_time[] and checked it against the global counter, <i>OCbusy_cnt</i> , there was a mismatch.
Possible Error Effect:	The PH can potentially be in overload forever. Any non-zero value in the global counter <i>OCbusy_cnt</i> will force the PH to be in overload.
Corrective Action Taken:	The audit corrects the <i>OCbusy_cnt</i> counter with the count of non-zero entried in the OCstrt_time[] array it calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value found in <i>OCbusy_cnt</i> .
<i>Good Data =</i>	The value found in au_count, local variable to hold the count of non zero entries of the OCstrt_time[] array.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 6 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit found a LAPD LLCB on timing or on the timeout list, but it is not linked to any DCCB or any of the field(s), dchan, tix or sapi i is(are) bad.
Possible Error Effect:	Prints out a report and recover the LLCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will return the LAPD LLCB to the free linked list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's type field.
<i>Bad Data =</i>	LLCB's sapi_i field. Although in this field we have chosen to dump the sapi_i value as bad data, value of tix and/or dchan could also be bad. And they are dumped in "DUMP" field.
<i>Good Data =</i>	PSLLCBNULL - the valid null value for this field. Actually for valid link, the values of dchan, tix and sapi_i should be within range and should have such values that they form a link with LLCB.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The value of tix and dchan. This combined with "BAD DATA" (sapi_i) will determine which value(s) is(are) bad.

Table 7 Error Code: BLINK

Error Code:	BLINK
Error Description:	"Blocked IFD link". The audit found a LLCB whose "blocked" IFD field is out of range.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's IFD blocked field (blocked).
<i>Good Data =</i>	PSIFDNULL - valid null value for blocked field.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 8 Error Code: BUSYCHAN

Error Code:	BUSYCHAN
Error Description:	The OCstr_ttime[] array holds time stamps for busy channels and is used for the LAPB and LAPD LLCBs. The audit found that for the current LLCB being audited, the entry in the OCstr_ttime[] array has a valid timer value, that is not null and less than PSTMAX, but the own_busy bit is not set.
Possible Error Effect:	The PH can potentially be in overload forever as this field directly interfaces with the global <i>OCbusy_cnt</i> counter.
Corrective Action Taken:	The audit corrects the OCstr_ttime[] array entry to null, and returns AUFALSE to recover this LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index to the OCstr_ttime[] array. This is derived by subtracting PSSLLSIX value from the LLCB index dumped in the logical key field below.
<i>Bad Data =</i>	LLCB's own_busy field.
<i>Good Data =</i>	Value found in the OCstr_ttime[] array for this LLCB.
<i>Logical Index =</i>	LLCB index, the LLCB being audited.
<i>Dump =</i>	The LLCB being audited.

Table 9 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The audit found a LAN LLCB with stuck IFDs linked to it. The audit found no movement of IFDs from the LLCB for two audit cycles.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAN LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB timer type ("ttype").
<i>Bad Data =</i>	LLCB's frame age counter. This is bumped each time the audit finds non-moving IFDs. The maximum value is PSIFDMXCTR or 2.
<i>Good Data =</i>	LLCB's owner.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	LLCB's state.

Table 10 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The LLCB audit found a LAPB LLCB on timing or timeout with the DCCB indicated in the LLCB's dchan field is NOT pointing to this LLCB.
Possible Error Effect:	Prints out a report and recovers the LLCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The starting address of this LLCB.
<i>Bad Data =</i>	Any of the values of dchan, tix or sapi_i are bad for this LLCB. Value of sapi_i is dumped here. Values of dchan and tix are dumped in "DUMP".
<i>Good Data =</i>	Any valid value of DCCB index combined with the tix value of 0 and sapi_i value of PSKT (2) to give a valid link for this LLCB with DCCB.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The value of tix and dchan. The combination of these values with sapi_i value will determine which values are bad.

Table 11 Error Code: DRLPTR

Error Code:	DRLPTR
Error Description:	"Data ready link PTR". The audit found a LLCB whose "ready" IFD field is out of range.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's IFD ready field (ready).
<i>Good Data =</i>	PSIFDNULL - valid null value for ready field.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 12 Error Code: HEAD

Error Code:	HEAD
Error Description:	The audit found a LLCB (the IFD headcell) whose IFD linkages are bad. The check that the first IFD's owner is this LLCB, and the first IFD's back field is PSIFDNULL, and the last IFD's owner is this LLCB, and the last IFD's back field is PSIFDNULL has failed.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which causes the LLCB to be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	PSIFDNULL - valid null value for the back or forward IFD fields.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	AULNK defined in au/AUaps/AUextern.h (all sized as unsigned short): • "back" element's back field, • "back" struct.'s owner field, • "back" struct.'s next field, • current element being audited, • current element on this linked list, • current element's back field,

	• current element's owner, • data element null value, • head cell null value, • headcell's first element, • headcell's last element, • "next" element on this linked list, • next element's back field, • next element's owner field. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and so is filled with 0xaa.

Table 13 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The LLCB audit found a LAPB LLCB on no linked list whose forward and/or backward LLCB fields are not null.
Possible Error Effect:	Prints an error report and corrects the fields.
Corrective Action Taken:	The audit reports the error and sets the forward and backward LLCB fields to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's forward field (forw .llcb).
<i>Bad Data =</i>	LLCB's back field (back .llcb).
<i>Good Data =</i>	PSTLNULL - valid null field values.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 14 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	The audit found a LAPD LLCB on no linked list and either the forward LLCB pointer or the back LLCB pointer is not null. There should be no LLCBs linked to this LLCB.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPD LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's forward LLCB pointer.
<i>Bad Data =</i>	LLCB's back LLCB pointer.
<i>Good Data =</i>	PSTLNULL, the valid null value for these fields.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 15 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	"Error in no list owner". The audit found a intra-PH with its owner not set to no linked list.
Possible Error Effect:	Prints an error report and recovers the intra-PH LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which sets the intra-PH LLCB back to its default values.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's owner field.
<i>Good Data =</i>	PSLLNOLST - valid owner for the Intra-PH LLCB.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 16 Error Code: IL_PTYPE

Error Code:	IL_PTYPE
Error Description:	LAN LLCB is defined as an LLCB falling within the range of PSPLLSIX - PSPLLEIX. For a LAN LLCB, which is not an intra-ph LLCB, that is LLCB equal to PSINPHLL for PH processors; the type field was found not to be PSL PKTBUS, which is an error.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit reports the error and sets the LLCB back to its default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's type field.
<i>Good Data =</i>	PSL PKTBUS - valid value for type field for the LAN LLCBs.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 17 Error Code: INDEX

Error Code:	INDEX
Error Description:	"Invalid PH channel index". The audit found a LAPD LLCB whose "phctype" does not match the write protected D-channel data block's (DCDB) "phctype" field.
Possible Error Effect:	Prints an error report and corrects the LLCB's "phctype".
Corrective Action Taken:	The audit reports the error and sets the LLCB's "phctype" to the DCDB's "phctype" field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's phctype field.
<i>Good Data =</i>	DCDB's phctype field (write protected data).
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 18 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The OCstrt_time[] array holds time stamps for busy channels and is used for the LAPB and LAPD LLCBs. Audit found that for the current LLCB being audited, the entry in the OCstrt_time[] array does not have a null value(OCTNULL) and is out of range, that is, greater than PSTMAX.
Possible Error Effect:	The channel cannot be busy for more than 1 second. This channel will be released in one second.
Corrective Action Taken:	The audit corrects the OCstrt_time[] array entry to null value, updates the own_busy field of the LLCB to DBNO and returns AUFALSE to recover this LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index to the OCstrt_time[] array. This is derived by subtracting PSSLLSIX value from the LLCB index dumped in the Logical Key field.
<i>Bad Data =</i>	Value found in the OCstrt_time[] array for this LLCB.
<i>Good Data =</i>	LLCB's own_busy field.
<i>Logical Index =</i>	LLCB index, the LLCB being audited.
<i>Dump =</i>	NA.

Table 19 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The audit has found a resource recovery list LLCB where there is no IFD on the RRL LLCB but the owner is not PSLLNOST. There are no IFDs there should not be any timers and the LLCB should be on no linked list.
Possible Error Effect:	Prints out a report and recovers the RRL LLCB.
Corrective Action Taken:	The audit returns AUFALSE which will recover the LLCB and set the RRL LLCB to default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's d channel (dchan).
<i>Bad Data =</i>	LLCB's owner field.
<i>Good Data =</i>	LLCB's state.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 20 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The audit found a RRL or Intra-PH LLCB with an invalid owner field.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's owner.
<i>Good Data =</i>	LLCB index (NA here).
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 21 Error Code: INVOWNER2

Error Code:	INVOWNER2
Error Description:	The audit has found an invalid owner field for LAPD LLCB.
Possible Error Effect:	Prints out a report and recovers the LLCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's owner.
<i>Good Data =</i>	LLCB index.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 22 Error Code: IPBID

Error Code:	IPBID
Error Description:	"Invalid peer busy indicator". The audit found a LAPB LLCB with an inconsistent "peerbusy" field for its state.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's state field.
<i>Bad Data =</i>	LLCB's peer_busy field.
<i>Good Data =</i>	The expected peer_busy value for this state.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 23 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The audit found a D-channel index field is not null. Neither LAN nor RRL LLCBs should be linked to D-channel control blocks (DCCBs).
Possible Error Effect:	Prints an error report and nulls the field.

Corrective Action Taken:	The audit corrects the LLCB's dchan field to its null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's dchan field (D-channel).
<i>Good Data =</i>	PSMAXCHN - the correct dchan null value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 24 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The audit found a LAN LLCB that was not on timing, but its forward LLCB pointer or back LLCB pointer is not null.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAN LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB forward pointer to next LLCB.
<i>Bad Data =</i>	LLCB back pointer to previous LLCB.
<i>Good Data =</i>	PSTLNULL - the correct LLCB pointer null value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 25 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	The LLCB audit found a LAPB LLCB on timing but has either an out of range ALCB or the ALCB indicated in the LLCB is not pointing to this LLCB.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's alcb_index.
<i>Bad Data =</i>	LLCB's ALCB's llcb_index.
<i>Good Data =</i>	PSALCBNULL - null ALCB value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 26 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	The audit found a LAPB or LAPD LLCB on either the free linked list or NO list where either the ALCB index field or the D-channel field is not null. This LLCB should not be linked to any DCCBs or have any ALCBs linked to it.
Possible Error Effect:	Prints an error report and corrects the erroneous fields.
Corrective Action Taken:	The audit reports the error and sets the alcb_index and dchan (D channel) fields to null values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's pointer to D channel (dchan).
<i>Bad Data =</i>	LLCB's ALCB pointer (alcb_index).
<i>Good Data =</i>	PSALCBNULL - the valid null value for the alcb_index field.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 27 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	The audit found a resource recovery list LLCB or LAN LLCB whose ALCB index field is not null. There should not be any ALCBs linked to RRL or LAN LLCBs.
Possible Error Effect:	Prints an error report and nulls the field.
Corrective Action Taken:	The audit corrects the LLCB's alcb_index field to its null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	LLCB's alcb index.
<i>Good Data</i> =	PSALCBNULL - the correct ALCB index null value.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 28 Error Code: LNKILST

Error Code:	LNKILST
Error Description:	"Link to IFD idle list". The audit found a LLCB with the first and last IFD fields null but the ready or blocked fields are NOT null.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE to recover the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's ready IFD.
<i>Bad Data</i> =	LLCB's blocked IFD.
<i>Good Data</i> =	LLCB index.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 29 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The audit has found a LAPD LLCB not in multiframe establish or timer recovery state and the peer_busy bit is set. These are the only valid states for the bit to be set.
Possible Error Effect:	Prints out a report and correct the peer_busy field.
Corrective Action Taken:	The audit will report the error and set peer_busy bit to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's owner (indicates on timing or no link list).
<i>Bad Data</i> =	LLCB's state.
<i>Good Data</i> =	dchan - the channel the LLCB is on.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 30 Error Code: MAINT_STATE

Error Code:	MAINT_STATE
Error Description:	The audit found a LAPB LLCB with an out of range state field.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's state field.
<i>Good Data</i> =	LLCB index.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 31 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The audit found a LAPB LLCB on no linked list and the point-to/point-back checks to the DCCB failed.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's D-channel value (dchan).
<i>Bad Data</i> =	LLCB's DCCB's llcb index.
<i>Good Data</i> =	LLCB index (NA here).
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 32 Error Code: MASTER_OWN

Error Code:	MASTER_OWN
Error Description:	The audit found a LAPB LLCB on no linked list and the D-channel field is out of range.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's D-Channel field (dchan).
<i>Good Data =</i>	PSMAXCHN, one or more than the maximum valid value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 33 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	The audit found a member error in the linked list of the specified audit.
Possible Error Effect:	Prints an error report and corrects the linked list.
Corrective Action Taken:	Returns AUFALSE to calling function which rebuilds the linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Linked list null value.
<i>Logical Index =</i>	AUcount, the current element number.
<i>Dump =</i>	AULNK defined in au/AUps/AUextern.h (all sized as unsigned short): • "back" element's back field, • "back" struct.'s owner field, • "back" struct.'s next field, • current element's owner, • current element on this linked list, • current element's back field, • current element being audited, • data element null value, • head cell null value, • headcell's first element, • headcell's last element, • "next" element's owner field, • "next" element on this linked list, • "next" element's back field. NOTE: 0xaa is the default value and indicates either the field is not used or

the values are out of range and the data cannot be read and so is filled with 0xaa.

Table 34 Error Code: NO_OWN_TS

Error Code:	NO_OWN_TS
Error Description:	"No owner is indicated for timing specified". The audit found an EDSL channel that does not have a timer pending.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAN LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's owner field.
<i>Good Data =</i>	PSTMAX - One greater than the maximum value of the owner field.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 35 Error Code: NOACCBLK

Error Code:	NOACCBLK
Error Description:	"No accurate block IFD field". The audit found a LLCB where the point-to-point-back checks fail between the "blocked" IFD and the LLCB.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's IFD blocked field (blocked).
<i>Bad Data =</i>	Blocked IFD's LLCB field (should equal this LLCB).
<i>Good Data =</i>	LLCB index.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 36 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The audit found a LAN or RRL LLCB with its own busy bit set. Neither LAN nor RRL LLCBs should have the own_busy bit set.
Possible Error Effect:	Prints an error report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the "own_busy" bit to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's own_busy value.
<i>Good Data =</i>	DBNO - valid value for the own_busy bit.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 37 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The audit found an unused LLCB without the required fields nulled.
Possible Error Effect:	Prints an error report and nulls the LLCB.
Corrective Action Taken:	The LLCB is zeroed out and returns from the function for more corrective actions.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's DCHAN (pointer to parent DCCB).
<i>Bad Data =</i>	LLCB's ALCB index.
<i>Good Data =</i>	LLCB index (not applicable here).
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	(All sized as unsigned short.)
	• LLCB's owner.

	• LLCB's back pointer to the previous LLCB. • LLCB's forward pointer to the next LLCB.

Table 38 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	Flow control LLCBs are those LLCBs that fall with the range of PSSLLSIX-PSSLLEIX, and whose type field has a value of PSL_FLWCTRL. FCCB LLCBs are never attached to an ALCB, hence the alcb_index field of this LLCB should always be null (such as, PSHMAXALCB). A non null alcb_index field for this LLCB was found.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit reports the error and returns the LLCB back to its free list. A routine port condition of the PH involved will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The dchan field of the LLCB.
<i>Bad Data =</i>	LLCB's alcb_index field.
<i>Good Data =</i>	PSHMAXALCB - valid value for alcb_index field for this FCCB LLCB.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 39 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	For an LLCB linked to the free list, that is the owner field of this LLCB has a value of PSLLFRLST, the type field was found not to be PSL_NULL, which is an error.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit reports the error and returns the LLCB back to its free list. A routine port condition of the PH involved will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's owner field.
<i>Bad Data =</i>	LLCB's type field.
<i>Good Data =</i>	LLCB's dchan field.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 40 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	The audit found a LAPD LLCB on the timing or timeout list and has a valid ALCB index; however, the ALCB is not pointing to the LLCB.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPD LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's ALCB index field (alcb_index).
<i>Bad Data =</i>	LLCB's ALCB's llcb_index field.
<i>Good Data =</i>	LLCB index (NA here).
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 41 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit found a LLCB with either its next LLCB pointer or previous LLCB pointer out of range or not the expected null value.
Possible Error Effect:	Prints an error report and recovers the LLCB.

Corrective Action Taken:	The audit reports the error and returns AUFALSE which causes the LLCB to be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting range that the LLCB can reside in.
<i>Bad Data</i> =	Ending range that the LLCB can reside in.
<i>Good Data</i> =	nullval - valid null value that the next or previous LLCB pointer can have.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	AULNK defined in au/AUps/AUextern.h (all sized as unsigned short): • "back" element's back field, • "back" struct.'s owner field, • "back" struct.'s next field, • current element being audited, • current element on this linked list, • current element's back field, • current element's owner, • data element null value, • head cell null value, • headcell's first element, • headcell's last element, • "next" element on this linked list, • "next" element's back field, • "next" element's owner field. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and so is filled with 0xaa.

Table 42 Error Code: OWNER

Error Code:	OWNER
Error Description:	The audit found a LAPB LLCB with an invalid owner field.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's owner field.
<i>Good Data</i> =	LLCB index.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 43 Error Code: OWNLINK

Error Code:	OWNLINK

Error Description:	The audit found a LLCB whose owner field is invalid while checking the LLCB on a free, timer or timeout linked list.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's owner field.
<i>Good Data =</i>	LLCB index.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 44 Error Code: PARENT

Error Code:	PARENT
Error Description:	The LLCB audit found a LAPB LLCB on timing or timeout list whose "dchan" field is out of range.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit will report the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's starting address.
<i>Bad Data =</i>	LLCB's D-channel (dchan).
<i>Good Data =</i>	Any value between 0 and 127 will be a good data.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 45 Error Code: PKT_TYPE

Error Code:	PKT_TYPE
Error Description:	At the point we are reporting the error, we are expecting only LAPB, LAPD and flow control LLCBs determined from the type field and the function we are in. But we found an LLCB whose type field does not indicate it to be LAPB, LAPD or flow control LLCB.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Since this LLCB has to be either LAPB, LAPD or flow control LLCB, its type must have been set incorrectly. The audit reports the error and returns the LLCB back to its free list. A routine port condition of the PH involved will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's type field.
<i>Good Data =</i>	LLCB's dchan field.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 46 Error Code: PNT2PNT

Error Code:	PNT2PNT
Error Description:	Flow control LLCBs are those LLCBs that fall with the range of PSSLLSIX-PSSLLEIX, and whose type field has a value of PSL_FLWCTRL. For an FCCB LLCB, the point to point back check using the dchan field failed. The check involved getting an index in to the PSfccb[] array using the dchan field and checking the value of the LLCB index found at that array index in the PSfccb[] array to be equal to the index of the of the LLCB being audited.
Possible Error Effect:	Any EOC channels using this FCCB LLCB for its work may fail as this FCCB LLCB has not been initialized properly or has been corrupted.
Corrective Action Taken:	The audit reports the error and returns the LLCB back to its free list. A routine port condition of the PH involved will be scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The dchan field of the LLCB.

<i>Bad Data</i> =	The value of LLCB index found in the PSfccb[] array.
<i>Good Data</i> =	The LLCB index, that we should have found in the PSfccb[] array.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 47 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The audit has found a LAN LLCB whose D-channel index field is not null. This LLCB should not be linked to D-channel control blocks (DCCBs).
Possible Error Effect:	Prints out a report and nulls the field.
Corrective Action Taken:	The audit will correct the LLCB's dchan field to its null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's dchan field (D-channel).
<i>Good Data</i> =	PSMAXCHN - the correct dchan null value.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 48 Error Code: PRC

Error Code:	PRC
Error Description:	"Pointer corrupted". The audit found a LLCB whose first IFD, last IFD, ready IFD or blocked IFD is out of range.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's first IFD.
<i>Bad Data</i> =	LLCB's last IFD.
<i>Good Data</i> =	PSRIFDS - One more than maximum value the IFD value can be.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	(All sized as unsigned short.)
	• LLCB's IFD ready field. • LLCB's IFD blocked field.

Table 49 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The audit found a LAN or RRL LLCB with its peer_busy bit set. Neither LAN nor RRL LLCBs should have the peer_busy bit set.
Possible Error Effect:	Prints an error report and corrects the field.
Corrective Action Taken:	The audit reports the error and sets the "peer_busy" bit to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's state.
<i>Bad Data</i> =	LLCB's peer_busy value.
<i>Good Data</i> =	DBNO - valid value for the own_busy bit.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 50 Error Code: QUAR

Error Code:	QUAR
Error Description:	The audit found a LAN or RRL LLCB with its quarantine bit set.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAN or RRL LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.

<i>Bad Data</i> =	LLCB's quarantine bit value (quar_bit).
<i>Good Data</i> =	DBNO - valid value for the quarantine bit indicating no problems.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 51 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The audit found a LAPB/LAPD LLCB with its quarantine bit set.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB/LAPD LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's quarantine bit value (quar_bit).
<i>Good Data</i> =	DBNO - valid value for the quarantine bit indicating no problems.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	None.

Table 52 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	The audit found a LLCB where the point-to/point-back checks failed between the ready IFD and the LLCB.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's IFD ready field (ready).
<i>Bad Data</i> =	Ready IFD's llcb field (should equal this LLCB).
<i>Good Data</i> =	LLCB index.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 53 Error Code: RRLINK

Error Code:	RRLINK
Error Description:	The audit found a resource recovery list LLCB whose owner is not on timing; specifically the owner is found to be NO LIST.
Possible Error Effect:	Prints an error report and recovers the RRL LLCB.
Corrective Action Taken:	The audit returns AUFALSE which recovers the LLCB and puts the RRL LLCB back on timing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA. This field is used to dump values of "dchan" field of this LLCB.
<i>Bad Data</i> =	LLCB's owner field.
<i>Good Data</i> =	NA. This field is used to dump value of "state" field of this LLCB.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 54 Error Code: SAPIBAD

Error Code:	SAPIBAD
Error Description:	The audit has found a LAPB or LAPD LLCB either on timing or timeout list whose sapi_i field is out of range.
Possible Error Effect:	Prints out a report and recovers the LLCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's starting address.
<i>Bad Data</i> =	LLCB's sapi_i field.
<i>Good Data</i> =	PSPKT value.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 55 Error Code: STATE

Error Code:	STATE
Error Description:	The audit found a LAPD LLCB's state field out of range.
Possible Error Effect:	Prints an error report and recovers the LAPD LLCB.
Corrective Action Taken:	The audit reports the error and returns the LAPD LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's state field.
<i>Good Data =</i>	PSDS_SXTGI - maximum state value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 56 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The audit found an intra-PH with its state not set to 0.
Possible Error Effect:	Prints an error report and recovers the intra-PH LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which sets the intra-PH LLCB back to its default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LLCB's state field.
<i>Good Data =</i>	LLCB index.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 57 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The audit found a LAPB LLCB in a S5, S6, S7, S9, S12 or S15 state and its "own_busy" bit is set to DBYES. This bit should not be set while in any of these states.
Possible Error Effect:	Prints an error report and corrects the LLCB field.
Corrective Action Taken:	The audit reports the error and calls PSL2CLRBUSY to clear the "own_busy" bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's state.
<i>Bad Data =</i>	LLCB's own_busy field.
<i>Good Data =</i>	DBNO - the valid value of the "own_busy" bit.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	NA.

Table 58 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The audit found a LAPB LLCB in a state other than S1, S5 or S8 but it is not on timing. The owner field of the LLCB must indicate that it currently has a timer running.
Possible Error Effect:	Prints an error report and recovers the LAPB LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB LLCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's state.
<i>Bad Data =</i>	LLCB's owner field (index into timer array).
<i>Good Data =</i>	PSTMAX - One more than the maximum value that owner field should have.
<i>Logical Index =</i>	LLCB index.
<i>Dump 1 =</i>	The LLCB.
<i>Dump 2 =</i>	If LLCB.alcb_i is in range, the associated ALCB's al_rst_state field will be dumped.

Table 59 Error Code: TIX

Error Code:	TIX
Error Description:	The audit has found a LAPB or LAPD LLCB either on timing or timeout list whose <i>tix</i> field is out of range.
Possible Error Effect:	Prints out a report and recovers the LLCB.
Corrective Action Taken:	The audit will report the error and return AUFALSE which will recover the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB's starting address.
<i>Bad Data</i> =	LLCB's <i>tix</i> field.
<i>Good Data</i> =	Any value between 0 and 7 (inclusive).
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 60 Error Code: TYPE

Error Code:	TYPE
Error Description:	"Invalid PH channel index". The audit found a LAPB LLCB whose "phctype" does not match the write protected D-channel data block's (DCDB) "phctype" field.
Possible Error Effect:	Prints an error report and corrects the LLCB's "phctype".
Corrective Action Taken:	The audit reports the error and sets the LLCB's "phctype" to the DCDB's "phctype" field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's phctype field.
<i>Good Data</i> =	DCDB's phctype field (write protected data).
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	NA.

Table 61 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	This is an intra-ph LLCB but the type field was found not to be PSL_INTRAPH, which is an error.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit reports the error sets the LLCB back to its default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's type field.
<i>Good Data</i> =	PSL_INTRAPH - valid value for type field for the intra-PH LLCBs.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 62 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	For an RRL LLCB, the type field was found not to be PSL_RRL, which is an error.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit reports the error sets the LLCB back to its default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB's type field.
<i>Good Data</i> =	PSL_RRL - valid value for type field for the RRL LLCBs.
<i>Logical Index</i> =	LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 63 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The audit found a LAPB/LAPD LLCB on the free linked list with either the first

	IFD, last IFD, ready IFD or blocked IFD not null.
Possible Error Effect:	Prints an error report and recovers the LLCB.
Corrective Action Taken:	The audit reports the error and returns AUFALSE which recovers the LAPB/LAPD to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB's first IFD.
<i>Bad Data =</i>	LLCB's last IFD.
<i>Good Data =</i>	PSIFDNULL - null IFD field value.
<i>Logical Index =</i>	LLCB index.
<i>Dump =</i>	(All sized as unsigned short.)
	• LLCB's IFD ready field. • LLCB's IFD blocked field.

LLCB-B

Software Release: 5E12 only

1. LOGICAL LINK CONTROL BLOCK (LLCB) AUDIT

The LLCB audit checks logical link control blocks.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description."

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description."

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ACT_LNK

Error Code:	ACT_LNK
Error Description:	A non-idle LLCB (LLCB.type != PSL_NULL) has non-null LLCB to LLCB linkage fields (LLCB.forw llcb != PSLLCBNULL or LLCB>back llcb != PSLLCBNULL).
Possible Error Effect:	The LLCB may be linked in the idle list in which case it could be reallocated.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB.type.
<i>Bad Data =</i>	LLCB.forw llcb.
<i>Good Data =</i>	LLCB.back llcb.
<i>Logical Index =</i>	The LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 2 Error Code: ALCBIND

Error Code:	ALCBIND
Error Description:	An LLCB.alcb_index field is out of range (!= PSALCBNULL and ≥ PSHMAXALCB).
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	The audit idles the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB.type.
<i>Bad Data =</i>	LLCB.alcb_index.
<i>Good Data =</i>	PSHMAXALCB.
<i>Logical Index =</i>	The LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 3 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The LLCB contains the index of an IFD which is not owned by the LLCB (IFD.client != PSICLLCB or IFD.clientidx != the LLCB index).

Possible Error Effect:	Frames may be dropped or their contents corrupted.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump 1</i> =	IFD linkage information for this list.
<i>Dump 2</i> =	The LLCB.

Table 4 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit was unable to retrieve IFD linkage information for the IFD list being audited.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The top 16 bits contain the number of IFD lists owned by this structure. The lower 16 bits contain an index indicating which list is currently being audited.
<i>Bad Data</i> =	A value used internally for debugging.
<i>Good Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump</i> =	The LLCB.

Table 5 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The global counter OCbusy_cnt did not equal the audit's count of non-null entries in OCstr_time[].
Possible Error Effect:	The PH can potentially be in overload forever.
Corrective Action Taken:	The audit sets OCbusy_cnt equal to the value calculated by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of OCbusy_cnt.
<i>Good Data</i> =	The value calculated by the audit.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 6 Error Code: BUSYCHAN

Error Code:	BUSYCHAN
Error Description:	The audit found an in-use OCstr_time[llcb_index] entry where the LLCB.own_busy field was DBNO.
Possible Error Effect:	The PH may invalidly go into overload since each non-null entry in OCstr_time[] contributes to an overload control counter.
Corrective Action Taken:	The audit sets the OCstr_time[] array entry to OCTNULL and idles the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.own_busy.
<i>Good Data</i> =	OCstr_time[LLCB index].
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 7 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	An LLCB.dchan field is out of range ($\geq$ PSMAXCHN) for a non-idle LLCB.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is

	corrupt.
Corrective Action Taken:	The audit idles the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.dchan.
<i>Good Data</i> =	PSMAXCHN.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 8 Error Code: HEAD

Error Code:	HEAD
Error Description:	An IFD list owned by this LLCB is invalid. Either the <i>ready</i> or <i>blocked</i> indexes are not null (PSIFDNULL) and not valid ($\geq$ PSRIFDS).
Possible Error Effect:	IFDs may be owned when they should not be, possibly contributing to resource overloads which may in turn cause frames to be dropped.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump 1</i> =	IFD linkage information for this list.
<i>Dump 2</i> =	The LLCB.

Table 9 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	An idle LLCB (LLCB.type = PSL_NULL) is not linked to the LLCB idle list or appears to own other resources.
Possible Error Effect:	The LLCB and/or the resources it owns are not available for use. If enough resources are lost, call setups may be denied or packets dropped or the PH could go into overload.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LLCB idle list head (PSrlcbhd).
<i>Bad Data</i> =	A boolean flag indicating if there were linkage problems.
<i>Good Data</i> =	OCstr_t_time[LLCB index].
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The LLCB indexed by LLCB.fow_ llcb (if it is in range).
<i>Dump 3</i> =	The LLCB indexed by LLCB.back_ llcb (if it is in range).

Table 10 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	An OCstr_t_time[LLCB index] entry was out of range ($\neq$ OCTNULL and $\geq$ OCTMAX).
Possible Error Effect:	PH overloads may persist longer than they should.
Corrective Action Taken:	The audit sets the OCstr_t_time[] entry to OCTNULL and recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	The OCstr_t_time[] entry.
<i>Good Data</i> =	LLCB.own_busy.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 11 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The LLCB free list headcell contains invalid or inconsistent data. One of the following is true: • PSrllcbhd.first is out of range ($\geq$ PSMAXLLCB) and not null ($\neq$ PSLLCBNULL). • PSrllcbhd.last is out of range ($\geq$ PSMAXLLCB) and not null ($\neq$ PSLLCBNULL). • PSrllcbhd.first is in range ($<$ PSMAXLLCB) but PSrllcbhd.last is not, or vice versa.
Possible Error Effect:	LLCBs can not be allocated.
Corrective Action Taken:	The audit rebuilds the LLCB idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSrllcbhd.first.
<i>Bad Data</i> =	PSrllcbhd.last.
<i>Good Data</i> =	PSMAXLLCB
<i>Logical Index</i> =	NA.
<i>Dump</i> =	None.

Table 12 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	A flow control LLCB is linked to a DCCB using the DCCB.teib[].sapi2llcb[] fields or a non-flow control LLCB is linked to the FCCB.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB and the DCCB or FCCB are both corrupt.
Corrective Action Taken:	The audit recovers the LLCB to the idle list. Related audits will recover the related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.dchan (flow control) or the FCCB's LLCB index (non-flow control).
<i>Good Data</i> =	LLCB.tix (flow control) or LLCB.dchan (non-flow control).
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 13 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	Both indexes in the LLCB idle list head (PSrllcbhd.first and PSrllcbhd.last) are in range ($<$ PSMAXLLCB) but at least one of the LLCBs is not idle (LLCB.type $\neq$ PSL_NULL) or not at the end of the list (first LLCB.back_llcb $\neq$ PSLLCBNULL or last LLCB.forw_llcb $\neq$ PSLLCBNULL).
Possible Error Effect:	LLCBs can not be allocated.
Corrective Action Taken:	The audit rebuilds the LLCB idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSrllcbhd.first.
<i>Bad Data</i> =	PSrllcbhd.last.
<i>Good Data</i> =	PSLLCBNULL.
<i>Logical Index</i> =	PSL_NULL.
<i>Dump 1</i> =	The first LLCB.
<i>Dump 2</i> =	The last LLCB.

Table 14 Error Code: NONNULL

Error Code:	NONNULL

Error Description:	An IFD list owned by this LLCB is invalid. Although the "first" and "last" IFD indexes are null, either the <i>ready</i> or <i>blocked</i> indexes are not null (PSIFDNULL).
Possible Error Effect:	IFDs may be owned when they should not be, possibly contributing to resource overloads which may in turn cause frames to be dropped.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump 1</i> =	IFD linkage information for this list.
<i>Dump 2</i> =	The LLCB.

Table 15 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	A flow control LLCB may own an ALCB (LLCB.alcb_index != PSALCBNULL).
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt. It may also be holding resources which should be available for other kinds of links.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.alcb_index.
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 16 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A flow control LLCB is not correctly linked to the FCCB or a non-flow control LLCB is not correctly linked to the DCCB.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type (flow control) or LLCB.dchan (non-flow control).
<i>Bad Data</i> =	LLCB.dchan (flow control) or LLCB.tix (non-flow control).
<i>Good Data</i> =	FCCB's LLCB index (flow control) or LLCB.sapi_i (non-flow control).
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 17 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	The LLCB.phctype field is not equal to the associated D channel's PH channel type (LLCB.phctype != PSdcbd[LLCB.dchan].phctype).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit sets LLCB.phctype = DCDB.phctype.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.phctype.
<i>Good Data</i> =	DCDB.phctype.
<i>Logical Index</i> =	The LLCB index.

<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 18 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	An LLCB contains the index of an ALCB (LLCB.alcb) but the ALCB.llcb_index != the LLCB's index.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.alcb_index.
<i>Good Data</i> =	ALCB.llcb_index.
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The ALCB.

Table 19 Error Code: QUAR

Error Code:	QUAR
Error Description:	An LLCB.quar_bit field was set (= DBYES).
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt. It was first detected outside of the audit at which time the quarantine bit was set.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.dchan.
<i>Good Data</i> =	LLCB.alcb_index.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 20 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve IFD list information for the LLCB.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The number of IFD lists owned by the LLCB.
<i>Bad Data</i> =	A value used internally for debugging.
<i>Good Data</i> =	The maximum number of IFD lists which can be owned.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump</i> =	The LLCB.

Table 21 Error Code: STATE

Error Code:	STATE
Error Description:	There is an internal inconsistency between LLCB.state and the timers associated with the LLCB or the LLCB state is invalid.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	Depending on the specific error, the audit either recovers the LLCB to a valid state or recovers it to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	An internal value indicating the error type.
<i>Good Data</i> =	LLCB.dchan.

<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 22 Error Code: STUCK

Error Code:	STUCK
Error Description:	An LLCB has been holding IFDs for a longer time than is allowed based on LLCB.frmag_cnt.
Possible Error Effect:	The IFDs are not available for other users. If enough IFDs were lost, packets could be dropped and the PH could go into overload.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.frmag_cnt.
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 23 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	A timer index in an LLCB is out of range or indexes a timer which is not owned by the LLCB or is of an unexpected type.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit sets the timer index field in the LLCB to the null value (PSNULLTE) or takes other special recovery based on the timer type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An internal value used to indicate which timer index in the LLCB was being checked.
<i>Bad Data</i> =	An internal value used to indicate the type of error.
<i>Good Data</i> =	An internal value indicating which entry in an array of timers (as indicated by the <i>Error Address</i> value) was being checked.
<i>Logical Index</i> =	The timer element index.
<i>Dump</i> =	The LLCB.

Table 24 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	An LLCB which should not be throttled by overload control has a non-null (!= OCTNULL) value in the associated OCstr_t_time[LLCB index] entry.
Possible Error Effect:	The PH could erroneously go into overload.
Corrective Action Taken:	The audit sets OCstr_t_time[LLCB index] = OCTNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.phctype.
<i>Bad Data</i> =	OCstr_t_time[LLCB index].
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 25 Error Code: TYPE

Error Code:	TYPE
Error Description:	The LLCB.type field is out of range ($\geq$ PSL_MAX).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB.type.
<i>Good Data</i> =	PSL_MAX.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 26 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	If the index of the last LLCB being audited is valid, the audit recovers that LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSMAXLLCB.
<i>Logical Index</i> =	Index of the last LLCB audited.
<i>Dump</i> =	The last LLCB audited (if the index is valid).

LLCB-C

Software Release: 5E13 and later

1. LOGICAL LINK CONTROL BLOCK (LLCB) AUDIT

The LLCB audit checks logical link control blocks.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description."

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description."

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ACT_LNK

Error Code:	ACT_LNK
Error Description:	A non-idle LLCB (LLCB.type != PSL_NULL) has non-null LLCB to LLCB linkage fields (LLCB.forw llcb != PSLLCBNULL or LLCB>back llcb != PSLLCBNULL).
Possible Error Effect:	The LLCB may be linked in the idle list in which case it could be reallocated.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB.type.
<i>Bad Data =</i>	LLCB.forw llcb.
<i>Good Data =</i>	LLCB.back llcb.
<i>Logical Index =</i>	The LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 2 Error Code: ALCBIND

Error Code:	ALCBIND
Error Description:	An LLCB.alcb_index field is out of range (!= PSALCBNULL and ≥ PSHMAXALCB).
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	The audit idles the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB.type.
<i>Bad Data =</i>	LLCB.alcb_index.
<i>Good Data =</i>	PSHMAXALCB.
<i>Logical Index =</i>	The LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 3 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The LLCB contains the index of an IFD which is not owned by the LLCB (IFD.client != PSICLLCB or IFD.clientidx != the LLCB index).

Possible Error Effect:	Frames may be dropped or their contents corrupted.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump 1</i> =	IFD linkage information for this list.
<i>Dump 2</i> =	The LLCB.

Table 4 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit was unable to retrieve IFD linkage information for the IFD list being audited.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The top 16 bits contain the number of IFD lists owned by this structure. The lower 16 bits contain an index indicating which list is currently being audited.
<i>Bad Data</i> =	A value used internally for debugging.
<i>Good Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump</i> =	The LLCB.

Table 5 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The global counter OCbusy_cnt did not equal the audit's count of non-null entries in OCstr_t_time[].
Possible Error Effect:	The PH can potentially be in overload forever.
Corrective Action Taken:	The audit sets OCbusy_cnt equal to the value calculated by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The value of OCbusy_cnt.
<i>Good Data</i> =	The value calculated by the audit.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 6 Error Code: BUSYCHAN

Error Code:	BUSYCHAN
Error Description:	The audit found an in-use OCstr_t_time[llcb_index] entry where the LLCB.own_busy field was DBNO.
Possible Error Effect:	The PH may invalidly go into overload since each non-null entry in OCstr_t_time[] contributes to an overload control counter.
Corrective Action Taken:	The audit sets the OCstr_t_time[] array entry to OCTNULL and idles the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.own_busy.
<i>Good Data</i> =	OCstr_t_time[LLCB index].
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 7 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	An LLCB.dchan field is out of range ($\geq$ PSMAXCHN) for a non-idle LLCB.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is

	corrupt.
Corrective Action Taken:	The audit idles the LLCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB.type.
<i>Bad Data =</i>	LLCB.dchan.
<i>Good Data =</i>	PSMAXCHN.
<i>Logical Index =</i>	The LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 8 Error Code: HEAD

Error Code:	HEAD
Error Description:	An IFD list owned by this LLCB is invalid. Either the <i>ready</i> or <i>blocked</i> indexes are not null (PSIFDNULL) and not valid ($\geq$ PSRIFDS).
Possible Error Effect:	IFDs may be owned when they should not be, possibly contributing to resource overloads which may in turn cause frames to be dropped.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data =</i>	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Good Data =</i>	The number of IFD lists owned by the structure under audit.
<i>Logical Index =</i>	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump 1 =</i>	IFD linkage information for this list.
<i>Dump 2 =</i>	The LLCB.

Table 9 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	An idle LLCB (LLCB.type = PSL_NULL) is not linked to the LLCB idle list or appears to own other resources.
Possible Error Effect:	The LLCB and/or the resources it owns are not available for use. If enough resources are lost, call setups may be denied or packets dropped or the PH could go into overload.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The LLCB idle list head (PSRlcbhd).
<i>Bad Data =</i>	A boolean flag indicating if there were linkage problems.
<i>Good Data =</i>	OCstr_t_time[LLCB index].
<i>Logical Index =</i>	The LLCB index.
<i>Dump 1 =</i>	The LLCB being audited.
<i>Dump 2 =</i>	The LLCB indexed by LLCB.fow_ llcb (if it is in range).
<i>Dump 3 =</i>	The LLCB indexed by LLCB.back_ llcb (if it is in range).

Table 10 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	An OCstr_t_time[LLCB index] entry was out of range ($\neq$ OCTNULL and $\geq$ OCTMAX).
Possible Error Effect:	PH overloads may persist longer than they should.
Corrective Action Taken:	The audit sets the OCstr_t_time[] entry to OCTNULL and recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LLCB.type.
<i>Bad Data =</i>	The OCstr_t_time[] entry.
<i>Good Data =</i>	LLCB.own_busy.
<i>Logical Index =</i>	The LLCB index.
<i>Dump =</i>	The LLCB being audited.

Table 11 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The LLCB free list headcell contains invalid or inconsistent data. One of the following is true: • PSrllcbhd.first is out of range ($\geq$ PSMAXLLCB) and not null ($\neq$ PSLLCBNULL). • PSrllcbhd.last is out of range ($\geq$ PSMAXLLCB) and not null ($\neq$ PSLLCBNULL). • PSrllcbhd.first is in range ($<$ PSMAXLLCB) but PSrllcbhd.last is not, or vice versa.
Possible Error Effect:	LLCBs can not be allocated.
Corrective Action Taken:	The audit rebuilds the LLCB idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSrllcbhd.first.
<i>Bad Data</i> =	PSrllcbhd.last.
<i>Good Data</i> =	PSMAXLLCB
<i>Logical Index</i> =	NA.
<i>Dump</i> =	None.

Table 12 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	A flow control LLCB is linked to a DCCB using the DCCB.teib[].sapi2llcb[] fields or a non-flow control LLCB is linked to the FCCB.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB and the DCCB or FCCB are both corrupt.
Corrective Action Taken:	The audit recovers the LLCB to the idle list. Related audits will recover the related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.dchan (flow control) or the FCCB's LLCB index (non-flow control).
<i>Good Data</i> =	LLCB.tix (flow control) or LLCB.dchan (non-flow control).
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 13 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	Both indexes in the LLCB idle list head (PSrllcbhd.first and PSrllcbhd.last) are in range ($<$ PSMAXLLCB) but at least one of the LLCBs is not idle (LLCB.type $\neq$ PSL_NULL) or not at the end of the list (first LLCB.back_llcb $\neq$ PSLLCBNULL or last LLCB.forw_llcb $\neq$ PSLLCBNULL).
Possible Error Effect:	LLCBs can not be allocated.
Corrective Action Taken:	The audit rebuilds the LLCB idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PSrllcbhd.first.
<i>Bad Data</i> =	PSrllcbhd.last.
<i>Good Data</i> =	PSLLCBNULL.
<i>Logical Index</i> =	PSL_NULL.
<i>Dump 1</i> =	The first LLCB.
<i>Dump 2</i> =	The last LLCB.

Table 14 Error Code: NONNULL

Error Code:	NONNULL

Error Description:	An IFD list owned by this LLCB is invalid. Although the "first" and "last" IFD indexes are null, either the <i>ready</i> or <i>blocked</i> indexes are not null (PSIFDNULL).
Possible Error Effect:	IFDs may be owned when they should not be, possibly contributing to resource overloads which may in turn cause frames to be dropped.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump 1</i> =	IFD linkage information for this list.
<i>Dump 2</i> =	The LLCB.

Table 15 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	A flow control LLCB may own an ALCB (LLCB.alcb_index != PSALCBNULL).
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt. It may also be holding resources which should be available for other kinds of links.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.alcb_index.
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 16 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A flow control LLCB is not correctly linked to the FCCB or a non-flow control LLCB is not correctly linked to the DCCB.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type (flow control) or LLCB.dchan (non-flow control).
<i>Bad Data</i> =	LLCB.dchan (flow control) or LLCB.tix (non-flow control).
<i>Good Data</i> =	FCCB's LLCB index (flow control) or LLCB.sapi_i (non-flow control).
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 17 Error Code: PHCTYPE

Error Code:	PHCTYPE
Error Description:	The LLCB.phctype field is not equal to the associated D channel's PH channel type (LLCB.phctype != PSdcbd[LLCB.dchan].phctype).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit sets LLCB.phctype = DCDB.phctype.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.phctype.
<i>Good Data</i> =	DCDB.phctype.
<i>Logical Index</i> =	The LLCB index.

<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The DCDB indexed by LLCB.dchan.
<i>Dump 3</i> =	The DCCB indexed by LLCB.dchan.

Table 18 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	An LLCB contains the index of an ALCB (LLCB.alcb) but the ALCB.llcb_index != the LLCB's index or the LLCB should not own any ALCBs based on LLCB.phctype.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.alcb_index.
<i>Good Data</i> =	ALCB.llcb_index.
<i>Logical Index</i> =	The LLCB index.
<i>Dump 1</i> =	The LLCB being audited.
<i>Dump 2</i> =	The ALCB (PSalcb[LLCB.alcb_index]).
<i>Dump 3</i> =	The DCDB (PSdccb[LLCB.dchan]).
<i>Dump 4</i> =	The DCCB (PSdccb[LLCB.dchan]).

Table 19 Error Code: QUAR

Error Code:	QUAR
Error Description:	An LLCB.quar_bit field was set (= DBYES).
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt. It was first detected outside of the audit at which time the quarantine bit was set.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.dchan.
<i>Good Data</i> =	LLCB.alcb_index.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 20 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve IFD list information for the LLCB.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The number of IFD lists owned by the LLCB..
<i>Bad Data</i> =	A value used internally for debugging.
<i>Good Data</i> =	The maximum number of IFD lists which can be owned.
<i>Logical Index</i> =	The top 16 bits contain PSICLLCB. The bottom 16 bits contain the LLCB index.
<i>Dump</i> =	The LLCB.

Table 21 Error Code: STATE

Error Code:	STATE
Error Description:	There is an internal inconsistency between LLCB.state and the timers associated with the LLCB or the LLCB state is invalid.
Possible Error Effect:	Varies according to conditions. The protocol stack associated with this LLCB is corrupt.
Corrective Action Taken:	Depending on the specific error, the audit either recovers the LLCB to a valid state or recovers it to the idle list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	An internal value indicating the error type.
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 22 Error Code: STUCK

Error Code:	STUCK
Error Description:	An LLCB has been holding IFDs for a longer time than is allowed based on LLCB.frmag_cnt.
Possible Error Effect:	The IFDs are not available for other users. If enough IFDs were lost, packets could be dropped and the PH could go into overload.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.type.
<i>Bad Data</i> =	LLCB.frmag_cnt.
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 23 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	A timer index in an LLCB is out of range or indexes a timer which is not owned by the LLCB or is of an unexpected type.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit sets the timer index field in the LLCB to the null value (PSNULLTE) or takes other special recovery based on the timer type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An internal value used to indicate which timer index in the LLCB was being checked.
<i>Bad Data</i> =	An internal value used to indicate the type of error.
<i>Good Data</i> =	An internal value indicating which entry in an array of timers (as indicated by the <i>Error Address</i> value) was being checked.
<i>Logical Index</i> =	The timer element index.
<i>Dump</i> =	The LLCB.

Table 24 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	An LLCB which should not be throttled by overload control has a non-null (!= OCTNULL) value in the associated OCstr_t_time[LLCB index] entry.
Possible Error Effect:	The PH could erroneously go into overload.
Corrective Action Taken:	The audit sets OCstr_t_time[LLCB index] = OCTNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LLCB.phctype.
<i>Bad Data</i> =	OCstr_t_time[LLCB index].
<i>Good Data</i> =	LLCB.dchan.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 25 Error Code: TYPE

Error Code:	TYPE
Error Description:	The LLCB.type field is out of range ($\geq$ PSL_MAX).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit recovers the LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LLCB.type.

<i>Good Data</i> =	PSL_MAX.
<i>Logical Index</i> =	The LLCB index.
<i>Dump</i> =	The LLCB being audited.

Table 26 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	If the index of the last LLCB being audited is valid, the audit recovers that LLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSMAXLLCB.
<i>Logical Index</i> =	Index of the last LLCB audited.
<i>Dump</i> =	The last LLCB audited (if the index is valid).

LLMIDCU**Software Release:** 5E10 - 5E11**1. LOGICAL LINK MAP INTEGRATED DIGITAL SERVICE UNIT (IDCU) - (LLMIDCU) AUDIT**

The LLMIDCU audit will performs the dynamic to static and the semantic checks of the Logical Link Map data structure rLLM_IDCU.

2. DATA STRUCTURES

The data structure being audited by LLMIDCU audit is the rLLM_IDCU. This structure holds the information about the status of the IDCU Embedded Operational Channel (EOC) and the Terminal Maintenance Channel (TMC). The structure fields and a brief description of them is as follows:

e_sw	= Indicates if EOC switch is in progress.
e_inhsw	= Indicates if EOC is allowed to switch.
e_p_inh	= Indicates if primary EOC is in service.
e_p_link	= Internal circuit name of a primary EOC.
e_p_ph	= PH information for primary EOC.
e_p_chan	= PH channel information for primary EOC.
e_m_inh	= Indicates if mate EOC is in service.
e_m_link	= Internal circuit name of mate EOC.
e_m_ph	= Contains PH information for mate EOC.
e_m_chan	= Contains PH channel information for mate EOC.
t_sw	= Indicates if TMC switch is in progress.
t_inhsw	= Indicates if TMC is allowed to switch.
t_p_inh	= Indicates if primary TMC is in service.
t_p_link	= Internal circuit name of primary TMC.
t_p_ph	= PH information for primary TMC.
t_p_chan	= PH channel information for primary TMC.
t_m_inh	= Indicates if mate TMC is in service.
t_m_link	= Internal circuit name of mate TMC.
t_m_ph	= PH information for mate TMC.
t_m_chan	= PH channel information for mate TMC.
e_rcvb	= Indicates if EOC messages are received on both primary and mate EOC.
t_rcvb	= Indicates if TMC messages are received on both primary and mate TMC.

e_rcvb_tmr = EOC "receive both" timer.

t_rcvb_tmr = TMC "receive both" timer.

e_v120rexmt = Indicates that an EOC path switch has occurred.

3. AUDIT CHECKS

The LLMIDCU audit will perform the dynamic to static and the semantic checks of the rLLM_IDCU tuples.

EXTRATUP = Dynamic to static check of the rLLM_IDCU tuples.

LLM_CKT = Audit the EOC/TMC circuit name in the rLLM_IDCU tuple.

SMEST = Verify that the rSMEST tuples exist for an EOC/TMC circuit.

LLM_SW = Audit the rLLM_IDCU.llm_sw field.

LLM_RCVB = Audit the rLLM_IDCU.e_rcvb and rLLM_IDCU.t_rcvb fields.

LLM_STAT = Audit the rLLM_IDCU.t_p_inh, rLLM_IDCU.t_m_inh, rLLM_IDCU.e_p_inh and rLLM_IDCU.e_m_inh fields.

ODDLOSTTUP = Verify that the rIDSLEQUIP tuple exists for the EOC/TMC circuits.

LLM_DSLEQ = Verify that the PH information is consistent between the rLLM_IDCU and the rIDSLEQUIP tuples.

4. ERROR HANDLING

The general audit error handling mechanism is to print the error code and load the circuit EOC/TMC on the MRA recovery queue.

The audit produces the following error code(s).

Table 1 Error Code: LLM_CKT

Error Code:	LLM_CKT
Error Description:	The EOC or TMC circuit name in the rLLM_IDCU tuple could not be calculated.
Possible Error Effect:	The EOC/TMC at fault will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the calculated circuit on the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLLM_IDCU tuple at fault.
<i>Bad Data</i> =	The EOC/TMC circuit name as it appears in the rLLM_IDCU tuple.
<i>Good Data</i> =	The loop control 0-3, it tells which of the EOC/TMC was at fault as follows.
	0 = Implies that the primary EOC circuit in the rLLM_IDCU tuple is bad.
	1 = Implies that the primary TMC circuit in the rLLM_IDCU tuple is bad.
	2 = Implies that the mate EOC circuit in the rLLM_IDCU tuple is bad.
	3 = Implies that the mate TMC circuit in the rLLM_IDCU tuple is bad.
<i>Logical Index</i> =	The rLLM_IDCU.idt_rdt key field.
<i>Dump</i> =	The contents of the rLLM_IDCU tuple.

Table 2 Error Code: LLM_DSLEQ

Error Code:	LLM_DSLEQ
Error Description:	The rIDSLEQUIP.dsl_group field does not match the PH information in the rLLM_IDCU tuple.
Possible Error Effect:	The EOC/TMC will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the circuit on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_IDCU tuple.
<i>Bad Data =</i>	The rIDSLEQUIP.dsl_group field.
<i>Good Data =</i>	The rIDSLEQUIP.member field.
<i>Logical Index =</i>	The contents of a local loop variable. The value of 0 means it is the primary EOC circuit. Valid values are: 1 = It is the primary TMC circuit. 2 = It is the mate EOC circuit. 3 = It is the mate TMC circuit.
<i>Dump1 =</i>	The contents of the rLLM_IDCU tuple.
<i>Dump2 =</i>	The contents of the rIDSLEQUIP tuple.

Table 3 Error Code: LLM_RCVB

Error Code:	LLM_RCVB
Error Description:	The rLLM_IDCU.e_rcvb or rLLM_IDCU.t_rcvb fields state that there is no switch in progress but the rSMEST tuple associated with this circuit is validly busy.
Possible Error Effect:	The switch of the EOC/TMC will not be possible since it is stuck in this state.
Corrective Action Taken:	The audit will load the circuit at fault on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_IDCU tuple.
<i>Bad Data =</i>	The contents of the "llm_rcvb" field.
<i>Good Data =</i>	The rSMEST.stat_act field. The value of 0 means busy and 1 means idle.
<i>Logical Index =</i>	The contents of a local loop variable. The value of 0 means it is the EOC switch that was stuck, 1 means it was the TMC switch that was stuck.
<i>Dump1 =</i>	The contents of the rLLM_IDCU tuple.
<i>Dump2 =</i>	The contents of the rSMEST tuple.

Table 4 Error Code: LLM_STAT

Error Code:	LLM_STAT
Error Description:	The IS/OOS status of the EOC/TMC does not match the IS/OOS status of the rSMEST tuple.
Possible Error Effect:	The EOC/TMC will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the circuit on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_IDCU tuple.
<i>Bad Data =</i>	The "llm_stat" field from the rLLM_IDCU tuple.
<i>Good Data =</i>	The "bas_stat" field from the rSMEST tuple. Valid values are: 1 = Active. 2 = Stand_by. 4 = OOS.
<i>Logical Index =</i>	The contents of a local loop variable. Valid values are: 0 = It is the primary EOC circuit.

	1 = It is the primary TMC circuit. 2 = It is the mate EOC circuit. 3 = It is the mate TMC circuit.
<i>Dump1</i> =	The contents of the rLLM_IDCU tuple.
<i>Dump2</i> =	The contents of the rSMEST tuple.

Table 5 Error Code: LLM_SW

Error Code:	LLM_SW
Error Description:	The "llm_sw" but states that there is a switch in progress but the rSMEST tuple associated with this circuit is not busy.
Possible Error Effect:	The switch of the EOC/TMC will not be possible since it is stuck in this state.
Corrective Action Taken:	The audit will reset the bits at fault and the circuit will be loaded on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLLM_IDCU tuple.
<i>Bad Data</i> =	The contents of the "llm_sw" field.
<i>Good Data</i> =	The "stat_act" field from the rSMEST tuple. Valid values are: 0 = Busy. 1 = Idle.
<i>Logical Index</i> =	The contents of a local loop variable. Valid values are: 0 = It is the EOC switch that was stuck. 1 = It was the TMC switch that was stuck.
<i>Dump1</i> =	The contents of the rLLM_IDCU tuple.
<i>Dump2</i> =	The contents of the rSMEST tuple.

Table 6 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit could not find an rDSLEQUIP tuple in the database with the EOC/TMC circuit name.
Possible Error Effect:	The EOC/TMC will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the circuit on the MRA queue.
Manual Action Required:	The user should investigate the missing static rDSLEQUIP tuple in the database.
Dump Description:	
<i>Error Address</i> =	The address of the rLLM_IDCU tuple.
<i>Bad Data</i> =	The return from the database read function. 0 means a successful read, any other value implies a failure in the database read.
<i>Good Data</i> =	The EOC/TMC circuit name. This is the key used in reading the missing rDSLEQUIP tuple.
<i>Logical Index</i> =	The contents of a local loop variable. Valid values are: 0 = It is the primary EOC circuit. 1 = It is the primary TMC circuit. 2 = It is the mate EOC circuit. 3 = It is the mate TMC circuit.
<i>Dump1</i> =	The contents of the rLLM_IDCU tuple.

<i>Dump2</i> =	The contents of the rSMEST tuple.

Table 7 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for a given circuit is missing from the database.
Possible Error Effect:	The maintenance of the circuit will not be possible.
Corrective Action Taken:	The audit will schedule the CKTDATA audit which will delete the rSMEST tuple from the database. No corrective action will be taken by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLLM_IDCU tuple at fault.
<i>Bad Data</i> =	The circuit name of the missing rSMEST tuple.
<i>Good Data</i> =	For which EOC/TMC circuit's rSMEST tuple is missing. Valid values are: 0 = Primary EOC's. 1 = Primary TMC's. 2 = Mate EOC. 3 = Mate TMC.
<i>Logical Index</i> =	The rLLM_IDCU.idt_rdt key field.
<i>Dump</i> =	The contents of the rLLM_IDCU tuple.

LLMRDT

Software Release: 5E12 and later

1. LOGICAL LINK MAP REMOTE DIGITAL TERMINAL (LLMRDT) AUDIT

The TR-303 interface on a digital networking unit-synchronous optical network (SONET) (DNU-S) feature provides the capability to assign a TR-303 remote digital terminal (RDT) to digital signal level 1 (DS1) facilities terminating on a DNU-S. The data for the active EOC and TMC data link names for a given TR-303 RDT is stored in the RLLLM_RDT relation. Data in this relation is audited by the LLMRDT audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLLLM_RDT relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The general audit error handling mechanism is to print the error code and load the circuit EOC/TMC on the MRA recovery queue.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rISMEST tuple for a given circuit is missing from the database.
Possible Error Effect:	The maintenance of the circuit will not be possible.
Corrective Action Taken:	The audit will schedule the CKTDATA audit which will delete the rISMEST tuple from the database. No corrective action will be taken by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLLM_RDT tuple at fault.
<i>Bad Data =</i>	The circuit name of the missing rISMEST tuple.
<i>Good Data =</i>	The data for which EOC/TMC circuit's rISMEST tuple is missing Valid values are: 0 = Primary EOC's. 1 = Primary TMC's. 2 = Mate EOC. 3 = Mate TMC.
<i>Logical Index =</i>	The rLLLM_RDT.rdt_unit key field.
<i>Dump =</i>	The contents of the rLLLM_RDT tuple.

Table 2 Error Code: LLM_CKT

Error Code:	LLM_CKT
Error Description:	The EOC or TMC circuit name in the rLLLM_RDT tuple could not be calculated.

Possible Error Effect:	The EOC/TMC at fault will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the calculated circuit on the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_RDT tuple at fault.
<i>Bad Data =</i>	The EOC/TMC circuit name as it appears in the rLLM_RDT tuple.
<i>Good Data =</i>	The loop control 0-3, it tells which of the EOC/TMC was at fault as follows: 0 = Implies that the primary EOC circuit in the rLLM_RDT tuple is bad. 1 = Implies that the primary TMC circuit in the rLLM_RDT tuple is bad. 2 = Implies that the mate EOC circuit in the rLLM_RDT tuple is bad. 3 = Implies that the mate TMC circuit in the rLLM_RDT tuple is bad.
<i>Logical Index =</i>	The rLLM_RDT.rdt_unit key field.
<i>Dump =</i>	The contents of the rLLM_RDT tuple.

Table 3 Error Code: LLM_DSLEQ

Error Code:	LLM_DSLEQ
Error Description:	The rIDSLEQUIP.dsl_group field does not match the rLLM_RDT.ph field or the rIDSLEQUIP.member field does not match the rLLM_RDT.chan field.
Possible Error Effect:	The EOC/TMC will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the circuit on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_RDT tuple.
<i>Bad Data =</i>	The rIDSLEQUIP.dsl_group field.
<i>Good Data =</i>	The rIDSLEQUIP.member field.
<i>Logical Index =</i>	The contents of a local loop variable. Valid values are: 0 = The primary EOC circuit. 1 = The primary TMC circuit. 2 = The mate EOC circuit. 3 = The mate TMC circuit.
<i>Dump 1 =</i>	The contents of the rLLM_RDT tuple.
<i>Dump 2 =</i>	The contents of the rIDSLEQUIP tuple.

Table 4 Error Code: LLM_RCVB

Error Code:	LLM_RCVB
Error Description:	The rLLM_RDT.e_rcvb or rLLM_RDT.t_rcvb fields state that there is no switch in progress but the rSMEST tuple associated with this circuit is validly busy.
Possible Error Effect:	The switch of the EOC/TMC will not be possible since it is stuck in this state.
Corrective Action Taken:	The audit will load the circuit at fault on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_RDT tuple.
<i>Bad Data =</i>	The contents of the "llm_rcvb" field.
<i>Good Data =</i>	The rSMEST.stat_act field. Valid values are: 0 = Busy. 1 = Idle.
<i>Logical Index =</i>	The contents of a local loop variable. Valid values are: 0 = The EOC switch was stuck.

	1 = The TMC switch was stuck.
<i>Dump 1 =</i>	The contents of the rLLM_RDT tuple.
<i>Dump 2 =</i>	The contents of the rSMEST tuple.

Table 5 Error Code: LLM_STAT

Error Code:	LLM_STAT
Error Description:	The IS/OOS status of the EOC/TMC does not match the IS/OOS status of the rSMEST tuple.
Possible Error Effect:	The EOC/TMC will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the circuit on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_RDT tuple.
<i>Bad Data =</i>	The "llm_stat" field from the rLLM_RDT tuple.
<i>Good Data =</i>	The "bas_stat" field from the rSMEST tuple. Valid values are: 1 = Active. 2 = Stand by. 4 = OOS.
<i>Logical Index =</i>	The contents of a local loop variable. Valid values are: 0 = The primary EOC circuit. 1 = The primary TMC circuit. 2 = The mate EOC circuit. 3 = The mate TMC circuit.
<i>Dump 1 =</i>	The contents of the rLLM_RDT tuple.
<i>Dump 2 =</i>	The contents of the rSMEST tuple.

Table 6 Error Code: LLM_SW

Error Code:	LLM_SW
Error Description:	The "llm_sw" but states that there is a switch in progress but the rSMEST tuple associated with this circuit is not busy.
Possible Error Effect:	The switch of the EOC/TMC will not be possible since it is stuck in this state.
Corrective Action Taken:	The audit will reset the bits at fault and the circuit will be loaded on the MRA queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_RDT tuple.
<i>Bad Data =</i>	The contents of the "llm_sw" field.
<i>Good Data =</i>	The "stat_act" field from the rSMEST tuple. Valid values are: 0 = Busy. 1 = Idle.
<i>Logical Index =</i>	The contents of a local loop variable. Valid values are: 0 = The EOC switch was stuck. 1 = The TMC switch was stuck.
<i>Dump 1 =</i>	The contents of the rLLM_RDT tuple.

<i>Dump 2 =</i>	The contents of the rISMEST tuple.

Table 7 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit could not find an rICKTDATA tuple in the database with the TR303 RDT circuit name (rname).
Possible Error Effect:	The TR303 RDT will not be usable by call processing or maintenance.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should investigate the missing static rICKTDATA tuple in the database.
Dump Description:	
<i>Error Address =</i>	The address of the rLLM_RDT tuple.
<i>Bad Data =</i>	The return from the database read function. Valid values are: 0 = A successful read. Any other value implies a failure in the database read.
<i>Good Data =</i>	The TR303 RDT circuit name (rname). This is the key used in reading the missing rICKTDATA tuple.
<i>Logical Index =</i>	The TR303 RDT unit number (rdt_unit). This is the key used in reading the rLLM_RDT tuple.
<i>Dump 1 =</i>	The contents of the rLLM_RDT tuple.
<i>Dump 2 =</i>	The contents of the rUNIT2RDT tuple.

Table 8 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit could not find an rDSLEQUIP tuple in the database with the EOC/TMC circuit name.
Possible Error Effect:	The EOC/TMC will not be usable by call processing or maintenance.
Corrective Action Taken:	The audit will load the circuit on the MRA queue.
Manual Action Required:	Office personnel should investigate the missing static rDSLEQUIP tuple in the database.
Dump Description:	
<i>Error Address =</i>	The number 1, for error code identification.
<i>Bad Data =</i>	The return from the database read function. Valid values are: 0 = A successful read. Any other value implies a failure in the database read.
<i>Good Data =</i>	The EOC/TMC circuit name. This is the key used in reading the missing rDSLEQUIP tuple.
<i>Logical Index =</i>	The contents of a local loop variable. Valid values are: 0 = The primary EOC circuit. 1 = The primary TMC circuit. 2 = The mate EOC circuit. 3 = The mate TMC circuit.
<i>Dump 1 =</i>	The contents of the rLLM_RDT tuple.
<i>Dump 2 =</i>	The contents of the rISMEST tuple.

Table 9 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rISMEST tuple for a given circuit is missing from the database.
Possible Error Effect:	The maintenance of the circuit will not be possible.

Corrective Action Taken:	The audit will schedule the CKTDATA audit which will delete the rSMEST tuple from the database. No corrective action will be taken by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The identification number of the error code.
<i>Bad Data</i> =	The circuit name of the missing rSMEST tuple.
<i>Good Data</i> =	The data for which EOC/TMC circuit's rSMEST tuple is missing. Valid values are: 0 = Primary EOC's. 1 = Primary TMC's. 2 = Mate EOC. 3 = Mate TMC.
<i>Logical Index</i> =	The rLLM_RDT.rdt_unit key field.
<i>Dump</i> =	The contents of the rLLM_RDT tuple.

Table 10 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	Semantic data (rdt_unit, tsp_utype, tsp_unit, or rtnum fields) is corrupted in the rLLM_RDT tuple.
Possible Error Effect:	None.
Corrective Action Taken:	The audit will correct the semantic data in the rLLM_RDT tuple using the static data in the rUNIT2RDT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLLM_RDT tuple at fault.
<i>Bad Data</i> =	The circuit name of the missing rSMEST tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The rLLM_RDT.rdt_unit key field.
<i>Dump 1</i> =	The contents of the rLLM_RDT tuple.
<i>Dump 2</i> =	The contents of the rUNIT2RDT tuple.

Table 11 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The rUNIT2RDT tuple for a given circuit is missing from the database.
Possible Error Effect:	The TR303 RDT will not be usable by call processing or maintenance.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should investigate the missing static rUNIT2RDT tuple in the database.
Dump Description:	
<i>Error Address</i> =	The address of the rLLM_RDT tuple at fault.
<i>Bad Data</i> =	The address of the RLUNIT2RDT relation.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The rLLM_RDT.rdt_unit key field.
<i>Dump</i> =	The contents of the rLLM_RDT tuple.

LMDB

Software Release: 5E10 and later

1. LINK MANAGEMENT DATA BLOCK (LMDB)

Link manager data blocks (LMDBs), elements of QLlmbd[], are used to represent quad linked packet switched (QLPS) network link information. They represent the state of the link and also contain per-network QLPS endpoint information.

2. DATA STRUCTURES

The following data is used by the LMDB audit:

- (1) QLlcb[] - QLPS logical link blocks
- (2) QLlmbd[] - QLPS link manager data blocks
- (3) QLTIMER - QL timer data structure.

3. AUDIT CHECKS

The following checks are performed by the LMDB audit:

ZAP	= A reset was taken during the audit.
BADRANGE_INDEX	= Check of out-of-range lcb index in LMDB.
ONE2ONE	= Check for invalid LMDB linkage.
OUTOFRANGE	= Check for out-of-range LMDB requester.
INV_TIMER	= Check for out-of-service LMDB having a non-idle timer.
TIME_EXC	= Check for ACT or AVL LMDB having a five second timer.
GBGTIMER	= Check for invalid timer type in LMDB entry.
STATE	= Check for LMDB invalid out-of-range state.
BADIDLE	= Check softswitch flag against running of softswitch timer.
MISSING	= Endpoint existence checks.

4. ERROR HANDLING

Different recovery is taken depending upon the error reported. Refer to the "Corrective Action Taken" section of the individual error.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The softswitch flag associated with a QLPS (quad linked packet switch) network is set (QLImdb[network].soft = DBYES), but the softswitch timer associated with the network is not running (QLImdb[network].timer[QLTSSTIMER] is idle).
Possible Error Effect:	The QLPS network appears to be in softswitch when it is not and may be degraded.
Corrective Action Taken:	The audit clears the softswitch flag QLImdb[network].soft = DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Link manager's state (QLLMDB.state).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the QLLMDB (index into QLImdb[]).
<i>Dump =</i>	QLLMDB without the endpoint map (QLImdb[].llcb_ix[]).

Table 2 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	An entry in a QLPS network link manager's llcb_ix[] field is out-of-range (that is, is not QLLIXNULL and is not a valid index into QLlcb[]).
Possible Error Effect:	The endpoint associated with this entry (that is, the endpoint whose number is the index into QLLMDB.llcb_ix[]) can not be used. This may degrade the QLPS network.
Corrective Action Taken:	The audit clears the entry (QLLMDB.llcb_ix[] = QLLIXNULL). If this endpoint should be up, level 3 checks detects that it is not and causes the endpoint to be reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into QLLMDB.llcb_ix[].
<i>Bad Data =</i>	Contents of QLLMDB.llcb_ix[].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the QLLMDB (index into QLImdb[]).
<i>Dump =</i>	QLLMDB without the endpoint map (QLImdb[].llcb_ix[]).

Table 3 Error Code: GBGTIMER

Error Code:	GBGTIMER
Error Description:	An active QLPS network link manager (QLLMDB.state = QLMOOSAVL or QLMACT) has an invalid timer type in one of its QLLMDB.timer[] entries (QLTIMER.type != QLNULL and QLTIMER.type ≥ QLTNL2TYPES).
Possible Error Effect:	If the bad type is masking what should be a pending timer, the link protocol does not work correctly. Otherwise, operational code may assert on the invalid value.
Corrective Action Taken:	The audit idles the bad timer entry (sets QLTIMER.type = QLNULL).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into QLLMDB.timer[].
<i>Bad Data =</i>	Invalid timer type (QLLMDB.timer[].type).
<i>Good Data =</i>	Link manager's state (QLLMDB.state).
<i>Logical Index =</i>	Index of the QLLMDB (index into QLImdb[]).
<i>Dump =</i>	QLLMDB without the endpoint map (QLImdb[].llcb_ix[]).

Table 4 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	An out-of-service QLPS network link manager (QLLMDB.state = QLMOOSL1) has a non-idle timer. That is, QLLMDB.timer[timer_index] is not idle (QLTIMER.type != QLNULL or QLTIMER.count != QLNULL).
Possible Error Effect:	If the timer times out, it is likely to cause asserts.
Corrective Action Taken:	The audit idles the timer (sets QLTIMER.type = QLNULL and QLTIMER.count = QLNULL).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into the timer array (QLLMDB.timer[]).
<i>Bad Data =</i>	Non-idle timer's type (QLTIMER.type).

<i>Good Data</i> =	Link manager's state (QLLMDB.state).
<i>Logical Index</i> =	Index of the QLLMDB (index into QLLmdb[]).
<i>Dump</i> =	QLLMDB without the endpoint map (QLImdb[].llcb_ix[]).

Table 5 Error Code: MISSING

Error Code:	MISSING
Error Description:	On a message handler (MH) or quad gateway processor (QGP), one of the following cases is true. The value dumped in the error address field of the audit report can be used to distinguish the two cases: SMQEPOSDS or SMQEPCCS: A QLPS network endpoint should be equipped according to SMqlendpts[network][endpoint], but the endpoint is not established. That is, QLLmdb[network].llcb_ix[endpoint] contains a null (QLLIXNULL) or invalid ($\geq$ QLNLLCBS) QLCB logical link control block (QLLCB) index. SMQEPNULL: A QLPS network endpoint should not be equipped according to SMqlendpts[network][endpoint], but the endpoint appears to be established. That is, QLLmdb[network].llcb_ix[endpoint] contains a valid QLLCB index.
Possible Error Effect:	QLPS network connectivity from the processor on which the audit is running to the endpoint identified in the error report is incorrect. The QLPS network may be degraded.
Corrective Action Taken:	Recovery depends on the value dumped in the error address field of the audit report: SMQEPOSDS or SMQEPCCS: The audit requests that the endpoint be established. SMQEPNULL: The audit sets the QLLmdb[network].llcb_ix[endpoint] = QLLIXNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The endpoint type.
<i>Bad Data</i> =	QLLCB index from QLLMDB.llcb_ix[endpoint] (index into QLLmdb[]).
<i>Good Data</i> =	Network number (index into QLLmdb[]).
<i>Logical Index</i> =	Endpoint index (index into QLLMDB.llcb_ix[]).
<i>Dump</i> =	QLLMDB without the endpoint map (QLImdb[].llcb_ix[]).

Table 6 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	Either a QLPS network link manager is using endpoint 0, or has an invalid linkage to some other endpoint. That is, one of the following is true: - QLLMDB.llcb_ix[0] is a valid QLLCB (QLPS logical link control block) index. - QLLMDB.llcb_ix[endpoint] is a valid QLLCB index, but the associated QLLCB does not point back (QLLCB.network != QLLMDB index or QLLCB.far_end != endpoint).
Possible Error Effect:	If the QLLMDB owns a QLLCB on endpoint 0, the QLLCB and any associated resources are effectively lost to the system. This could prevent a valid endpoint from coming up and so degrade the QLPS network. If the linkage from a QLLMDB to a QLLCB for a valid endpoint is broken, the

	endpoint may not be able to communicate with the QLPS network. The QLPS network may be degraded.
Corrective Action Taken:	If a QLLCB is linked to the QLLMDB for endpoint 0 (that is, QLLCB.network = QLLMDB.index and QLLCB.far_end = 0), the audit releases the QLLCB and any associated resources. In all cases, the audit clears the endpoint entry being checked (QLLMDB.llcb_ix[endpoint] = QLLIXNULL).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Endpoint index (index into QLLMDB.llcb_ix[]).
<i>Bad Data =</i>	QLLCB index from QLLMDB.llcb_ix[endpoint] (index into QLLlcb[]).
<i>Good Data =</i>	Logical link's endpoint (QLLCB.far_end).
<i>Logical Index =</i>	Index of the QLLMDB (index into QLLmdb[]).
<i>Dump 1 =</i>	QLLCB.
<i>Dump 2 =</i>	QLLMDB without the endpoint map (QLImdb[]).llcb_ix[]).

Table 7 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A QLPS network link manager's requester field is out-of-range (QLLMDB.requester ≥ QLREQMAX).
Possible Error Effect:	Operational code may assert on the out-of-range value.
Corrective Action Taken:	The audit sets QLLMDB.requester to the default value (QLNONE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Invalid requester (QLLMDB.requester).
<i>Good Data =</i>	Link manager's state (QLLMDB.state).
<i>Logical Index =</i>	Index of the QLLMDB (index into QLLmdb[]).
<i>Dump =</i>	QLLMDB without the endpoint map (QLImdb[]).llcb_ix[]).

Table 8 Error Code: STATE

Error Code:	STATE
Error Description:	An QLPS network link manager is in an invalid state (QLLMDB.state is out-of-range).
Possible Error Effect:	The QLPS network pipe associated with the QLLMDB is not usable. The QLPS network may be degraded. It is also likely that asserts fire.
Corrective Action Taken:	The audit invokes error handling code to take down the QLPS network pipe associated with this QLLMDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	QLLMDB requester field.
<i>Bad Data =</i>	Bad QLPS network link manager state (QLLMDB.state).
<i>Good Data =</i>	Number of loopback retries (QLLMDB.retry).
<i>Logical Index =</i>	Index of the QLLMDB (index into QLLmdb[]).
<i>Dump =</i>	QLLMDB without the endpoint map (QLImdb[]).llcb_ix[]).

Table 9 Error Code: TIME_EXC

Error Code:	TIME_EXC
Error Description:	A QLPS network link manager is in an active or available state (QLLMDB.state = QLMACT or QLMOOSAVL), but does not have a five second timer (QLLMDB.timer[QLT5SECTIMER].type is not valid).
Possible Error Effect:	The QLPS network pipe associated with the QLLMDB may be stuck in an inappropriate protocol state and so not handle QLPS traffic correctly. The QLPS network may be degraded.
Corrective Action Taken:	The audit starts the five second timer.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	type in the five second timer (QLLMDB.timer[QLT5SECTIMER].type).
<i>Bad Data =</i>	count in the five second timer (QLLMDB.timer[QLT5SECTIMER].count).
<i>Good Data =</i>	Link manager's state (QLLMDB.state).

<i>Logical Index</i> =	Index of the QLLMDB (index into QLImdbf[]).
<i>Dump</i> =	QLLMDB without the endpoint map (QLImdbf[]..llcb_ix[]).

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write-protect violation, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit is reentered to clean up any data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	In the MH or QGP, this is the base level loop index (that is, an indicator of what code was executing).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the last QLLMDB audited (index into QLImdbf[]).
<i>Dump</i> =	QLLMDB without the endpoint map (QLImdbf[]..llcb_ix[]).

LNKDATA

Software Release: 5E11 and later

1. LINK STATUS DATA (LNKDATA) AUDIT

LNKDATA audit steps through the RLAPC_INFO (adjacent point code information) relation. For each tuple in the RLAPC_INFO relation, the audit makes sure the existence of rLNK_STAT (link status) tuple. Any missing tuple will be inserted by the audit. It also checks the dynamic RLLNK_STAT and RLLS_STAT (link set status) relations against the static relations RLAPC_INFO and RLDSLEQUIP for the correctness of semantic data.

2. DATA STRUCTURES

For details on the fields contained in the RLLNK_STAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_FAILED

Error Code:	BAD_FAILED
Error Description:	Port in-service, but marked failed in the RLLNK_STAT relation.
Possible Error Effect:	Status of links in missing link set tuple cannot be changed. Change over and change back do not occur properly. Link cannot be inhibited or uninhibited. Other link status changes do not occur properly.
Corrective Action Taken:	Audit calls signaling link message function.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLNK_STAT tuple.
<i>Bad Data =</i>	Failed in RLLNK_STAT.
<i>Good Data =</i>	Port.
<i>Logical Index =</i>	Key of RLLNK_STAT.
<i>Dump =</i>	NA.

Table 2 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The tuple with key dltpor from the RLLNK_STAT relation was not found in the RLLNKP_LS relation.
Possible Error Effect:	Signaling link selection (SLSs) fields are not assigned to links correctly, so message signal unit (MSU) traffic may not be distributed correctly.
Corrective Action Taken:	None.
Manual Action Required:	Investigate office-dependent data (ODD) relation RLLNKP_LS.
Dump Description:	
<i>Error Address =</i>	Address of the rLNK_STAT tuple.
<i>Bad Data =</i>	0 or 1 (to distinguish the two occurrences of this error).
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	dltpor value from rILNK_STAT tuple.
<i>Dump</i> =	rILNK_STAT tuple.

Table 3 Error Code: BAD_LNK_STAT_AVAIL

Error Code:	BAD_LNK_STAT_AVAIL
Error Description:	Link inhibited, blocked, or failed, but marked available in the RLLNK_STAT relation.
Possible Error Effect:	Traffic routed to link is lost.
Corrective Action Taken:	Audit corrects available.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILNK_STAT tuple.
<i>Bad Data</i> =	Avail in RLLNK_STAT.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of RLLNK_STAT.
<i>Dump</i> =	Tuple in error from RLLNK_STAT.

Table 4 Error Code: BAD_SDLCO

Error Code:	BAD_SDLCO
Error Description:	The port for this link set member does not have an associated terminal process. The value of sdlco_ack should be DBNO, but it is not.
Possible Error Effect:	Signaling data link connection order messages are improperly processed.
Corrective Action Taken:	The audit corrects sdlco_ack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILNK_STAT tuple.
<i>Bad Data</i> =	Value of sdlco_ack in RLLNK_STAT.
<i>Good Data</i> =	0 (DBNO).
<i>Logical Index</i> =	lnk_indx (key) of RLLNK_STAT.
<i>Dump</i> =	NA.

Table 5 Error Code: BAD_STPH

Error Code:	BAD_STPH
Error Description:	Signaling terminal protocol handler (STPH) in the RLLNK_STAT relation is not equal to dsl_group in the RLDSLEQUIP relation.
Possible Error Effect:	Incorrect STPH is given to the signaling message handler (SMH).
Corrective Action Taken:	Audit corrects value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILNK_STAT tuple.
<i>Bad Data</i> =	stph in RLLNK_STAT.
<i>Good Data</i> =	dsl_group in RLDSLEQUIP.
<i>Logical Index</i> =	Key of RLLNK_STAT.
<i>Dump</i> =	NA.

Table 6 Error Code: CCNON

Error Code:	CCNON
Error Description:	In order to set the norm_sls field, the audit must invoke a common channel signaling (CCS) function. However, this function was not able to provide the norm_sls needed to construct or check the tuple.
Possible Error Effect:	SLSs fields are not assigned to link correctly. Therefore, MSU traffic may not be distributed correctly.
Corrective Action Taken:	The audit tries again the next time it runs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rILS_STAT tuple.
<i>Bad Data</i> =	Return from the CCS function call.
<i>Good Data</i> =	One of the following: 1 = Indicates call made for a missing rILNK_STAT tuple.

	2 = Indicates call made for an existing rLNK_STAT tuple.
<i>Logical Index</i> =	lsnum(key) of rLS_STAT.
<i>Dump</i> =	NA.

Table 7 Error Code: CHNUM_BAD

Error Code:	CHNUM_BAD
Error Description:	The chnum (channel number) field of this rLNK_STAT tuple differs from the logical channel value derived from the member field of the corresponding rDSLEQUIP tuple.
Possible Error Effect:	It is impossible to identify the correct protocol handler (PH) channel corresponding to this link.
Corrective Action Taken:	The audit corrects the chnum field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLNK_STAT tuple.
<i>Bad Data</i> =	Value of chnum in RLLNK_STAT.
<i>Good Data</i> =	Logical channel value derived from member in corresponding rDSLEQUIP tuple.
<i>Logical Index</i> =	lnk_indx (key) of rLNK_STAT tuple.
<i>Dump</i> =	NA.

Table 8 Error Code: CPRIOR

Error Code:	CPRIOR
Error Description:	The priority field of an rLNK_STAT tuple differs from the priority field in the corresponding rLNKP_LS tuple.
Possible Error Effect:	A link that should be carrying traffic is not doing so, or one that should not be carrying traffic is doing so.
Corrective Action Taken:	The priority field of the rLNK_STAT tuple is corrected to that from the rLNKP_LS tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLNK_STAT tuple.
<i>Bad Data</i> =	Priority field of rLNK_STAT tuple.
<i>Good Data</i> =	Priority field of rLNKP_LS tuple.
<i>Logical Index</i> =	lnk_indx (key) field of rLNK_STAT tuple.
<i>Dump</i> =	rLNK_STAT tuple.

Table 9 Error Code: DLTPORT_BAD

Error Code:	DLTPORT_BAD
Error Description:	Dltport in the RLLNK_STAT relation is not equal to sdlport in the RLLS_LNK relation.
Possible Error Effect:	Incorrect data about STPH could be transmitted to the SMH.
Corrective Action Taken:	Audit corrects value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLNK_STAT tuple.
<i>Bad Data</i> =	dltport in RLLNK_STAT.
<i>Good Data</i> =	sdlport in RLLS_LNK.
<i>Logical Index</i> =	Key of RLLNK_STAT.
<i>Dump</i> =	NA.

Table 10 Error Code: LNKISET_BAD

Error Code:	LNKISET_BAD
Error Description:	Value of link index of member of array lnkiset in the RLLS_STAT relation not equal to lnkindx of same member of same lsnum of the RLAPC_INFO relation.
Possible Error Effect:	Link status changes do not occur properly.
Corrective Action Taken:	Audit corrects value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLS_STAT tuple.

<i>Bad Data</i> =	Link index in RLLS_STAT.
<i>Good Data</i> =	Link index in RLAPC_INFO.
<i>Logical Index</i> =	Key of RLLS_STAT.
<i>Dump</i> =	Tuple in error from RLLS_STAT.

Table 11 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The tuple with key lkindx from the RLAPC_INFO relation was not found in dynamic relation RLLNK_STAT.
Possible Error Effect:	Data for link is not inserted in dynamic data. Status of link cannot be changed. Change over and change back do not occur properly. Other link status changes do not occur properly.
Corrective Action Taken:	Audit tries to construct and insert missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPC_INFO tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	lkindx from RLAPC_INFO
<i>Logical Index</i> =	apcindx (key) of RLAPC_INFO
<i>Dump</i> =	NA.

Table 12 Error Code: NORM1_SLS_BAD

Error Code:	NORM1_SLS_BAD
Error Description:	The norm_sls field of an rLNK_STAT tuple differs from the norm_sls field in the corresponding rLNKP_LS tuple.
Possible Error Effect:	SLSs fields are not assigned to link correctly, so MSU traffic may not be distributed correctly.
Corrective Action Taken:	The norm_sls field of the rLNK_STAT tuple is corrected to that from the rLNKP_LS tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLNK_STAT tuple.
<i>Bad Data</i> =	norm_sls field of rLNK_STAT tuple.
<i>Good Data</i> =	norm_sls field of rLNKP_LS tuple.
<i>Logical Index</i> =	lnk_indx (key) field of rLNK_STAT tuple.
<i>Dump</i> =	rLNK_STAT tuple.

Table 13 Error Code: NORM2_SLS_BAD

Error Code:	NORM2_SLS_BAD
Error Description:	The norm_sls field of an rLNK_STAT tuple differs from the norm_sls field returned by the CCS function.
Possible Error Effect:	SLSs fields are not assigned to link correctly. Therefore, MSU traffic may not be distributed correctly.
Corrective Action Taken:	None. The function has already put the correct norm_sls into the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLNK_STAT tuple.
<i>Bad Data</i> =	norm_sls field of the rLNK_STAT tuple.
<i>Good Data</i> =	norm_sls field returned by the function.
<i>Logical Index</i> =	lnk_indx (key) field of the rLNK_STAT tuple.
<i>Dump</i> =	NA.

Table 14 Error Code: NO_1DSLEQUIP

Error Code:	NO_1DSLEQUIP
Error Description:	Tuple with key st_dlt from the RLLS_LNK relation is not found in the RLDSLEQUIP relation.
Possible Error Effect:	Data for link is not inserted in dynamic data. Status of link cannot be changed. Change over and change back do not occur properly. Other link status changes do not occur properly.
Corrective Action Taken:	None.

Manual Action Required:	Investigate ODD (office-dependent data) relation RLDSLEQUIP.
Dump Description:	
<i>Error Address</i> =	Address of rILS_LNK tuple.
<i>Bad Data</i> =	member in RLLS_LNK (second half of key to RLLS_LNK).
<i>Good Data</i> =	St_dlt in RLLS_LNK.
<i>Logical Index</i> =	lsum in RLLS_LNK (first half of key to RLLS_LNK).
<i>Dump</i> =	NA.

Table 15 Error Code: NO_1LS_LNK

Error Code:	NO_1LS_LNK
Error Description:	Tuple with keys lsum and member from the RLAPC_INFO relation not found in static relation RLLS_LNK.
Possible Error Effect:	Data for link is not inserted in dynamic data. Status of link cannot be changed. Change over and change back do not occur properly. Other link status changes do not occur properly.
Corrective Action Taken:	None.
Manual Action Required:	Investigate ODD relation RLLS_LNK.
Dump Description:	
<i>Error Address</i> =	Address of rLAPC_INFO tuple.
<i>Bad Data</i> =	lsum in RLAPC_INFO.
<i>Good Data</i> =	member in RLAPC_INFO.
<i>Logical Index</i> =	Key of RLAPC_INFO.
<i>Dump</i> =	NA.

Table 16 Error Code: NO_2DSLEQUIP

Error Code:	NO_2DSLEQUIP
Error Description:	Tuple with key st_dlt from the RLLS_LNK relation is not found in the RLDSLEQUIP relation.
Possible Error Effect:	During selective initialization, the dynamic data may not be populated correctly resulting in inability to change the status of the link.
Corrective Action Taken:	None.
Manual Action Required:	Investigate ODD relation RLDSLEQUIP.
Dump Description:	
<i>Error Address</i> =	Address of rILS_LNK tuple.
<i>Bad Data</i> =	member in RLLS_LNK (second half of key to RLLS_LNK).
<i>Good Data</i> =	st_dlt in RLLS_LNK.
<i>Logical Index</i> =	lsum in RLLS_LNK (first half of key to RLLS_LNK).
<i>Dump</i> =	NA.

Table 17 Error Code: NO_2LS_LNK

Error Code:	NO_2LS_LNK
Error Description:	Tuple with keys lsum and member from the RLAPC_INFO relation is not found in static relation RLLS_LNK.
Possible Error Effect:	Status of link cannot be changed. Change over and change back do not occur properly. Other link status changes do not occur properly.
Corrective Action Taken:	None.
Manual Action Required:	Investigate ODD relation RLLS_LNK.
Dump Description:	
<i>Error Address</i> =	Address of RLAPC_INFO.
<i>Bad Data</i> =	lsum in RLAPC_INFO (first half of key to RLLS_LNK).
<i>Good Data</i> =	member in RLAPC_INFO (second half of key to RLLS_LNK).
<i>Logical Index</i> =	Key of RLAPC_INFO.
<i>Dump</i> =	NA.

Table 18 Error Code: NO_MATCH_LS_STAT

Error Code:	NO_MATCH_LS_STAT
Error Description:	Tuple in static data does not exist in the dynamic relation, RLLS_STAT.
Possible Error Effect:	Status of links in missing link set tuple cannot be changed. Change over and change back do not occur properly. Link cannot be inhibited or uninhibited. Other link status changes do not occur properly.
Corrective Action Taken:	Related audit runs and inserts tuple in the RLLS_STAT relation.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Link set number in RLAPC_INFO.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of RLAPC_INFO.
<i>Dump</i> =	NA.

Table 19 Error Code: NO_PORTLA

Error Code:	NO_PORTLA
Error Description:	Port derived from dltport of the RLLNK_STAT relation not found in dynamic relation RLPORTLA.
Possible Error Effect:	Status of port cannot be verified. Link cannot be activated.
Corrective Action Taken:	Related audit corrects RLPORTLA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLLNK_STAT tuple.
<i>Bad Data</i> =	dltport in RLLNK_STAT.
<i>Good Data</i> =	Port derived from dltport.
<i>Logical Index</i> =	Key of RLLNK_STAT.
<i>Dump</i> =	NA.

Table 20 Error Code: SLC_BAD

Error Code:	SLC_BAD
Error Description:	slc in the RLLNK_STAT relation not equal to slc in the RLAPC_INFO relation.
Possible Error Effect:	SMH receives incorrect routing data.
Corrective Action Taken:	Audit corrects value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLLNK_STAT tuple.
<i>Bad Data</i> =	slc in RLLNK_STAT.
<i>Good Data</i> =	slc in RLAPC_INFO.
<i>Logical Index</i> =	Key of RLLNK_STAT.
<i>Dump</i> =	NA.

LNKSLS-A

Software Release: 5E11 - 5E13

1. SIGNALING LINK SELECTION (LNKSLS) AUDIT

The LNKSLS audit steps through the RLLS_GRP (link set group relation) and checks the data in the RLLNK_STAT (link state relation) and RLLS_STAT (link set status relation).

2. DATA STRUCTURES

For details on the fields contained in the RLLS_GRP, RLLNK_STAT and RLLS_STAT relations refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AVAIL_BAD_LS_STAT

Error Code:	AVAIL_BAD_LS_STAT
Error Description:	The cur_status field in the NMslsc7buff[] entry has an incorrect value.
Possible Error Effect:	An incorrect signaling link (SL) status could be reported to the network management center.
Corrective Action Taken:	The audit sets the cur_status field in NMslsc7buff[] opposite from the avail attribute in rLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the tuple containing the error in NMslsc7buff[].
<i>Bad Data =</i>	cur_status field in NMslsc7buff[].
<i>Good Data =</i>	avail attribute in rLS_STAT tuple.
<i>Logical Index =</i>	Index to NMslsc7buff[].
<i>Dump =</i>	NA.

Table 2 Error Code: BAD_CONGEST

Error Code:	BAD_CONGEST
Error Description:	One of the congestion indicators (cong1, cong2, or cong3) is not consistent with the cg field in the rLNK_STAT tuple.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	The audit corrects the congestion field (cong1, cong2, or cong3) in NMslsc7buff[], depending on the value of the cg attribute in the rLNK_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the tuple containing the error in NMslsc7buff[].
<i>Bad Data =</i>	Congestion value. The three bit values obtained by combining the bit values of cong1, cong2, and cong3 fields in NMslsc7buff[]. For example, if bit0 is set, then cong1 in NMslsc7buff[] is DBYES.
<i>Good Data =</i>	Value of cg field in rLNK_STAT tuple.
<i>Logical Index =</i>	Index to NMslsc7buff[].

<i>Dump</i> =	NA.

Table 3 Error Code: BAD_DDTBL

Error Code:	BAD_DDTBL
Error Description:	Unable to retrieve the gsml field from the RL GSM OPC relation.
Possible Error Effect:	The slsid field in both NMslsc7buff[] and NMslc7buff[] cannot be audited.
Corrective Action Taken:	None.
Manual Action Required:	Populate the rLGSM OPC table.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The key field (Gl_pcrld) to read relation RL GSM OPC.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: BAD_FAILED

Error Code:	BAD_FAILED
Error Description:	The slfail field in NMslc7buff[] does not agree with the failed attribute in the rLNK_STAT tuple.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	The audit sets the slfail field in NMslc7buff[] to agree with the failed attribute in the rLNK_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the tuple containing the error in NMslc7buff[].
<i>Bad Data</i> =	slfail field in NMslc7buff[].
<i>Good Data</i> =	Failed attribute in the rLNK_STAT tuple.
<i>Logical Index</i> =	Index to NMslc7buff[].
<i>Dump</i> =	NA.

Table 5 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The audit cannot read static relation APC_INFO with key retrieved from rLS_STAT (apci attribute) and rLNK_STAT (slc attribute), or the audit cannot read (static) relation LNKP_LS with key from rLNK_STAT (attribute dltport). Good data is used to differentiate between the two cases.
Possible Error Effect:	The possible error effects are: 1 = If only the slc attribute is wrong, incorrect updates may be sent to the protocol handler (PH) during traffic diversion. If either the slc or the apci is wrong, then operational code is not able to map link set to APC (adjacent point code). 2 = If the dltport attribute is wrong, operational code is not able to access data about link from static relation LNKP_LS.
Corrective Action Taken:	The audit takes a longer route to read rAPC_INFO and rLNKP_LS. First it reads rLS_LNK (using link set number and index into the lnkiset array), and then rLNKP_LS and rAPC_INFO. If the audit cannot successfully read rLS_LNK, it assumes that the link index in the lnkiset array is bad and zeroes it to audit the next link index. If the link index does not match with lnkidx in rAPC_INFO, the audit sets the link index in the lnkiset array to the value of lnkidx in rAPC_INFO and proceeds with the rest of the checks.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Corresponds to the Possible Error Effects as follows:

	1 = audit cannot read rIAPC_INFO.
	2 = audit cannot read rILNKP_LS.
<i>Logical Index</i> =	Logical key (link set number) of rILS_STAT.
<i>Dump</i> =	Contents of rILS_STAT and rILNK_STAT tuples.

Table 6 Error Code: CHANGE_OVER

Error Code:	CHANGE_OVER
Error Description:	The error is due to one of the following: 1 = The value of the laco field within one tuple of NMslc7buf[] is not DBNO when the deloaded attribute in rILNK_STAT tuple has the value of DBNO. 2 = The value of the lmco field within one tuple of NMslc7buf[] is not DBNO when the deloaded attribute in rILNK_STAT tuple has the value of DBNO. 3 = Neither the laco field nor the lmco field within one tuple of NMslc7buf[] has the value of DBYES when the deloaded attribute in rILNK_STAT tuple is DBYES, the norm_sls attribute in the rILNK_STAT tuple equals zero, and the sls attribute in rILNK_STAT tuple is zero.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit corrects the laco field in NMslc7buf[] to the value of DBNO. 2 = The audit corrects the lmco field in NMslc7buf[] to the value of DBNO. 3 = The audit sets the laco field to the value of DBYES and the lmco field to the value of DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the tuple containing the error in NMslc7buf[].
<i>Bad Data</i> =	Corresponds to the Error Description and Corrective Action as follows: 1 = laco field in NMslc7buf[]. 2 = lmco field in NMslc7buf[]. 3 = laco field in NMslc7buf[].
<i>Good Data</i> =	Corresponds to the Error Description as follows: 1 = Indicates that the laco field in NMslc7buf[] is incorrect. 2 = Indicates that the lmco field in NMslc7buf[] is incorrect. 3 = Indicates that the laco field in NMslc7buf[] is incorrect.
<i>Logical Index</i> =	Index to NMslc7buf[].
<i>Dump</i> =	NA.

Table 7 Error Code: LINKISET_ERROR

Error Code:	LINKISET_ERROR
Error Description:	The link index in lnlkiset array of rILS_STAT is bad. There are three cases; good data is used to differentiate between them. The audit cannot read rILNK_STAT, link index in lnlkiset array does not agree with lnlindx in rIAPC_INFO, or audit

	cannot read static relation LS_LNK.
Possible Error Effect:	Operational code is not able to access link data or may access data for the wrong link. Traffic diversion for the link in question may not be done at all or may be done incorrectly.
Corrective Action Taken:	For the first two cases (good data 1 or 2), the audit sets the link index in Inksset to agree with the Inkindx in rIAPC_INFO. In the last case, the audit zeroes the link index in Inksset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index into Inksset array (0-7).
<i>Bad Data</i> =	Link index (contents of Inksset array).
<i>Good Data</i> =	Valid values are: 1 = Audit cannot read rLNK_STAT using link index from Inksset array. 2 = Link index (of Inksset) does not match with Inkindx in static data in rIAPC_INFO. 3 = Audit cannot find rLS_LNK tuple using link set number and position into Inksset array as a key.
<i>Logical Index</i> =	Logical key (link set) of rLS_STAT.
<i>Dump</i> =	NA.

Table 8 Error Code: LINK_SLS

Error Code:	LINK_SLS
Error Description:	A link set is marked unavailable, but some of its links carry signaling link selections (SLSs), or a link that is marked available does not carry its normal SLSs. Good data is used to differentiate between the two cases.
Possible Error Effect:	Links may be carrying bogus traffic, or traffic for the link could be lost.
Corrective Action Taken:	The audit sends the STRM_SP (signaling traffic and route management system process) a message to do SLS correction. Message sent is type MGSS7TRM with msubtype attribute set to CCS7DSC. If the link set is marked unavailable and carries SLSs, the sls field for the link is set to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Index into Inksset array (0-7).
<i>Bad Data</i> =	Link index (contents of Inksset).
<i>Good Data</i> =	Valid values are: 1 = Link set is marked unavailable but one of its links carries SLSs. 2 = A link is marked available but does not carry all its normal SLSs.
<i>Logical Index</i> =	Logical key (link set) of rLS_STAT.
<i>Dump</i> =	Contents of rLNK_STAT tuple.

Table 9 Error Code: LS_STAT_NO

Error Code:	LS_STAT_NO
Error Description:	The link set contained in static relation LS_GRP is not in dynamic relation LS_STAT.
Possible Error Effect:	Operational code is not able to access dynamic data about link set or any link in the link set. No traffic diversion for the missing link or the missing link set can be performed.
Corrective Action Taken:	This audit reports the error and has related audit, LSDATA, insert the missing link set into rLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (link set number) of rLS_STAT.
<i>Dump</i> =	NA.

Table 10 Error Code: MEMBER_BAD

Error Code:	MEMBER_BAD
Error Description:	The member field in NMslc7buff[] does not agree with the index into the lnkiset[] attribute in the rLS_STAT tuple.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	The audit sets the member field in NMslc7buff[] to agree with the index into the lnkiset[] attribute in the rLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the tuple containing the error in NMslc7buff[].
<i>Bad Data</i> =	member field in NMslc7buff[].
<i>Good Data</i> =	Index into the lnkiset[] attribute in the rLS_STAT tuple.
<i>Logical Index</i> =	Index to NMslc7buff[].
<i>Dump</i> =	NA.

Table 11 Error Code: MLINK_LNK_STAT

Error Code:	MLINK_LNK_STAT
Error Description:	A link index exists in static data (rAPC_INFO) data, but does not exist in dynamic rLNK_STAT.
Possible Error Effect:	Traffic diversion for that link is not performed.
Corrective Action Taken:	The audit reports the error and has related audit, LNKDATA, insert the tuple in rLNK_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLS_STAT tuple.
<i>Bad Data</i> =	Missing link index tuple.
<i>Good Data</i> =	Index into lnkiset array (0-7).
<i>Logical Index</i> =	Logical key (link set) of rLS_STAT.
<i>Dump</i> =	Contents of rLS_STAT tuple.

Table 12 Error Code: NO_PORTLA

Error Code:	NO_PORTLA
Error Description:	The port derived from dltport of rLNK_STAT was not found in the dynamic relation rPORTLA.
Possible Error Effect:	The status of the port cannot be verified. Links cannot be activated.
Corrective Action Taken:	A related audit corrects rPORTLA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLNK_STAT tuple.
<i>Bad Data</i> =	dltport in rLNK_STAT.
<i>Good Data</i> =	Port derived from the TMDLTPORT[] macro.
<i>Logical Index</i> =	Index to NMslc7buff[]
<i>Dump</i> =	NA.

Table 13 Error Code: RDATA_BAD

Error Code:	RDATA_BAD
Error Description:	Data in lnkiset array does not match with static data of rAPC_INFO. There are three cases; good data is used to differentiate them. The link set number, the link index, or the position into lnkiset array does not match with static data in rAPC_INFO.
Possible Error Effect:	Operational code is not able to access link data or may access data for the wrong link. Traffic diversion for the link in question may not be done at all or may be done incorrectly.
Corrective Action Taken:	The audit sets the lnkiset data to agree with static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLS_STAT.

<i>Bad Data</i> =	Depending on good data, the bad link set number, the bad link index, or the bad index into the lnkiset array.
<i>Good Data</i> =	Valid values are: 1 = The link set number does not match with the lsnum attribute in rIAPC_INFO. 2 = Index in lnkiset does not match with the lnkindx attribute in rIAPC_INFO. 3 = Into lnkiset array does not match with the member attribute in rIAPC_INFO.
<i>Logical Index</i> =	Logical index (link set) of rILS_STAT.
<i>Dump</i> =	Contents of rILS_STAT tuple, and contents of rILNK_STAT tuple.

Table 14 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	Static to static inconsistency. There are three cases; good data is used to differentiate between them. The audit cannot read the LNKP_LS tuple using the st_dlt attribute from static relation LS_LNK as key, the audit cannot read APC_INFO tuple using attribute apci from rILS_GRP and slc from LNKP_LS, or the member attribute in rILS_LNK does not match with the member attribute in rIAPC_INFO.
Possible Error Effect:	Operational code is not able to access link data or may access data for the wrong link. Traffic diversion for the link in question may not be done at all or may be done incorrectly.
Corrective Action Taken:	None.
Manual Action Required:	Resolve the office-dependent data (ODD) inconsistencies.
Dump Description:	
<i>Error Address</i> =	If <i>Good Data</i> is 1, then <i>Error Address</i> is the st_dlt attribute of rILS_LNK. If <i>Good Data</i> is 2, then <i>Error Address</i> is the apci of rILS_GRP. If <i>Good Data</i> is 3, then <i>Error Address</i> is the member value of rIAPC_INFO.
<i>Bad Data</i> =	If <i>Good Data</i> is 1, then <i>Bad Data</i> is NA. If <i>Good Data</i> is 2, then <i>Bad Data</i> is the slc attribute of rILNK_LS. If <i>Good Data</i> is 3, then <i>Bad Data</i> is the index into lnkiset array (0-7).
<i>Good Data</i> =	Valid values are: 1 = Audit cannot read rILNKP_LS using st_dlt of rILS_LNK. 2 = Audit cannot read rIAPC_INFO tuple. 3 = Member attribute of rILS_LNK does not match the member attribute of rIAPC_INFO.
<i>Logical Index</i> =	Logical key (link set) of rILS_STAT
<i>Dump</i> =	NA.

Table 15 Error Code: SLSID_BAD

Error Code:	SLSID_BAD
Error Description:	The error is due to one of the following: 1 = One entry field (slsid) for NMslsc7buf[] is incorrect. 2 = One entry field (slsid) for NMslc7buf[] is incorrect.
Possible Error Effect:	An incorrect slsid could be reported to the network management center.
Corrective Action Taken:	Corresponds the Error Description as follows: 1 = The audit corrects the slsid field in the NMslsc7buf[].

	2 = The audit corrects the slsid field in NMslsc7buf[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Corresponds the Error Description as follows: 1 = Starting address of the tuple containing the error NMslsc7buf[]. 2 = Starting address of the tuple containing the error NMslsc7buf[].
<i>Bad Data</i> =	Corresponds the Error Description as follows: 1 = slsid field in NMslsc7buf[]. 2 = slsid field in NMslsc7buf[].
<i>Good Data</i> =	Corresponds the Error Description as follows: 1 = indicates that the slsid in NMslsc7buf[] is incorrect. 2 = indicates that the slsid in NMslsc7buf[] is incorrect.
<i>Logical Index</i> =	Corresponds the Error Description as follows: 1 = Isnum attribute in the rLS_STAT tuple. 2 = lnk_indx attribute in the rLNK_STAT tuple.
<i>Dump</i> =	Correct slsid.

Table 16 Error Code: SLS_CLINK

Error Code:	SLS_CLINK
Error Description:	All SLSs are not carried by links in link set, or SLSs are carried by more than one link in link set. Good data differentiates between the two cases.
Possible Error Effect:	The possible error effects are: 1 = Traffic for link set is lost. 2 = The wrong link can be carrying traffic and may be carrying more than its share of traffic.
Corrective Action Taken:	The audit sends the STRM_SP (signaling traffic and route management system process) a message to do SLS correction. Message sent is type MGSS7TRM with msubtype attribute set to CCS7DSC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Corresponds to Possible Error Effects as follows: 1 = All SLSs are not carried by links in link set. 2 = SLSs are carried by more than one link within the link set.
<i>Logical Index</i> =	Logical key (link set) of rLS_STAT.
<i>Dump</i> =	Contents of rLS_STAT tuple, and contents of array holding SLSs for each link index.

Table 17 Error Code: SLS_EXTRA

Error Code:	SLS EXTRA
Error Description:	The error is due to one of the following: 1 = An extra entry in NMslsc7buf[] is found. 2 = An extra entry in NMslc7buf[] is found.
Possible Error Effect:	The extra SLS or SL could be reported to the network management center.
Corrective Action Taken:	The audit deletes the extra entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 1 = Starting address of the tuple containing the error NMslsc7buf[]. 2 = Starting address of the tuple containing the error NMslc7buf[].
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 1 = slsid field in NMslsc7buf[]. 2 = slsid field in NMslc7buf[].
<i>Good Data =</i>	Corresponds to the Error Description as follows: 1 = Indicates that an extra entry to NMslsc7buf[] is found. 2 = Indicates that an extra entry to NMslc7buf[] is found.
<i>Logical Index =</i>	Corresponds to the Error Description as follows: 1 = Index to NMslsc7buf[]. 2 = Index to NMslc7buf[].
<i>Dump =</i>	NA.

LNKSLS-B

Software Release: 5E14 and later

1. SIGNALING LINK SELECTION (LNKSLS) AUDIT

The LNKSLS audit steps through the RLLS_GRP (link set group relation) and checks the data in the RLLNK_STAT (link state relation) and RLLS_STAT (link set status relation).

2. DATA STRUCTURES

For details on the fields contained in the RLLS_GRP, RLLNK_STAT and RLLS_STAT relations refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AVAIL_BAD_LS_STAT

Error Code:	AVAIL_BAD_LS_STAT
Error Description:	The cur_status field in the NMslsc7buff[] entry has an incorrect value.
Possible Error Effect:	An incorrect signaling link (SL) status could be reported to the network management center.
Corrective Action Taken:	The audit sets the cur_status field in NMslsc7buff[] opposite from the avail attribute in rLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the tuple containing the error in NMslsc7buff[].
<i>Bad Data =</i>	cur_status field in NMslsc7buff[].
<i>Good Data =</i>	avail attribute in rLS_STAT tuple.
<i>Logical Index =</i>	Index to NMslsc7buff[].
<i>Dump =</i>	NA.

Table 2 Error Code: BAD_CONGEST

Error Code:	BAD_CONGEST
Error Description:	One of the congestion indicators (cong1, cong2, or cong3) is not consistent with the cg field in the rLNK_STAT tuple.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	The audit corrects the congestion field (cong1, cong2, or cong3) in NMslsc7buff[], depending on the value of the cg attribute in the rLNK_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Starting address of the tuple containing the error in NMslsc7buff[].
<i>Bad Data =</i>	Congestion value. The three bit values obtained by combining the bit values of cong1, cong2, and cong3 fields in NMslsc7buff[]. For example, if bit0 is set, then cong1 in NMslsc7buff[] is DBYES.
<i>Good Data =</i>	Value of cg field in rLNK_STAT tuple.
<i>Logical Index =</i>	Index to NMslsc7buff[].

<i>Dump</i> =	NA.

Table 3 Error Code: BAD_DDTBL

Error Code:	BAD_DDTBL
Error Description:	Unable to retrieve the gsml field from the RL GSM OPC relation.
Possible Error Effect:	The slsid field in both NMslsc7buff[] and NMslc7buff[] cannot be audited.
Corrective Action Taken:	None.
Manual Action Required:	Populate the rLGSM_OPC table.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The key field (Gl_pcrld) to read relation RL GSM OPC.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: BAD_FAILED

Error Code:	BAD_FAILED
Error Description:	The slfail field in NMslc7buff[] does not agree with the failed attribute in the rLNK_STAT tuple.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	The audit sets the slfail field in NMslc7buff[] to agree with the failed attribute in the rLNK_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the tuple containing the error in NMslc7buff[].
<i>Bad Data</i> =	slfail field in NMslc7buff[].
<i>Good Data</i> =	Failed attribute in the rLNK_STAT tuple.
<i>Logical Index</i> =	Index to NMslc7buff[].
<i>Dump</i> =	NA.

Table 5 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The audit cannot read static relation APC_INFO with key retrieved from rLS_STAT (apci attribute) and rLNK_STAT (slc attribute), or the audit cannot read (static) relation LNKP_LS with key from rLNK_STAT (attribute dltport). Good data is used to differentiate between the two cases.
Possible Error Effect:	The possible error effects are: 1 = If only the slc attribute is wrong, incorrect updates may be sent to the protocol handler (PH) during traffic diversion. If either the slc or the apci is wrong, then operational code is not able to map link set to APC (adjacent point code). 2 = If the dltport attribute is wrong, operational code is not able to access data about link from static relation LNKP_LS.
Corrective Action Taken:	The audit takes a longer route to read rAPC_INFO and rLNKP_LS. First it reads rLS_LNK (using link set number and index into the lnkiset array), and then rLNKP_LS and rAPC_INFO. If the audit cannot successfully read rLS_LNK, it assumes that the link index in the lnkiset array is bad and zeroes it to audit the next link index. If the link index does not match with lnkidx in rAPC_INFO, the audit sets the link index in the lnkiset array to the value of lnkidx in rAPC_INFO and proceeds with the rest of the checks.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Corresponds to the Possible Error Effects as follows:

	1 = audit cannot read rIAPC_INFO.
	2 = audit cannot read rILNK_LS.
<i>Logical Index</i> =	Logical key (link set number) of rILS_STAT.
<i>Dump</i> =	Contents of rILS_STAT and rILNK_STAT tuples.

Table 6 Error Code: CHANGE_OVER

Error Code:	CHANGE_OVER
Error Description:	The error is due to one of the following: 1 = The value of the laco field within one tuple of NMslc7buf[] is not DBNO when the deloaded attribute in rILNK_STAT tuple has the value of DBNO. 2 = The value of the lmco field within one tuple of NMslc7buf[] is not DBNO when the deloaded attribute in rILNK_STAT tuple has the value of DBNO. 3 = Neither the laco field nor the lmco field within one tuple of NMslc7buf[] has the value of DBYES when the deloaded attribute in rILNK_STAT tuple is DBYES, the norm_sls attribute in the rILNK_STAT tuple equals zero, and the sls attribute in rILNK_STAT tuple is zero.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The audit corrects the laco field in NMslc7buf[] to the value of DBNO. 2 = The audit corrects the lmco field in NMslc7buf[] to the value of DBNO. 3 = The audit sets the laco field to the value of DBYES and the lmco field to the value of DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the tuple containing the error in NMslc7buf[].
<i>Bad Data</i> =	Corresponds to the Error Description and Corrective Action as follows: 1 = laco field in NMslc7buf[]. 2 = lmco field in NMslc7buf[]. 3 = laco field in NMslc7buf[].
<i>Good Data</i> =	Corresponds to the Error Description as follows: 1 = Indicates that the laco field in NMslc7buf[] is incorrect. 2 = Indicates that the lmco field in NMslc7buf[] is incorrect. 3 = Indicates that the laco field in NMslc7buf[] is incorrect.
<i>Logical Index</i> =	Index to NMslc7buf[].
<i>Dump</i> =	NA.

Table 7 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit attempts to read an rIAPC_STAT tuple using rILS_STAT.apci as the key. The read fails.
Possible Error Effect:	Certain network management checks cannot be done.

Corrective Action Taken:	The audit continues checking the rLS_STAT tuple and schedules the APCIAPC audit to correct the rIAPC_STAT error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rLS_GRP.Isnum (link set number).
<i>Bad Data =</i>	rIAPC_STAT.apci (adjacent point code index).
<i>Good Data =</i>	rLS_STAT.apci (adjacent point code index).
<i>Logical Index =</i>	rLS_STAT.Isnum (key of tuple which has the error).
<i>Dump 1 =</i>	Contents of rLS_STAT tuple.
<i>Dump 2 =</i>	Contents of rIAPC_STAT tuple.

Table 8 Error Code: LINKSET_ERROR

Error Code:	LINKSET_ERROR
Error Description:	The link index in Inkiset array of rLS_STAT is bad. There are three cases; good data is used to differentiate between them. The audit cannot read rLNK_STAT, link index in Inkiset array does not agree with Inkindx in rIAPC_INFO, or audit cannot read static relation LS_LNK.
Possible Error Effect:	Operational code is not able to access link data or may access data for the wrong link. Traffic diversion for the link in question may not be done at all or may be done incorrectly.
Corrective Action Taken:	For the first two cases (good data 1 or 2), the audit sets the link index in Inkiset to agree with the Inkindx in rIAPC_INFO. In the last case, the audit zeroes the link index in Inkiset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into Inkiset array (0-7).
<i>Bad Data =</i>	Link index (contents of Inkiset array).
<i>Good Data =</i>	Valid values are: 1 = Audit cannot read rLNK_STAT using link index from Inkiset array. 2 = Link index (of Inkiset) does not match with Inkindx in static data in rIAPC_INFO. 3 = Audit cannot find rLS_LNK tuple using link set number and position into Inkiset array as a key.
<i>Logical Index =</i>	Logical key (link set) of rLS_STAT.
<i>Dump =</i>	NA.

Table 9 Error Code: LINK_SLS

Error Code:	LINK_SLS
Error Description:	A link set is marked unavailable, but some of its links carry signaling link selections (SLSs), or a link that is marked available does not carry its normal SLSs. Good data is used to differentiate between the two cases.
Possible Error Effect:	Links may be carrying bogus traffic, or traffic for the link could be lost.
Corrective Action Taken:	The audit sends the STRM_SP (signaling traffic and route management system process) a message to do SLS correction. Message sent is type MGSS7TRM with msbtype attribute set to CCS7DSC. If the link set is marked unavailable and carries SLSs, the sls field for the link is set to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into Inkiset array (0-7).
<i>Bad Data =</i>	Link index (contents of Inkiset).
<i>Good Data =</i>	Valid values are: 1 = Link set is marked unavailable but one of its links carries SLSs. 2 = A link is marked available but does not carry all its normal SLSs.

<i>Logical Index</i> =	Logical key (link set) of rLS_STAT.
<i>Dump</i> =	Contents of rLNK_STAT tuple.

Table 10 Error Code: LS_STAT_NO

Error Code:	LS_STAT_NO
Error Description:	The link set contained in static relation LS_GRP is not in dynamic relation LS_STAT.
Possible Error Effect:	Operational code is not able to access dynamic data about link set or any link in the link set. No traffic diversion for the missing link or the missing link set can be performed.
Corrective Action Taken:	This audit reports the error and has related audit, LSDATA, insert the missing link set into rLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (link set number) of rLS_STAT.
<i>Dump</i> =	NA.

Table 11 Error Code: MEMBER_BAD

Error Code:	MEMBER_BAD
Error Description:	The member field in NMslc7buff[] does not agree with the index into the Inkiset[] attribute in the rLS_STAT tuple.
Possible Error Effect:	An incorrect SL status could be reported to the network management center.
Corrective Action Taken:	The audit sets the member field in NMslc7buff[] to agree with the index into the Inkiset[] attribute in the rLS_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting address of the tuple containing the error in NMslc7buff[].
<i>Bad Data</i> =	member field in NMslc7buff[].
<i>Good Data</i> =	Index into the Inkiset[] attribute in the rLS_STAT tuple.
<i>Logical Index</i> =	Index to NMslc7buff[].
<i>Dump</i> =	NA.

Table 12 Error Code: MLINK_LNK_STAT

Error Code:	MLINK_LNK_STAT
Error Description:	A link index exists in static data (rIAPC_INFO) data, but does not exist in dynamic rLNK_STAT.
Possible Error Effect:	Traffic diversion for that link is not performed.
Corrective Action Taken:	The audit reports the error and has related audit, LNKDATA, insert the tuple in rLNK_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLS_STAT tuple.
<i>Bad Data</i> =	Missing link index tuple.
<i>Good Data</i> =	Index into Inkiset array (0-7).
<i>Logical Index</i> =	Logical key (link set) of rLS_STAT.
<i>Dump</i> =	Contents of rLS_STAT tuple.

Table 13 Error Code: NO_PORTLA

Error Code:	NO_PORTLA
Error Description:	The port derived from dltport of rLNK_STAT was not found in the dynamic relation rIORTLA.
Possible Error Effect:	The status of the port cannot be verified. Links cannot be activated.
Corrective Action Taken:	A related audit corrects rIORTLA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLNK_STAT tuple.
<i>Bad Data</i> =	dltport in rLNK_STAT.
<i>Good Data</i> =	Port derived from the TMDLTPORT[] macro.

<i>Logical Index</i> =	Index to NMslc7buff[]
<i>Dump</i> =	NA.

Table 14 Error Code: RDATA_BAD

Error Code:	RDATA_BAD
Error Description:	Data in Inkiset array does not match with static data of rIAPC_INFO. There are three cases; good data is used to differentiate them. The link set number, the link index, or the position into Inkiset array does not match with static data in rIAPC_INFO.
Possible Error Effect:	Operational code is not able to access link data or may access data for the wrong link. Traffic diversion for the link in question may not be done at all or may be done incorrectly.
Corrective Action Taken:	The audit sets the Inkiset data to agree with static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILS_STAT.
<i>Bad Data</i> =	Depending on good data, the bad link set number, the bad link index, or the bad index into the Inkiset array.
<i>Good Data</i> =	Valid values are: 1 = The link set number does not match with the lsnum attribute in rIAPC_INFO. 2 = Index in Inkiset does not match with the lkindx attribute in rIAPC_INFO. 3 = Into Inkiset array does not match with the member attribute in rIAPC_INFO.
<i>Logical Index</i> =	Logical index (link set) of rILS_STAT.
<i>Dump</i> =	Contents of rILS_STAT tuple, and contents of rILNK_STAT tuple.

Table 15 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	Static to static inconsistency. There are three cases; good data is used to differentiate between them. The audit cannot read the LNKP_LS tuple using the st_dlt attribute from static relation LS_LNK as key, the audit cannot read APC_INFO tuple using attribute apci from rILS_GRP and slc from LNKP_LS, or the member attribute in rILS_LNK does not match with the member attribute in rIAPC_INFO.
Possible Error Effect:	Operational code is not able to access link data or may access data for the wrong link. Traffic diversion for the link in question may not be done at all or may be done incorrectly.
Corrective Action Taken:	None.
Manual Action Required:	Resolve the office-dependent data (ODD) inconsistencies.
Dump Description:	
<i>Error Address</i> =	If <i>Good Data</i> is 1, then <i>Error Address</i> is the st_dlt attribute of rILS_LNK. If <i>Good Data</i> is 2, then <i>Error Address</i> is the apci of rILS_GRP. If <i>Good Data</i> is 3, then <i>Error Address</i> is the member value of rIAPC_INFO.
<i>Bad Data</i> =	If <i>Good Data</i> is 1, then <i>Bad Data</i> is NA. If <i>Good Data</i> is 2, then <i>Bad Data</i> is the slc attribute of rILNK_LS. If <i>Good Data</i> is 3, then <i>Bad Data</i> is the index into Inkiset array (0-7).
<i>Good Data</i> =	Valid values are: 1 = Audit cannot read rILNKP_LS using st_dlt of rILS_LNK. 2 = Audit cannot read rIAPC_INFO tuple. 3 = Member attribute of rILS_LNK does not match the member attribute of

	rlAPC_INFO.
<i>Logical Index</i> =	Logical key (link set) of rLS_STAT
<i>Dump</i> =	NA.

Table 16 Error Code: SLSID_BAD

Error Code:	SLSID_BAD
Error Description:	The error is due to one of the following: 1 = One entry field (slsid) for NMslsc7buf[] is incorrect. 2 = One entry field (slsid) for NMslc7buf[] is incorrect.
Possible Error Effect:	An incorrect slsid could be reported to the network management center.
Corrective Action Taken:	Corresponds the Error Description as follows: 1 = The audit corrects the slsid field in the NMslsc7buf[]. 2 = The audit corrects the slsid field in NMslc7buf[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Corresponds the Error Description as follows: 1 = Starting address of the tuple containing the error NMslsc7buf[]. 2 = Starting address of the tuple containing the error NMslc7buf[].
<i>Bad Data</i> =	Corresponds the Error Description as follows: 1 = slsid field in NMslsc7buf[]. 2 = slsid field in NMslc7buf[].
<i>Good Data</i> =	Corresponds the Error Description as follows: 1 = indicates that the slsid in NMslsc7buf[] is incorrect. 2 = indicates that the slsid in NMslc7buf[] is incorrect.
<i>Logical Index</i> =	Corresponds the Error Description as follows: 1 = lsum attribute in the rLS_STAT tuple. 2 = lnc_idx attribute in the rLNK_STAT tuple.
<i>Dump</i> =	Correct slsid.

Table 17 Error Code: SLS_CLINK

Error Code:	SLS_CLINK
Error Description:	All SLSs are not carried by links in link set, or SLSs are carried by more than one link in link set. Good data differentiates between the two cases.
Possible Error Effect:	The possible error effects are: 1 = Traffic for link set is lost. 2 = The wrong link can be carrying traffic and may be carrying more than its share of traffic.
Corrective Action Taken:	The audit sends the STRM_SP (signaling traffic and route management system

	process) a message to do SLS correction. Message sent is type MGSS7TRM with msubtype attribute set to CCS7DSC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Corresponds to Possible Error Effects as follows: 1 = All SLSs are not carried by links in link set. 2 = SLSs are carried by more than one link within the link set.
<i>Logical Index =</i>	Logical key (link set) of rLS_STAT.
<i>Dump =</i>	Contents of rLS_STAT tuple, and contents of array holding SLSs for each link index.

Table 18 Error Code: SLS_EXTRA

Error Code:	SLS_EXTRA
Error Description:	The error is due to one of the following: 1 = An extra entry in NMslsc7buf[] is found. 2 = An extra entry in NMslc7buf[] is found.
Possible Error Effect:	The extra SLS or SL could be reported to the network management center.
Corrective Action Taken:	The audit deletes the extra entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 1 = Starting address of the tuple containing the error NMslsc7buf[]. 2 = Starting address of the tuple containing the error NMslc7buf[].
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 1 = slsid field in NMslsc7buf[]. 2 = slsid field in NMslc7buf[].
<i>Good Data =</i>	Corresponds to the Error Description as follows: 1 = Indicates that an extra entry to NMslsc7buf[] is found. 2 = Indicates that an extra entry to NMslc7buf[] is found.
<i>Logical Index =</i>	Corresponds to the Error Description as follows: 1 = Index to NMslsc7buf[]. 2 = Index to NMslc7buf[].
<i>Dump =</i>	NA.

LNKUSE

Software Release: 5E10 and later

1. TIME SLOT INTERCHANGE LINK USAGE (LNKUSE) AUDIT

The LNKUSE audit checks the TSI link type (RLLNK_USE) and the peripheral interface data bus (RLPIDB) relations looking for vacant channel data block (RLCHDB) relations or vacant TSDB general data (RLTSDB) relations. The audit runs only in SM-2000 processors.

In the case of digital networking unit - synchronous optical network (SONET) (DNU_S) facilities, the LNKUSE audit checks the virtual peripheral interface data bus (VPIDB_F) relation looking for vacant rCHDB tuples.

2. DATA STRUCTURES

The LNKUSE audit uses seven data structures: RLLNK_USE, RLCHDB, RLVPIDB_F, RLTSDB, RLPIDB, RLLNK_PLI and RLPICB. The RLLNK_USE relation resides only in SM-2000 processors and contains TSI link type information. The two rLLNK_USE fields used in the audit are link [TSI link (key)] and use (link usage).

The RLVPIDB_F relation resides only in SM-2000 processors and is used to map virtual peripheral interface data bus to a DNU_S facility number.

3. AUDIT CHECKS

The LNKUSE audit uses the DRSTATIC driver to traverse rCHDB tuples and network rITSDB tuples.

The audit first checks for the existence of the rCHDB and rITSDB (both network and peripheral) head cells.

For the RLLNK_USE relation, static to dynamic checks are made between the RLLNK_USE and the RLCHDB relation or between the RLLNK_USE and the RLTSDB relation, depending on whether the time slot is a loop time slot or not.

For the RLVPIDB_F relation, static to dynamic checks are made between the RLVPIDB_F and the RLCHDB relation. For each rVPIDB_F buffer, there should be 24 rCHDB tuples representing time slots on a DNU_S facility. The corresponding rPLI_LNK tuple is read to obtain the peripheral link number of the DNU_S facility. The RLPLI_LNK relation maps from the TSI end of the peripheral link to the physical link position on the peripheral unit.

The TSI link and link time slot are converted to the TSI time slot. A check is made to determine if this is a loop time slot. A check for a valid rCHDB tuple is performed for loop time slot links. If the time slot is used for diagnostics, the time slot is skipped.

If the link is connected to a network link interface (NLI) a check is made to determine if there is a valid rITSDB. The TSI Time Slot is converted to a network time slot.

For the peripheral interface data bus relation (RLPIDB) part of the LNKUSE audit, a static to dynamic check from the RLPIDB relation to the RLCHDB relation is made for loop channels. For each rPIDB tuple, all the associated time slots are checked.

ODDLOSTTUP = Check for associated rPICB tuple from rPIDB tuple.

4. ERROR HANDLING

For all vacant rCHDB tuples and rITSDB tuples found and most of the checks mentioned above, an audit error message is printed, the tuple is zeroed out and the zeroed tuple is inserted into the database.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not

found below.

The audit produces the following error code(s).

Table 1 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has successfully read an rPIDB tuple. However, the corresponding rPICB tuple was not found in the database when using the peripheral interface control bus (PICB) attribute (rPIDB.picb) of the RLPIDB relation as a key (rPICB.picb) to the RLPICB relation. This error will continue to print out until fixed by the CRAFT.
Possible Error Effect:	May cause asserts and/or interrupts by various application programs.
Corrective Action Taken:	None.
Manual Action Required:	Either the picb attribute (rPIDB.picb) is not correct and must be updated to the correct value, or a rPICB tuple must be inserted into the database. The key (rPICB.picb) of the rPICB tuple can be obtained from the <i>Bad Data</i> field and the peripheral interface data bus (PICB) number can be obtained from the logical key field of this dump.
Dump Description:	
<i>Error Address</i> =	Address of rPICB tuple.
<i>Bad Data</i> =	rPICB.picb
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	rPICB tuple.

Table 2 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	The two global parameters Glpts_sps and Glnts_sps are not set properly according to the static relation rLNK_USE. This error will continue to print in the ROP until Glpts_sps and Glnts_sps are correctly set.
Possible Error Effect:	The switch may not detect signaling from the user or network due to error in Glpts_sps and Glnts_sps. This is only applied to SM2000.
Corrective Action Taken:	This requires the manual action to correct Glpts_sps and Glnts_sps.
Manual Action Required:	If rLNK_USE is in operational state, the corresponding bit in Glpts_sps(for PLI and DX) and Glnts_sps(for NLI) should be set to 1 for signaling purpose. Please see rLNK_USE population rule for detail about Glpts_sps and Glnts_sps.
Dump Description:	
<i>Error Address</i> =	1 for PLI or DX, 2 for NLI.
<i>Bad Data</i> =	Value of Glpts_sps.
<i>Good Data</i> =	Value of Glnts_sps.
<i>Logical Index</i> =	Logical key of rLNK_USE.
<i>Dump</i> =	Dump of rLNK_USE tuple.

LNSTAT

Software Release: 5E10 and later

1. LINE STATUS (LNSTAT) AUDIT

To process a call, the routing and terminal allocation (RTA) routing process in the switching module (SM), optical remote switching module (ORM), or a remote switching module (RSM) must determine which ports are available for terminating calls. Information about the types of hunted ports is kept on bit maps. A bit map is included in each tuple in the line status relation (RLLNSTAT), which contains information about small and average multi-line hunt groups (MLHGs) (up to 63 lines per group) of hunted lines. The RLLNSTAT relation resides in the SM, but it may also reside in an RSM or ORM. In any SM type, the RLLNSTAT relation contains tuples for all multiline hunt group ports that exist in that SM/ORM/RSM. In the SM, designated as the hunting SM, the rLNSTAT relation contains tuples for all global and local MLHG ports belonging to that hunting SM. During the stand-alone operation, the tuples for other ports that the RSM/ORM contains, are inserted into rLNSTAT; when the RSM/ORM leaves the stand-alone operation, these tuples are deleted.

The line status audit (LNSTAT) checks tuples in the dynamic relation RLLNSTAT for extra tuples. It also performs a check on the status bits for each group. For information about larger groups of hunted lines, refer to the large line status audit (LGLNSTAT). For information about groups of hunted trunks, refer to the trunk status audit (TKSTAT) and the large trunk status audit (LGTKSTAT).

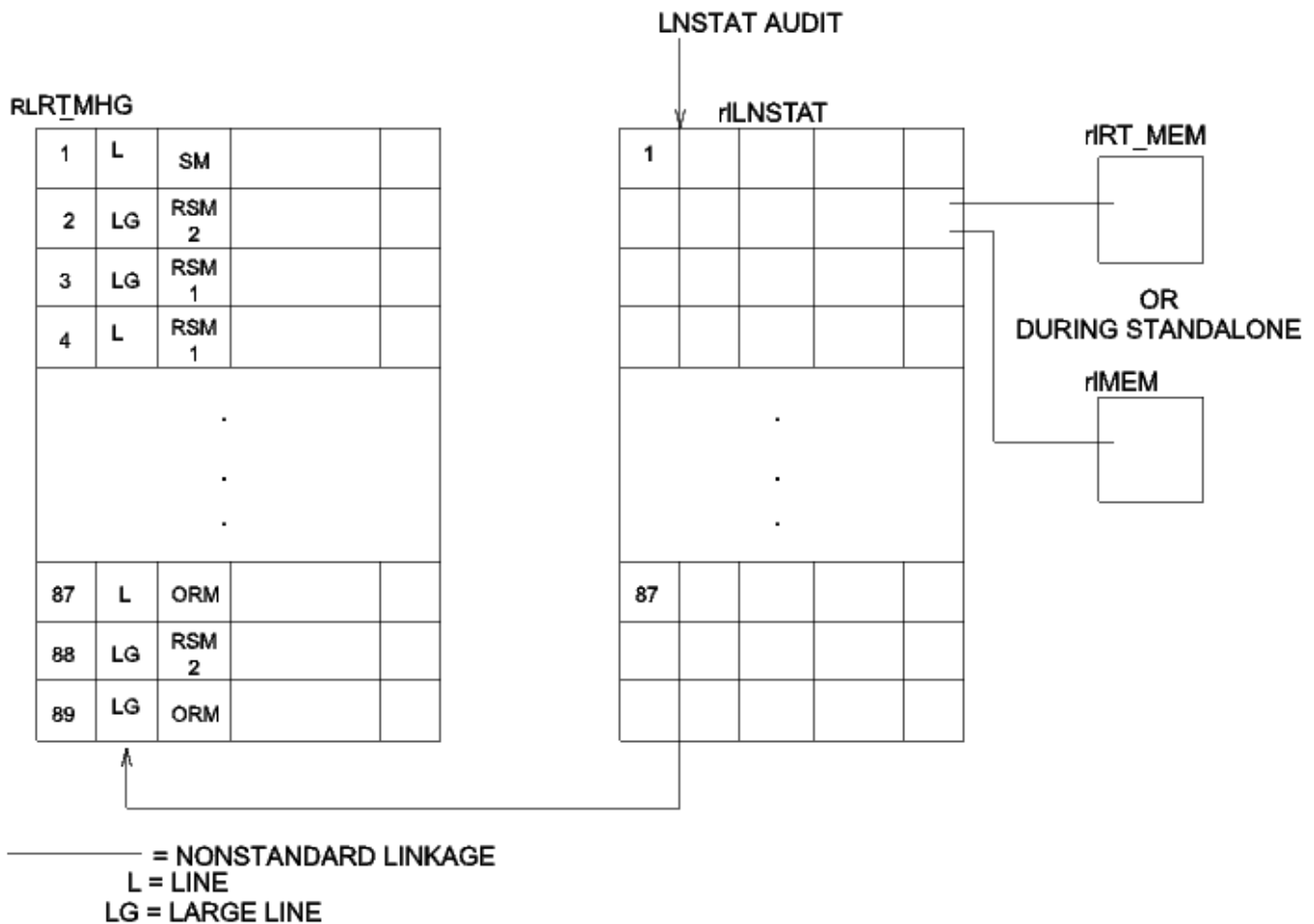
This audit is part of the RTA audits subgroup. In the SM/ORM/RSM modules, the audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

The data structures checked by the LNSTAT audit are the MLHG information relation (RLRT_MHG), the line status relation (RLLNSTAT), the static MLHG member information relation (RLRT_MEM), and the MLHG member status dynamic relation (RLMEM) or RSM/ORM in the standalone operation. Figure 1 illustrates the four relations. The RLRT_MHG relation is a static (write-protected) relation that contains information about all MLHGs. This information includes a key (grpnum), the hunt type, the hunting SM number, and the type of group (local or global) it is. Both RLRT_MEM and RLMEM relations contain information about each MLHG member port equipage. The RLRT_MEM relation resides on the designated hunting module, SMORM/RSM. The RLMEM relation is created only on the RSM/ORM, with standalone capability. The line status audit steps through the RLLNSTAT relation and verifies it against the information contained in the rLRT_MHG and either the rLRT_MEM or rLMEM.

SM & RSM*

AM



*IN AN RSM ENVIRONMENT, THE rLNSTAT RELATION
RESIDES IN BOTH THE HUNTING SM AND THE RSM.

Figure 1 (LNSTAT-1) Line Status Audit

3. AUDIT CHECKS

The LNSTAT audit uses the static driver (DRSTATIC) to step through the rLNSTAT relation and sequentially retrieve tuples. The DRSTATIC driver invokes a semantic check function that performs the following checks:

- (a) The audit verifies each rLNSTAT tuple against the corresponding rIRT_MHG tuple by using the key (group number) of rLNSTAT to read the rIRT_MHG relation. The audit looks specifically for extra rLNSTAT tuples.
- (b) The audit performs a Boolean check between the following bit arrays: port available flag (PORTAVAIL), stop hunt flag (STOPHUNT), make busy flag (MAKEBUSY), and the group member flag (GMFLG). A boolean check uses the Boolean operators AND, NOT, and OR. The check is as follows:

If the PORTAVAIL flag is 1 and not the MAKEBUSY flag or the STOPHUNT flag is 1, then the GMFLAG should be 1.

- (c) The audit checks the hunt type. Only RTH_REGULAR, RTH_UCD, RTH_CIRCULAR, RTH_PSUCD, RTH_ISAT and RTH_PPSMHG are valid.

- (d) The audit ensures that the module number is correct for the tuple.
- (e) The audit verifies that the busy/idle status are busy for MLHG members that are not equipped with ports.

4. ERROR HANDLING

Excess tuples, tuples not in the appropriate module, and tuples with wrong hunt types are deleted from the relation. Boolean checks that pass, are further checked for the group member flag to be 1. If it is not 1, the group member flag is set to 1. Status bit checks that fail for nonequipped members are corrected by the LNSTAT audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	While auditing an rLNSTAT tuple, a DAGETMEM did not find the associate rLGMEM tuple. Since during stand alone RLMEM relation is audited against static RLGROUP_PORT relation, audit should believe dynamic rLGMEM tuple and delete dynamic rLNSTAT tuple.
Possible Error Effect:	Incorrect use of data resources for MLHGs.
Corrective Action Taken:	The extra rLNSTAT tuple deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLMEM tuple.
<i>Bad Data =</i>	The key of the extra rLNSTAT tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The busy/idle status in a tuple of the relation rLNSTAT is inconsistent with the corresponding member equipage tuple bit map. The nonequipped member should be marked busy. If the KEY field of the dump is 1 (SM stand alone) then an rLNSTAT tuple is inconsistent with an rLMEM tuple, or if it is 0 then the rLNSTAT tuple is inconsistent with an rIRT MEM tuple.
Possible Error Effect:	Failure in terminating to an invalidly marked idle members.
Corrective Action Taken:	The audit set the nonequipped members to correct status, busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLNSTAT tuple having the wrong busy/idle status.
<i>Bad Data =</i>	The array containing the members busy/idle status.
<i>Good Data =</i>	The array containing the members equipage information.
<i>Logical Index =</i>	This is not a key. Verification of busy/idle status is with rLMEM tuple if this dump is 1, or else with rIRT MEM tuple if this dump is 0.
<i>Dump =</i>	NA.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	rLNSTAT tuple.

Table 4 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	The group is of the hunt type RTH_PSUCD but the makebusy[] or stophunt[] fields are not zero.
Possible Error Effect:	The group may not be hunted correctly.
Corrective Action Taken:	The fields are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLNSTAT tuple.
<i>Bad Data</i> =	The erroneous makebusy[] value.
<i>Good Data</i> =	The erroneous stophunt[] value.
<i>Logical Index</i> =	grpnum of the tuple.
<i>Dump</i> =	NA.

Table 5 Error Code: GMFLG

Error Code:	GMFLG
Error Description:	The boolean check between the status flag bits (portavail, makebusy, and stophunt) has passed, but a gmflg status bit(s) was found to be zero (not set) when it should be set to 1. There are five possible combinations for this error to occur.
Possible Error Effect:	If this group member is picked by RTA for routing, the erroneous status data will propagate further inconsistencies.
Corrective Action Taken:	The gmflg status bit(s) is set to 1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLNSTAT tuple.
<i>Bad Data</i> =	Erroneous word in the gmflg bit array.
<i>Good Data</i> =	Corrected word in the gmflg bit array.
<i>Logical Index</i> =	Group member of this tuple.
<i>Dump</i> =	NA.

Table 6 Error Code: KEY

Error Code:	KEY
Error Description:	The tuple of the relation rLNSTAT has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLNSTAT tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 7 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit has found an inconsistency between the static relations RLRT_MEM and RLRT_MHG. Current SM is the hunting SM of the MLHG as indicated by the field in rIRT_MHG tuple and therefore the key of this tuple is used to access the MLHG member equipage tuple in RLRT_MEM relation but no matching tuple is found and the audit cannot continue to do checks.
Possible Error Effect:	Audit continue to the next tuple of RLLGLNTAT relation.
Corrective Action Taken:	Audit report an error.
Manual Action Required:	Verify tuples rIRT_MHG and rIRT_MEM in the ODD, using the rIRT_MHG tuple key found in the "KEY" field of the audit report. Corresponding tuple rIRT_MEM

	with the same key should exist on the same SM. Correct any inconsistencies found.
Dump Description:	
<i>Error Address</i> =	The address of the rIRT_MHG tuple.
<i>Bad Data</i> =	The key of rIRT_MEM tuple returned from the database.
<i>Good Data</i> =	The hunt module of MLHG.
<i>Logical Index</i> =	The key of rIRT_MHG tuple used to access rIRT_MEM tuple.
<i>Dump</i> =	The contents of the rIRT_MHG tuple.

Table 8 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the AM, only tuples from global hunt groups should be present. In an RSM, only tuples from local hunt groups on that RSM should be present except during stand-alone operation when tuples from global hunt groups are allowed.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLNSTAT tuple.
<i>Bad Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rLNSTAT tuple.

Table 9 Error Code: RELQKEY

Error Code:	RELQKEY
Error Description:	The tuple of the relation RLMEN has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIMEM tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	This is not a key but the hunt module of the Multiline Hunt Group.
<i>Dump</i> =	NA.

Table 10 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in rLNSTAT (that is, RTH_REGULAR, RTH_CIRCULAR, RTH_UCD, RTH_PRREG, or RTH_PPSMHG) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLNSTAT tuple.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rLNSTAT tuple.

LOGINID

Software Release: 5E10 and later

1. LOGIN IDENTIFIER (LOGINID) AUDIT

The LOGINID audit runs in both the automatic call distributor management information system (ACD/MIS) and the operator services position system (OSPS). Both the ACD/MIS and OSPS features are similar. Both features are used to present queued calls to agents that are logged into the 5ESS[®] switch. In the OSPS feature, the operator would log into the 5ESS[®] switch through a basic services terminal (BST) and handle requests for operator services. In the ACD/MIS feature, the operator would log into the 5ESS[®] switch through an integrated services digital network (ISDN) position having full business and residential customer services (BRCS) features. In both features, calls are routed to available agent positions for processing. In addition, supervisors are able to monitor all agent activities. Each agent is part of a serving team that is capable of handling certain types of calls.

The RLLOGINID relation contains tuples that define both OSPS and ACD/MIS operators in the 5ESS[®] office. The purpose of the LOGINID audit is to ensure the integrity of the dynamic rLOGINID tuples defined in the RLLOGINID relation. The LOGINID audit runs in the communications module processor (CMP) in the operational kernel process (OKP) environment. The audit runs in segmented mode and is part of the OPAMBACD software package.

2. DATA STRUCTURES

Each tuple of the RLLOGINID relation contains information about the agent's ID, the agent's position number, the position type, and the position's associated port if it is an OSPS position. The RLLOGINID relation is an indexed compaction access relation that is keyed by the internal operator ID field, *intopid*. Each tuple in the RLLOGINID relation has a corresponding static tuple in the RLOLOGINID relation. For agent positions currently logged into the 5ESS[®] switch, there will also be a corresponding dynamic rISTQDB tuple representing the serving team assigned to that agent. The data structures used in the LOGINID audit are shown in Figure 1.

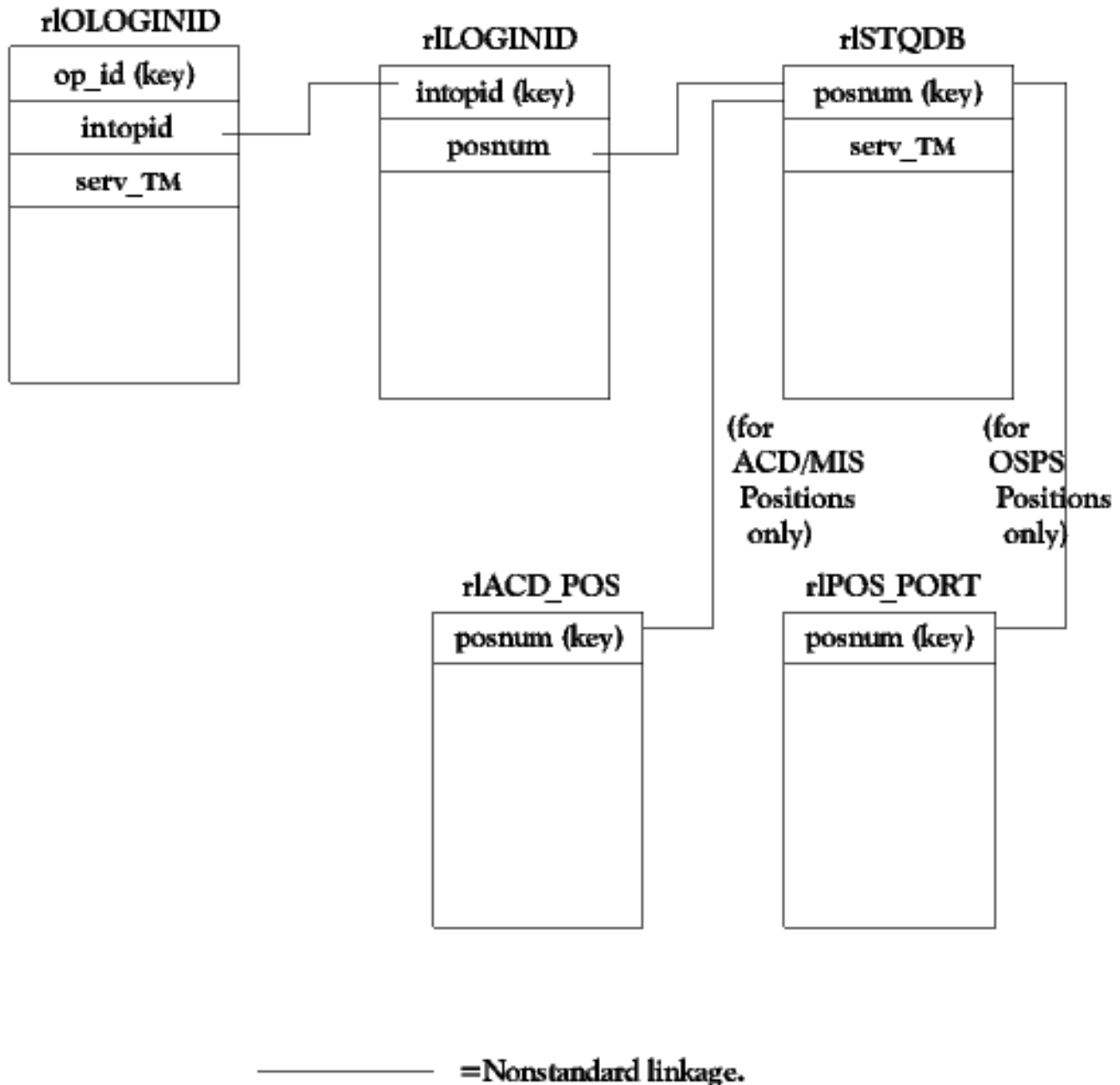


Figure 1 (LOGINID-1) LOGINID Data Structures

3. AUDIT CHECKS

The LOGINID audit uses the DRINDEX audit driver to traverse through each tuple in the RLLOGINID relation. For each tuple, the LOGINID audit will make the following checks: If the LOGINID audit is entered:

- BADTESTPOS** = Check to make sure that an rLOGINID tuple representing an agent position currently logged into the 5ESS® switch has a corresponding rISTQDB tuple.
- LOGBID** = Check to see if the internal operator ID matches the ID contained in the rISTQDB tuple.
- NOTZERO** = Check to make sure that the *gbport* field is zero for an rLOGINID tuple representing an

ACD/MIS agent position currently logged into the 5ESS[®] switch.

- R_PORT** = Check to make sure that the *gbport* field of an rLOGINID tuple representing an OSPS agent position currently logged into the 5ESS[®] switch matches the value defined in the corresponding rISTQDB tuple.
- TYPE** = Check to make sure that the *pos_type* field of an rLOGINID tuple representing an agent position currently logged into the 5ESS[®] switch matches the value defined in the corresponding rISTQDB tuple.
- UNEQUAL** = Check to make sure that the *intopid* field in the rLOGINID tuple matches the value defined in its corresponding static rILOGINID tuple.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BADTESTPOS

Error Code:	BADTESTPOS
Error Description:	The audit was unable to obtain a pointer to the corresponding rISTQDB tuple for an rLOGINID tuple representing an agent position logged into the 5ESS [®] switch.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The rLOGINID tuple is idled to represent an undefined position.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLOGINID tuple.
<i>Bad Data</i> =	The value of the <i>posnum</i> field obtained from the invalid rISTQDB tuple pointer.
<i>Good Data</i> =	The value of the <i>posnum</i> field obtained from the rLOGINID tuple.
<i>Logical Index</i> =	The logical key of the rLOGINID tuple.
<i>Dump</i> =	The rLOGINID tuple.

Table 2 Error Code: LOGBID

Error Code:	LOGBID
Error Description:	The audit found an rLOGINID tuple in which the internal operator ID, <i>intopid</i> does not match the corresponding rISTQDB operator ID field of <i>opr_id</i> .
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The rLOGINID tuple is idled out since there is no valid rISTQDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLOGINID tuple.
<i>Bad Data</i> =	The value of the <i>opr_id</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The value of the <i>posnum</i> field in the rLOGINID tuple.
<i>Logical Index</i> =	The logical key of the rLOGINID tuple.
<i>Dump 1</i> =	The rLOGINID tuple.
<i>Dump 2</i> =	The rISTQDB tuple.

Table 3 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The audit found an rLOGINID tuple that represents an ACD/MIS agent position currently logged into the 5ESS [®] switch whose corresponding <i>gbport</i> field is not zero.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The <i>gbport</i> field in the rLOGINID tuple is set to zero.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLOGINID tuple.
<i>Bad Data</i> =	The value of the <i>gbport.member</i> field in the rLOGINID tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rLOGINID tuple.
<i>Dump</i> =	The rLOGINID tuple.

Table 4 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The audit found an rLOGINID tuple that represents an OSPS agent position currently logged into the 5ESS [®] switch whose corresponding <i>gbport</i> field does not match the value defined in its associated rISTQDB tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The <i>gbport</i> field in the rLOGINID tuple is updated to the value defined in the corresponding rISTQDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLOGINID tuple.
<i>Bad Data</i> =	The value of the <i>gbport.member</i> field in the rLOGINID tuple.
<i>Good Data</i> =	The value of the <i>gbport.member</i> field in the rISTQDB tuple.
<i>Logical Index</i> =	The logical key of the rLOGINID tuple.
<i>Dump 1</i> =	The rLOGINID tuple.
<i>Dump 2</i> =	The rISTQDB tuple.

Table 5 Error Code: TYPE

Error Code:	TYPE
Error Description:	The audit found an rLOGINID tuple that represents an agent position currently logged into the 5ESS [®] switch whose position type field, <i>pos_type</i> , did not match the value defined in the corresponding rISTQDB tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The <i>pos_type</i> field in the rLOGINID tuple is updated to the value defined in the associated rISTQDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLOGINID tuple.
<i>Bad Data</i> =	The value of the <i>pos_type</i> field in the rLOGINID tuple.
<i>Good Data</i> =	The value of the <i>pos_type</i> field in the rISTQDB tuple.
<i>Logical Index</i> =	The logical key of the rLOGINID tuple.
<i>Dump 1</i> =	The rLOGINID tuple.
<i>Dump 2</i> =	The rISTQDB tuple.

Table 6 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The audit found an inconsistency in the values of the internal operator field, <i>intopid</i> , between the dynamic rLOGINID tuple and its corresponding static rOLOGINID tuple.
Possible Error Effect:	Call processing may not function properly for this agent position.
Corrective Action Taken:	The offending rLOGINID tuple is deleted. The related OLOGINID audit will be scheduled to run to reinsert a corrected rLOGINID tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLOGINID tuple.
<i>Bad Data</i> =	The value of the <i>intopid</i> field in the rLOGINID tuple.
<i>Good Data</i> =	The value of the <i>intopid</i> field in the rOLOGINID tuple.
<i>Logical Index</i> =	The logical key of the rLOGINID tuple.
<i>Dump</i> =	The rLOGINID tuple.

LOOPDB

Software Release: 5E10 and later

1. LOOP DATA BLOCK (LOOPDB) AUDIT

Design alternatives for operator actions (OA) data included using the stack or creating a new area or areas for them and linking them from the position data block. The loop data block relation (RLLOOPDB) contains all per call information such as position display and lamp status.

This audit is part of the Operator Services Position System (OSPS) audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode only.

2. DATA STRUCTURES

The data structure checked by the LOOPDB audit is the loop data block relation (RLLOOPDB). Figure 1 illustrates the layout of RLLOOPDB and shows the items checked by this audit. The RLLOOPDB is owned by the position data block (RLPOSDB) relation and the database head cells relation (RLDB_HEADC).

rlLOOPDB

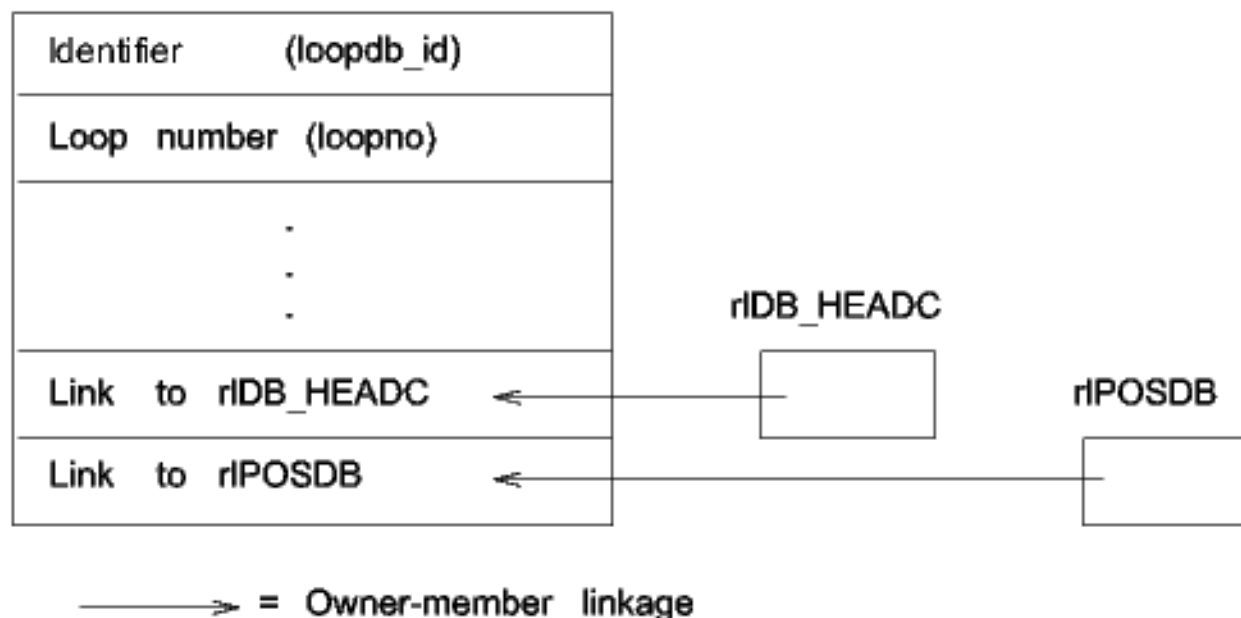


Figure 1 (LOOPDB-1) Data Structures

3. AUDIT CHECKS

This audit verifies the linkages to RLPOSDB and RLDB_HEADC by calling the indirect access sequential key driver (DRINASEK). The DRINASEK driver then invokes the semantic check function to perform the following checks:

- (1) The audit verifies that when the busy/idle state is idle, the tuple is linked to the idle list.
- (2) The audit verifies that when the busy/idle state is busy, the tuple is linked to the busy list.
- (3) The audit verifies that the loop number field (loopno) is within the valid range (0-1).

4. ERROR HANDLING

The audit reports errors when any linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples and links them to the appropriate idle list.

The audit schedules the position data block audit (POSDB) as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rLOOPDB is not busy when it should be.
Possible Error Effect:	The rLOOPDB will be lost to the system.
Corrective Action Taken:	The corresponding rLOOPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLOOPDB.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	rLOOPDB number.
<i>Dump</i> =	rLOOPDB.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rLOOPDB is not idle when it should be.
Possible Error Effect:	The rLOOPDB will be lost to the system.
Corrective Action Taken:	The corresponding rLOOPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLOOPDB.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	rLOOPDB number.
<i>Dump</i> =	rLOOPDB.

Table 3 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The LOOPNO range has a invalid status.
Possible Error Effect:	If LOOPNO is invalid, operators may get indications for calls that are not really there.
Corrective Action Taken:	The corresponding position terminal process will traverse the call and purge it from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLOOPDB.
<i>Bad Data</i> =	LOOPNO value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rLOOPDB.

LPRTACCS

Software Release: 5E10 and later

1. LOCAL PORT TO ACCS DATA BLOCK (LPRTACCS) AUDIT

The local port to the automated calling card service data block audit performs the static to dynamic (lost tuple) checks for the RLACCSDB relation. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

Figure 1 illustrates the basic audit structure. Figure 2 illustrates the layout of the local port to ACCS data block.

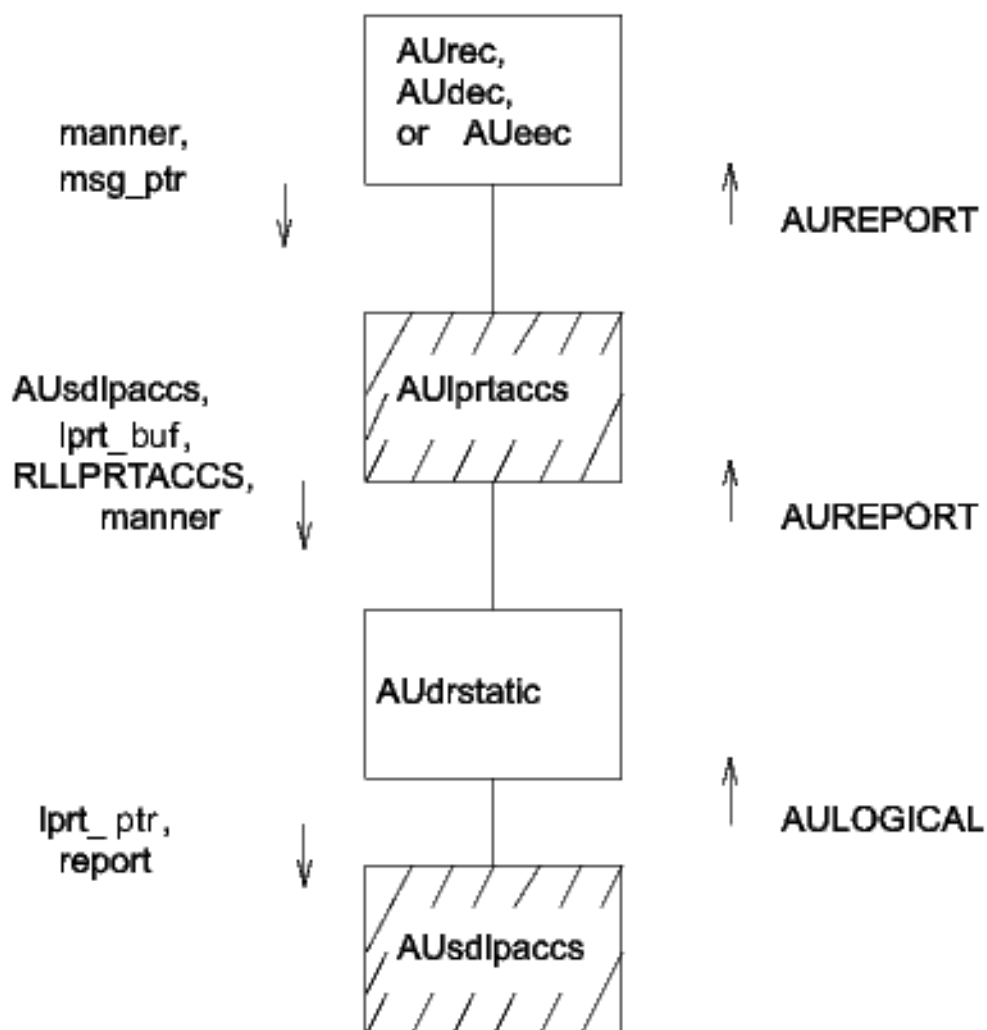


Figure 1 (LPRTACCS-1) Audit Structure Chart

LPRTACCS TUPLE 0

LPORT
ACCSDb_KEY

LPRTACCS TUPLE 1

LPORT
ACCSDb_KEY

·
·
·
·

LPRTACCS TUPLE n

LPORT
ACCSDb_KEY

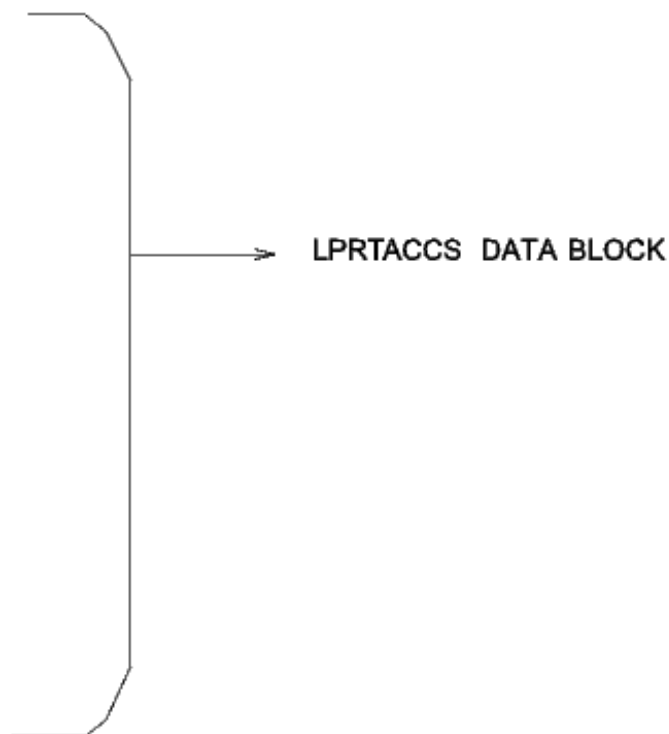


Figure 2 (LPRTACCS-2) Audited Fields in the LPRTACCS Tuple

3. AUDIT CHECKS

The audit driver AUdrstatic steps through the rLPRTACCS tuples, calling the static to dynamic check function for tuples that exist. The static to dynamic function makes an existence check (lost tuple) for the RLACCSDb relation. In addition, it checks that the lport attributes match.

4. ERROR HANDLING

When a rACCSDb tuple doesn't exist, one is created and initialized. When an inconsistency is detected in the lport attributes, the audit reports an error and corrects the inconsistency. Any errors found result in the ACCSDb audit being scheduled as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rACCSDb does not exist for a port when it should.
Possible Error Effect:	This will cause lost resources.
Corrective Action Taken:	The rACCSDb tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rACCSDb.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIACCSDB.
<i>Dump</i> =	NA.

Table 2 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The lport attribute in rIACCSDB does not match the lport in rILPRTACCS.
Possible Error Effect:	This will cause lost resources.
Corrective Action Taken:	The lport is initialized to the static relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	address of rIACCSDB.
<i>Bad Data</i> =	lport in rIACCSDB.
<i>Good Data</i> =	lport in rILPRTACCS.
<i>Logical Index</i> =	logical key of rILPRTACCS.
<i>Dump</i> =	NA.

LPRTPOS

Software Release: 5E10 and later

1. POSITION TO LOCAL PORT MAPPING (LPRTPOS) AUDIT

The terminal maintenance subsystem must know the status of ports to properly maintain resources. The position to local port mapping relation (RLLPRT_POS), a static relation in the SMs, performs existence checks on the position data block (RLPOSDB) relation, and the toll and assistance position data block (RLTAPOSDB) relation.

This audit is part of the Operator Services Position System (OSPS) audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It operates in the OSDS-M environment in the segmented mode only.

2. DATA STRUCTURES

The data structures checked by the LPRTPOS audit are the position-to-local port mapping relation (RLLPRT_POS), the position data block relation (RLPOSDB), and the toll and assistance position data block relation (RLTAPOSDB). Figure 1 shows the relationships between the three relations. The structure of LPRT_POS has been changed to identify the type of position (POSDB or TAPOSDB).

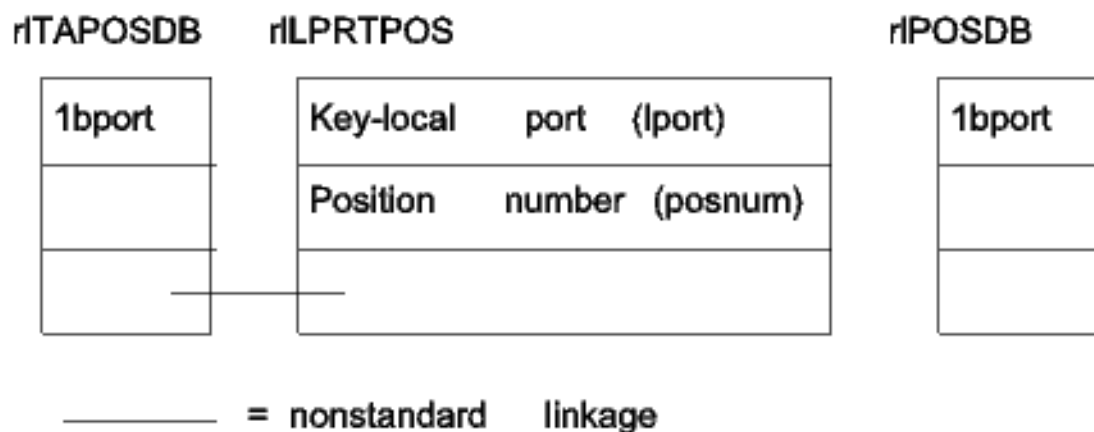


Figure 1 (LPRTPOS-1) Linkages of rLPRTPOS

3. AUDIT CHECKS

To verify the linkages in RLLPRT_POS, the audit calls the static driver (DRSTATIC). The driver calls the static-to-dynamic existence check function to perform the following checks:

- (a) If the position type is a basic services terminal (BST), the audit verifies that a corresponding tuple exists in the dynamic RLPOSDB relation.
- (b) If the position is a BST, the audit verifies the lbport attribute of the rIPOSDB tuple with the lbport attribute of the rLPRT_POS tuple.
- (c) If the position is a video display terminal (VDT), the audit verifies that a corresponding tuple exists in the dynamic RLTAPOSDB relation.
- (d) If the position is a VDT, the audit verifies the lbport attribute of the rTAPOSDB tuple with the lbport attribute of the rLPRT_POS tuple.

The LPRT_POS audit calls the AUdrstatic() driver to step through each rLPRT_POS tuple. The driver calls

AUslprtpos() to perform the static to dynamic checks.

4. ERROR HANDLING

All errors result in output messages. If a rIPOSDB is lost, a tuple will be created and initialized. An inconsistency found in the lbport attributes of either RLPOSDB or RLTAPOSDB will be connected from RLLPRT_POS.

The audit schedules the position data block audit (POSDB) and the toll and assistance position data block audit (TAPOSDB) as related audits.

The audit generates the following error codes and those produced by the DRSTATIC driver. Refer to the driver description for the DRSTATIC error codes.

Upon detection of errors in either the syntactic or semantic data, AUrectup() is called to recover the bad tuple. The recovery is done by AURposdb(). Whenever an error is found in a POSDB relation, the RLPOSDB contents will be printed and the related audits will be scheduled to run.

When a missing RLPOSDB or a RLTAPOSDB tuple is detected, the audit will insert it.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rIPOSDB does not exist for a port when it should.
Possible Error Effect:	This will cause lost resources.
Corrective Action Taken:	The rIPOSDB tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPOSDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIPOSDB.
<i>Dump 1 =</i>	NA.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rITAPOSDB does not exist for a port when it should.
Possible Error Effect:	This will cause lost resources.
Corrective Action Taken:	The rITAPOSDB tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rITAPOSDB.
<i>Dump 1 =</i>	NA.

Table 3 Error Code: PORT_TM

Error Code:	PORT_TM
Error Description:	An rITAPOSDB tuple has a different lbport value than the lbport field in rLLPRT_POS.
Possible Error Effect:	This will cause call information to go to the wrong port.
Corrective Action Taken:	The lbport field in the rITAPOSDB tuple will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB.

<i>Bad Data</i> =	lbport in rIAPOSDB.
<i>Good Data</i> =	lbport in rILPRT_POS.
<i>Logical Index</i> =	lbport in rILPRT_POS.
<i>Dump 1</i> =	NA.

Table 4 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	An rIPOSDB tuple has a different lbport value than the lbport field in rILPRT_POS.
Possible Error Effect:	This will cause call information to go to the wrong port.
Corrective Action Taken:	The lbport field in the rIPOSDB tuple will be corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPOSDB.
<i>Bad Data</i> =	lbport in rIPOSDB.
<i>Good Data</i> =	lbport in rILPRT_POS.
<i>Logical Index</i> =	lbport in rILPRT_POS.
<i>Dump 1</i> =	NA.

21. LQ - MO

LSCMDB

Software Release: 5E10 and later

1. LARGE SHARED CALL MODEL DATA BLOCK (LSCMDB) AUDIT

The LSCMDB audit insures the internal consistency of the RLLSCMDB relation and performs linkage checks between the RLLSCMDB relation and the RLDB_HEADC and RLCLDB relations.

2. DATA STRUCTURES

The RLLSCMDB relation is a dynamic indexed sequential relation that is used to save per call information for all the shared associated terminals in the key system group. When the rLSCMDB tuple is busy it is linked to an rCLDB tuple. All idle rLSCMDB tuples are linked to an idle list that indicates a specified tuple in the RLDB_HEADC relation.

3. AUDIT CHECKS

The LSCMDB audit is invoked through audit control which calls the LSCMDB audit entry function. The function invokes the indexed sequential driver (DRINASEK) to step through all tuples in the RLLSCMDB relation in a sequential manner.

If all the linkage and semantic checks pass then the driver steps to the next tuple and continues another cycle. If any checks fail, recovery is started.

The following is a list of semantic checks that are performed:

- (a) Since RLLSCMDB is only used by large shared key system groups, the rLSCMDB.lrg_grp field must be set to indicate a large group.
- (b) If the rLSCMDB tuple is on the idle link list, then the linkage to the rCLDB tuple should be null.

The following checks are only made if the rLSCMDB tuple is on the busy list:

- (a) The terminal process one field in the rLSCMDB should be associated with a live process.
- (b) If the bridge bit in the rLSCMDB tuple is set to DBNO then the terminal process two array in the rLSCMDB tuple should be null.
- (c) If the bridge bit in the rLSCMDB tuple is set to DBYES then the terminal process two array in the rLSCMDB tuple should contain at least one live process.

4. ERROR HANDLING

When an error is detected either in the semantic check or in the linkage checking routine, a failure return is returned back to the indexed sequential driver. If the rLSCMDB tuple is linked onto the idle list, the rIADB is unlinked from that list. If the rLSCMDB tuple is mutilated in such a way that it cannot be unlinked from the list the indexed sequential driver will patch the list. Recovery is performed on the rLSCMDB tuple that was found in error or any rLSCMDB tuples that were left hanging as a result of a list being patched.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	For an inactive bridge/conference there is a PID of the TP of a bridged party which is non-null.
Possible Error Effect:	The call will have an incorrect path.
Corrective Action Taken:	The tuple is reinitialized and returned to the idle list, purging the call.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLSCMDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The OSPID of the TP of the terminal actively on the call is not an active process for this module.
Possible Error Effect:	The rLSCMDB tuple is connected to a dead process and cannot be used for other calls.
Corrective Action Taken:	Process owning the rLSCMDB will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLSCMDB tuple.
<i>Bad Data</i> =	Process number of tp1 for rLSCMDB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 3 Error Code: FPID_DEAD

Error Code:	FPID_DEAD
Error Description:	The bridge bit is set to DBYES and the process (tp2) is not null, but tp2 is not an active process.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rLSCMDB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLSCMDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key to the rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 4 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	All tuples in RLLSCMDB are used for large key system groups; therefore, all tuples in this relation should reflect this in the "lrg_grp" field. This rLSCMDB tuple contains garbage in its lrg_grp field.
Possible Error Effect:	The rLSCMDB tuple may contain other erroneous data.
Corrective Action Taken:	The audit will idle the rLSCMDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLSCMDB tuple.
<i>Bad Data</i> =	Owner link of rLSCMDB tuple.
<i>Good Data</i> =	0
<i>Logical Index</i> =	Logical key of rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 5 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rLSCMDB tuple is on an invalid list.
Possible Error Effect:	The rLSCMDB tuple is in a limbo condition which is unusable by call

	processing.
Corrective Action Taken:	The audit will idle the rLSCMDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLSCMDB tuple.
<i>Bad Data</i> =	Owner link of rLSCMDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 6 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The TP (tp2) is null when the bridge bit is set to DBYES. The TP should not be null.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rLSCMDB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLSCMDB tuple.
<i>Bad Data</i> =	The process number for tp2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key to the rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 7 Error Code: PID

Error Code:	PID
Error Description:	When the rLSCMDB is busy, the PID should not be null. However, in this case the PID of tp1 is null.
Possible Error Effect:	Incorrect call path.
Corrective Action Taken:	The rLSCMDB tuple will be placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLSCMDB tuple.
<i>Bad Data</i> =	Contents of tp1 field in rLSCMDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

Table 8 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rLSCMDB tuple has a nonzero linkage in one of its link blocks.
Possible Error Effect:	Resource is linked to a rLSCMDB tuple on the idle list and could be unusable by call processing.
Corrective Action Taken:	The rLSCMDB tuple will be placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLSCMDB tuple.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rLSCMDB tuple.
<i>Dump</i> =	The contents of the rLSCMDB tuple.

LSDATA

Software Release: 5E11 and later

1. LINK SET STATUS (LSDATA) AUDIT

LSDATA audit will audit data in the dynamic relation RLLS_STAT against the data in the dynamic relations RLLNK_STAT and RLPORTLA and the static relation RLLS_GRP. The dynamic relation RLPCBLA is used to verify whether a port is owned; and the dynamic relation rLPORTLA is used to find the status of a port. The static relation RLLNKP_LS is used to check the priority of the link. The audit is driven from the RLLS_GRP static relation.

2. DATA STRUCTURES

For details on the fields contained in the RLLS_STAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AA_DATA_BAD

Error Code:	AA_DATA_BAD
Error Description:	Auto_alloc in rLS_STAT not equal to auto_alloc in rLS_GRP.
Possible Error Effect:	Master control center (MCC) pages may display incorrect information.
Corrective Action Taken:	The audit corrects the value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	Auto_alloc in rLS_STAT.
<i>Good Data =</i>	Auto_alloc in rLS_GRP.
<i>Logical Index =</i>	Isnum (key) of rLS_STAT.
<i>Dump =</i>	NA.

Table 2 Error Code: ACC_LSINDEX

Error Code:	ACC_LSINDEX
Error Description:	The rLS_STAT.acclsindx is the message transfer part (MTP) accounting link set index. It does not equal rLS_GRP.Isnum in the tuple whose rLS_GRP.Isnum equals rLS_STAT.Isnum.
Possible Error Effect:	The protocol handler (PH) does not account for this link set. Incorrect billing may result.
Corrective Action Taken:	The audit sets rLS_STAT.acclsindx equal to rLS_GRP.acclsindx.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	rLS_STAT.acclsindx.
<i>Good Data =</i>	rLS_GRP.acclsindx.
<i>Logical Index =</i>	rLS_STAT.Isnum.
<i>Dump =</i>	Contents of rLS_STAT tuple.

Table 3 Error Code: ACT_CNT_BAD

Error Code:	ACT_CNT_BAD
Error Description:	Count of active links in rLS_STAT not equal to count as taken from rPORTLA and rPCBLA.
Possible Error Effect:	If active link is less than verified, another link to replace down link is not activated. If active link is greater than verified, link in link set may be activated unnecessarily.
Corrective Action Taken:	Audit notifies signaling link management (SLM).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	actnum in rLS_STAT.
<i>Good Data =</i>	Count determined from rPORTLA and rPCBLA.
<i>Logical Index =</i>	Key of rLS_STAT.
<i>Dump =</i>	NA.

Table 4 Error Code: APCI_CON_ERR

Error Code:	APCI_CON_ERR
Error Description:	The APCI in dynamic rLS_STAT has been verified against APCI in static rLS_GRP, but when used to read static rAPCI_APC, no match is found.
Possible Error Effect:	The signaling link does not come into service.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the population of these relations.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	apci from rLS_STAT.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	lsnum (key) of rLS_STAT.
<i>Dump =</i>	NA.

Table 5 Error Code: AVAIL_BAD_LS_STAT

Error Code:	AVAIL_BAD_LS_STAT
Error Description:	Availability of link set improperly marked in rLS_STAT.
Possible Error Effect:	If available and marked unavailable, link set might not be carrying its normal traffic. If unavailable and marked available, traffic is lost.
Corrective Action Taken:	Audit corrects availability.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	Availability of link set in rLS_STAT.
<i>Good Data =</i>	Correct availability.
<i>Logical Index =</i>	Key of rLS_STAT.
<i>Dump =</i>	NA.

Table 6 Error Code: BAD2_AVAIL

Error Code:	BAD2_AVAIL
Error Description:	The avail flag in the MCC table CClsmmcstat[] is marked wrong, as determined from the data calculated for the link sets for this protocol.
Possible Error Effect:	The MCC pages may display incorrect information.
Corrective Action Taken:	The MCC table CClsmmcstat[] is corrected to reflect the correct avail flag for this link set and protocol.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CClsmmcstat[] table entry.
<i>Bad Data =</i>	Original avail flag value.
<i>Good Data =</i>	avail flag value of link set as determined by the audit.
<i>Logical Index =</i>	Key of rLS_STAT.
<i>Dump =</i>	NA.

Table 7 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The multistate flag in the MCC table CCIsMccstat[] is marked wrong, as determined from the data calculated for the link sets for this protocol.
Possible Error Effect:	The MCC pages may display incorrect information.
Corrective Action Taken:	The MCC table CCIsMccstat[] is corrected to reflect the correct multistate flag for this link set and protocol.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCIsMccstat[] table entry.
<i>Bad Data =</i>	Original multistate flag value.
<i>Good Data =</i>	Multistate flag value of link set as determined by the audit.
<i>Logical Index =</i>	Key of rILS_STAT.
<i>Dump =</i>	NA.

Table 8 Error Code: BAD_APCI

Error Code:	BAD_APCI
Error Description:	Apci in rILS_STAT not equal to apci in rILS_GRP.
Possible Error Effect:	Change over and change back do not occur properly.
Corrective Action Taken:	Audit corrects value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rILS_STAT tuple.
<i>Bad Data =</i>	apci in rILS_STAT.
<i>Good Data =</i>	apci in rIAPC_INFO.
<i>Logical Index =</i>	Key of rILS_STAT.
<i>Dump =</i>	NA.

Table 9 Error Code: BAD_CONGEST

Error Code:	BAD_CONGEST
Error Description:	Congestion of link set improperly marked in rILS_STAT.
Possible Error Effect:	Traffic is not routed over the link set.
Corrective Action Taken:	Audit corrects congestion status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rILS_STAT tuple.
<i>Bad Data =</i>	Congestion of link set in rILS_STAT.
<i>Good Data =</i>	Correct congestion.
<i>Logical Index =</i>	Key of rILS_STAT.
<i>Dump =</i>	NA.

Table 10 Error Code: BAD_PHYS_CNT

Error Code:	BAD_PHYS_CNT
Error Description:	Count of physical links in rILS_STAT not equal to nlinks in rILS_GRP.
Possible Error Effect:	Incorrect number of active links in link set.
Corrective Action Taken:	Send signaling link management (SLM) a message to activate or deactivate links to increase or decrease, respectively, the number of links in the link set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rILS_STAT tuple.
<i>Bad Data =</i>	Count of physical links available as read from rILS_STAT.
<i>Good Data =</i>	nlinks in rILS_GRP.
<i>Logical Index =</i>	Key of rILS_STAT.
<i>Dump =</i>	Tuple in error from rILS_STAT.

Table 11 Error Code: CONGESTION

Error Code:	CONGESTION
Error Description:	The congestion level for a link set stored in CCIs_info array does not match the congestion level for the same link set stored in RLLS_STAT.
Possible Error Effect:	Actions in the PH might be inconsistent with local congestion status, that is, transfer controlled messages might be sent when they are not needed, and vice versa.

Corrective Action Taken:	The congestion level in the array is set to the same as the congestion level in the relation for the link set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLS_STAT tuple.
<i>Bad Data</i> =	Congestion level stored in CCls_info[rLS_STAT.lsnun].
<i>Good Data</i> =	rLS_STAT.cg.
<i>Logical Index</i> =	rLS_STAT.lsnun.
<i>Dump</i> =	NA.

Table 12 Error Code: INT_DATA_BAD

Error Code:	INT_DATA_BAD
Error Description:	Internal in rLS_STAT is not equal to internal in rLS_GRP.
Possible Error Effect:	The signaling link does not come into service.
Corrective Action Taken:	The audit corrects the value in dynamic rLS_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLS_STAT tuple.
<i>Bad Data</i> =	Internal in rLS_STAT.
<i>Good Data</i> =	Internal in rLS_GRP.
<i>Logical Index</i> =	lsnum (key) of rLS_STAT.
<i>Dump</i> =	NA.

Table 13 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The link set state in the MCC table CCls_mccstat[] is marked wrong, as determined from the data calculated for the link sets for this protocol.
Possible Error Effect:	The MCC display may show the incorrect state of the link set.
Corrective Action Taken:	The MCC table CCls_mccstat[] is corrected to the correct state for this link set and protocol.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of CCls_mccstat[] table entry.
<i>Bad Data</i> =	Original link set state value.
<i>Good Data</i> =	State of link set as determined by the audit.
<i>Logical Index</i> =	Key of rLS_STAT.
<i>Dump</i> =	NA.

Table 14 Error Code: LNKS_MISSING

Error Code:	LNKS_MISSING
Error Description:	The CCls_mccstat[] table was missing a valid link set for this common channel signaling (CCS) protocol that exists in static rLS_GRP.
Possible Error Effect:	The MCC table would not be accurate; link set display would be erroneous.
Corrective Action Taken:	The audit inserts the missing link set into the MCC table in the appropriate position for this protocol. The MCC pages are be updated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of static rLS_GRP.
<i>Bad Data</i> =	-1 failure return form CCfindls.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rLS_GRP.
<i>Dump</i> =	NA.

Table 15 Error Code: LS_STAT_NO

Error Code:	LS_STAT_NO
Error Description:	Tuple in static data does not exist in dynamic rLS_STAT.
Possible Error Effect:	Status of links in missing link set tuple cannot be changed. Change over and change back do not occur properly. Link cannot be inhibited or uninhibited. Other link status changes do not occur properly.
Corrective Action Taken:	Audit inserts tuple in rLS_STAT.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rILS_GRP tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Link set number in rILS_GRP.
<i>Logical Index</i> =	Key of rILS_GRP.
<i>Dump</i> =	NA.

Table 16 Error Code: LSNUM_BAD

Error Code:	LSNUM_BAD
Error Description:	The CCacclsidx_Is[] holds accounting link set information for message transfer part (MTP). An CCacclsidx_Is[] entry whose index equals rILS_GRP.acclsidx does not equal the key of the rILS_GRP tuple (rILS_GRP.Isnum).
Possible Error Effect:	Incorrect translation occur while attempting to populate the counts message sent to the administration module (AM). All counts are lost for that index value due to the incorrect translations.
Corrective Action Taken:	The audit sets CCacclsidx_Is[rILS_GRP.acclsidx] equal to rILS_GRP.Isnum.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CCacclsidx_Is[rILS_GRP.acclsidx].
<i>Good Data</i> =	rILS_GRP.Isnum.
<i>Logical Index</i> =	rILS_STAT.acclsidx.
<i>Dump</i> =	NA.

Table 17 Error Code: NO2_LNKP_LS

Error Code:	NO2_LNKP_LS
Error Description:	Port derived from dltport of rILNK_STAT not found in static relation RLLNKP_LS. This error results from trying to check the priority of a stand-by port.
Possible Error Effect:	Status of port cannot be verified. Link cannot be activated or deactivated.
Corrective Action Taken:	None.
Manual Action Required:	Investigate office-dependent data (ODD) relation RLLNKP_LS.
Dump Description:	
<i>Error Address</i> =	Address of rILNK_STAT tuple.
<i>Bad Data</i> =	Port from rIORTLA.
<i>Good Data</i> =	dltport in rILNK_STAT.
<i>Logical Index</i> =	Key of rILNK_STAT.
<i>Dump</i> =	NA.

Table 18 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	An entry in CCacclsidx_Is[] is not in use because its index does not equal rIDX_DPCS.acclsidx from any rILS_GRP tuple. The entry should be set to its null value (0) but is not.
Possible Error Effect:	Incorrect translation occur while attempting to populate the counts message sent to the AM. All counts are lost for that index value due to the incorrect translations.
Corrective Action Taken:	The audit sets the CCacclsidx_Is[] entry to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CCacclsidx_Is[index].
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Index.
<i>Dump</i> =	NA.

Table 19 Error Code: NOT_ENOUGH_ACT_LINKS

Error Code:	NOT_ENOUGH_ACT_LINKS
Error Description:	The number of active links less than the number which should be active.
Possible Error Effect:	Not enough links active per requirements.
Corrective Action Taken:	Audit notifies signaling link management (SLM) to activate a link.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rILS_GRP tuple.
<i>Bad Data</i> =	alnks in rILS_GRP.
<i>Good Data</i> =	Active link count determined from rIPCBLA, rIORTLA, and rILNKP_LS.
<i>Logical Index</i> =	Key of rILS_GRP.
<i>Dump</i> =	NA.

Table 20 Error Code: NO_LNK_STAT

Error Code:	NO_LNK_STAT
Error Description:	Link index in rILS_STAT lnkiset array not found in rILNK_STAT.
Possible Error Effect:	Status of link cannot be changed.
Corrective Action Taken:	Related audit inserts tuple into RLLNK_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Link index in lnkiset in rILS_STAT.
<i>Logical Index</i> =	Key of rILS_STAT.
<i>Dump</i> =	Tuple in error from rILS_STAT.

Table 21 Error Code: NO_PORTLA

Error Code:	NO_PORTLA
Error Description:	Port derived from dltport of rILNK_STAT not found in dynamic relation RLPORTLA.
Possible Error Effect:	Status of port cannot be verified. Link cannot be activated.
Corrective Action Taken:	Related audit corrects rIORTLA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILNK_STAT tuple.
<i>Bad Data</i> =	dltport in rILNK_STAT.
<i>Good Data</i> =	Port derived from dltport.
<i>Logical Index</i> =	Key of rILNK_STAT.
<i>Dump</i> =	NA.

Table 22 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The network and the adjacent point code (APC) have been verified against the global parameter and rIAPCI_APC, respectively, but when used to read static rIGSM_OPC, no match was found.
Possible Error Effect:	The signaling link does not come into service.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the population of these relations and the global parameter for this switching module (SM).
Dump Description:	
<i>Error Address</i> =	Address of rILS_STAT tuple.
<i>Bad Data</i> =	Network for the APC.
<i>Good Data</i> =	APC used in read.
<i>Logical Index</i> =	lsum (key) of rILS_STAT.
<i>Dump</i> =	NA.

Table 23 Error Code: SELECT_BIT_BAD

Error Code:	SELECT_BIT_BAD
Error Description:	The link selection bit in the dynamic rILS_STAT does not match the link selection bit in the static rILS_GRP for a link set number.
Possible Error Effect:	The sls distribution in the link set is incorrect. The sls distribution in the combined link sets having this link set as a member also is incorrect.
Corrective Action Taken:	The audit corrects the dynamic data to match the static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rILS_STAT tuple.
<i>Bad Data</i> =	lnk_sel from rILS_STAT tuple.
<i>Good Data</i> =	lnk_sel from rILS_GRP tuple.

<i>Logical Index</i> =	Isnum (key) from RLIs_stat tuple.
<i>Dump</i> =	NA.

Table 24 Error Code: SMLCG_INC

Error Code:	SMLCG_INC
Error Description:	The global counter for the number of ITU-T signaling system number 7 link sets in congestion (NMsmc7lcg) is a negative value.
Possible Error Effect:	The NMCCS_CONG discrete may be set incorrectly.
Corrective Action Taken:	The LSLINK audit corrects this problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NMsmc7lcg.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 25 Error Code: TOO_MANY_ACT_LINKS

Error Code:	TOO_MANY_ACT_LINKS
Error Description:	The number of active links exceeds the number which should be active.
Possible Error Effect:	Too many links active per requirements.
Corrective Action Taken:	Audit notifies SLM to deactivate a link.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILS_GRP tuple.
<i>Bad Data</i> =	alnks in rILS_GRP.
<i>Good Data</i> =	Active link count determined from rIPCBLA, rIORTLA, and rILNKP_LS.
<i>Logical Index</i> =	Key of rILS_GRP.
<i>Dump</i> =	NA.

Table 26 Error Code: TO_GSM_DATA_BAD

Error Code:	TO_GSM_DATA_BAD
Error Description:	to_gsm in rILS_STAT is not equal to gsm in rIGSM_OPC.
Possible Error Effect:	Signaling link does not come into service.
Corrective Action Taken:	Audit corrects the value in dynamic RLIs_stat.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rILS_STAT tuple.
<i>Bad Data</i> =	to_gsm rILS_STAT.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Isnum (key) of rILS_STAT.
<i>Dump</i> =	NA.

Table 27 Error Code: TYPE1

Error Code:	TYPE1
Error Description:	The audit cannot find a network type for this SM.
Possible Error Effect:	The signaling link does not come into service.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the population of global parameters for networks.
Dump Description:	
<i>Error Address</i> =	Address of rILS_STAT tuple.
<i>Bad Data</i> =	SM number for which there is no network type.
<i>Good Data</i> =	Adjacent point code (APC).
<i>Logical Index</i> =	Isnum (key) of rILS_STAT.
<i>Dump</i> =	NA.

LSLINK

Software Release: 5E11 and later

1. LINK SET LINKAGE DATA (LSLINK) AUDIT

LSLINK audit step through the dynamic relation RLLS_STAT (link set status). For each tuple in the RLLS_STAT relation, the audit checks the linkage with the RLDPC_STAT (destination point code status) relation. It also checks the semantic data in the RLLS_STAT relation against RLDPC_STAT and RLLS_GRP (link set group data) relations.

2. DATA STRUCTURES

For details on the fields contained in the RLLS_STAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	The link set is marked unavailable in rLS_STAT, but owns dpcis.
Possible Error Effect:	Traffic toward destinations is be lost.
Corrective Action Taken:	Audit sends signaling traffic and route management (STRM) a message to do route correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS_STAT tuple.
<i>Bad Data =</i>	Incorrect head dpci of rLS_STAT tuple.
<i>Good Data =</i>	Incorrect tail dpci of rLS_STAT tuple.
<i>Logical Index =</i>	Key of rLS_STAT tuple.
<i>Dump =</i>	NA.

Table 2 Error Code: SMLCG_INC

Error Code:	SMLCG_INC
Error Description:	The NMsmc7lcg counter does not contain the correct number of congested link sets.
Possible Error Effect:	The NMCCS_CONG discrete could be set incorrectly.
Corrective Action Taken:	The audit sets NMsmc7lcg to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Value of NMsmc7lcg.
<i>Good Data =</i>	Correct number of congested link sets.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 3 Error Code: TUP_NOT_ZERO

Error Code:	TUP_NOT_ZERO

Error Description:	All fields except the key field in an unused tuple are not 0.
Possible Error Effect:	Incorrect data may be accessed.
Corrective Action Taken:	All fields except the key field are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rILS_STAT tuple.
<i>Dump</i> =	Dummy tuple of rILS_STAT.

Table 4 Error Code: ZERO_TUP_BAD

Error Code:	ZERO_TUP_BAD
Error Description:	All fields in the tuple with the key of 0, except the key field and the ls_dpc field, are not 0.
Possible Error Effect:	Incorrect data may be accessed.
Corrective Action Taken:	All fields except the key field and ls_dpc field are set to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rILS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rILS_STAT tuple.
<i>Dump</i> =	Zero tuple of rILS_STAT.

LSTAB

Software Release: 5E11 and later

1. LINK SET ROUTING DATA TABLE (LSTAB) AUDIT

LSTAB audit checks switching module processor (SMP) copy of the protocol handler (PH) dynamic data array CCsmstph_ht[][CCNO_LINKS] against data in the dynamic relations RLLNK_STAT and RLLS_STAT. The audit is driven from RLLS_STAT.

2. DATA STRUCTURES

For details on the fields contained in the RLLNK_STAT and RLLS_STAT relations, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CON_AVAIL_BAD

Error Code:	CON_AVAIL_BAD
Error Description:	The avail in rLNK_STAT for the linkindex does not match avail for the lsnm for the same linkindex in rLS_STAT.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The related audits LNKDATA, LNKSLs, and LSData run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rLNK_STAT tuple.
<i>Bad Data</i> =	Avail in rLNK_STAT.
<i>Good Data</i> =	Avail in rLS_STAT.
<i>Logical Index</i> =	Key of rLNK_STAT tuple.
<i>Dump</i> =	rLS_STAT tuple.

Table 2 Error Code: NO_LNK_STAT

Error Code:	NO_LNK_STAT
Error Description:	The link index from the llnkset array in rLS_STAT does not exist in the dynamic relation rLNK_STAT.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The related audits LNKDATA, LNKSLs, and LSData run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rLS_STAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	linkindex from llnkset array in rLS_STAT.
<i>Logical Index</i> =	Key of rLS_STAT tuple.
<i>Dump</i> =	rLS_STAT tuple.

Table 3 Error Code: SEG_ERROR

Error Code:	SEG_ERROR

Error Description:	There were errors in a segment of the SMP copy of PH dynamic data array.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit requests that the corrected segment be transmitted to all PHs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of PH dynamic data array.
<i>Bad Data</i> =	Number of bad tuples in this segment of the SMP copy.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: SLC_AVAIL_BAD

Error Code:	SLC_AVAIL_BAD
Error Description:	The avail for the slc of the lnum in the SMP copy of the PH dynamic data array does not equal avail from rLNK_STAT for the same lnum.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit calls to correct SMP copy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of the PH dynamic data array.
<i>Bad Data</i> =	Avail in SMP copy.
<i>Good Data</i> =	lnum from rLS_STAT (first index into array).
<i>Logical Index</i> =	slc from rLNK_STAT (second index into array).
<i>Dump</i> =	rLNK_STAT tuple.

Table 5 Error Code: SLC_CHNUM_BAD

Error Code:	SLC_CHNUM_BAD
Error Description:	The chnum for the slc of the lnum in the SMP copy of the PH dynamic data array does not equal chnum from rLNK_STAT for the same lnum.
Possible Error Effect:	It is impossible to identify the correct PH channel corresponding to this link.
Corrective Action Taken:	The audit calls to correct the SMP copy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of the PH dynamic data array.
<i>Bad Data</i> =	chnum in SMP copy.
<i>Good Data</i> =	lnum from rLS_STAT (first index into array).
<i>Logical Index</i> =	slc from rLNK_STAT (second index into array).
<i>Dump</i> =	rLNK_STAT tuple.

Table 6 Error Code: SLC_STPH_BAD

Error Code:	SLC_STPH_BAD
Error Description:	The stph for the slc of the lnum in the SMP copy of the PH dynamic data array does not equal stph from rLNK_STAT for the same lnum.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit calls to correct SMP copy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of the PH dynamic data array.
<i>Bad Data</i> =	stph in SMP copy.
<i>Good Data</i> =	lnum from rLS_STAT (first index into array).
<i>Logical Index</i> =	slc from rLNK_STAT (second index into array).
<i>Dump</i> =	rLNK_STAT tuple.

Table 7 Error Code: SLS_CHNUM_BAD

Error Code:	SLS_CHNUM_BAD
Error Description:	The chnum for the sls of the lnum in the SMP copy of the PH dynamic data array does not equal chnum from rLNK_STAT for the same lnum.
Possible Error Effect:	It is impossible to identify the correct PH channel corresponding to this link.
Corrective Action Taken:	The audit calls to correct the SMP copy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of the PH dynamic data array.

<i>Bad Data</i> =	chnum in SMP copy.
<i>Good Data</i> =	lsum from rLS_STAT (first_index into array).
<i>Logical Index</i> =	Position of sls bit set to 1 (second_index into array).
<i>Dump</i> =	rLNK_STAT tuple.

Table 8 Error Code: SLS_STPH_BAD

Error Code:	SLS_STPH_BAD
Error Description:	The stph for the sls of the lsum in the SMP copy of the PH dynamic data array does not equal stph from rLNK_STAT for the same lsum.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit calls to correct SMP copy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of the PH dynamic data array.
<i>Bad Data</i> =	stph in SMP copy.
<i>Good Data</i> =	lsum from rLS_STAT (first_index into array).
<i>Logical Index</i> =	Position of sls bit set to 1 (second_index into array).
<i>Dump</i> =	rLNK_STAT tuple.

Table 9 Error Code: TRANS_FAILED

Error Code:	TRANS_FAILED
Error Description:	The corrected segment of the SMP copy of the PH dynamic data array was not successfully transmitted to all PHs.
Possible Error Effect:	Unknown.
Corrective Action Taken:	When audits LSTAB, DPCTAB and TABDPC complete, audit LSTAB asks for a checksum to be done by the PHs. If the checksums do not match, the PHs ask for a refresh of their dynamic data array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMP copy of PH dynamic data array.
<i>Bad Data</i> =	Return value of transmission.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

LTMSTAT

Software Release: 5E10 and later

1. LOGICAL TERMINATION STATUS (LTMSTAT) AUDIT

The flexible call offering feature allows each direct number to have up to 16 call appearances. The busy/idle status of call appearances for custom and standard electronic key telephone sets (EKTS) are maintained by LTMSTAT tuple. The busy/idle bits are constantly updated by call processing. The main purpose of this audit is to clear out the sticking busy status bits so that call appearance can be reused.

2. DATA STRUCTURES

The data structure checked by the LTMSTAT audit is the logical termination status relation (RLLTMSTAT). Each LTMSTAT tuple contains the key (logical termination index) and 16 subaddress bits to indicate the busy/idle status of 16 call appearances of particular direct number.

3. AUDIT CHECKS

The LTMSTAT audit is part of ISDNBRCS package. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes. The LTMSTAT audit calls DRINASEK driver to step through the tuple and validate the key of the tuple. The following semantic checks are performed by this audit:

- (1) Perform dynamic-to-static existence check between LTMSTAT and IS_LINE relation. Every LTMSTAT tuple should have the corresponding IS_LINE tuple populated.
- (2) Verify the 16 busy/idle status bits contained at LTMSTAT are consistent with those indicated by subaddress bits of BRCSDB (BRCS data block used by ISDN BRCS call processing).
- (3) If the LTMSTAT tuple represents the Direct Number (DN) associated with analog shared DN key system group, the PORTLA and ADB (analog data blocks) are used for determining the status of LTMSTAT.

4. ERROR HANDLING

If any inconsistency is encountered by the LTMSTAT audit, the busy/idle status bits will be updated to reflect the current status of call appearance.

Key	16 subaddress bits
	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Logical Termination Index:

- 0 = idle
- 1 = busy

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	LTMSTAT audit expects the TEDB should have ISDN BRCS feature, but brcs_flag in the TEDB is not set.
Possible Error Effect:	This may cause application codes to reference wrong set of data or features.
Corrective Action Taken:	TEDB audit is requested as related audit to verify brcs_flag.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DBNO (brcs_flag is not set in the TEDB).
<i>Good Data =</i>	DBYES (brcs_flag should be set in the TEDB).
<i>Logical Index =</i>	LTMSTAT key (ltm_index).
<i>Dump =</i>	rLTMSTAT tuple.

Table 2 Error Code: BRCSLINK

Error Code:	BRCSLINK
Error Description:	The rIBRCSDB tuple linked to rIDALB tuple has key error.
Possible Error Effect:	The user may not be able to use the BRCS data subscribed.
Corrective Action Taken:	As a recovery action, related audits will run and recover the data back to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Key of rLTMSTAT tuple.
<i>Good Data =</i>	Key of rIBRCSDB tuple.
<i>Logical Index =</i>	Address of rLTMSTAT tuple.
<i>Dump =</i>	Logical key in rLTMSTAT state. rLTMSTAT tuple.

Table 3 Error Code: BU_STAT

Error Code:	BU_STAT
Error Description:	Inconsistency was found between the busy/idle call appearance status in the LTMSTAT tuple and in the BRCSDB tuples related to it.
Possible Error Effect:	Calls may not be able to be established on certain call appearances, and may appear to exist when they do not. Calls that are established may be lost.
Corrective Action Taken:	LTMSTAT bu_stat attribute is overwritten with correct call appearance status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The customer state (DMCSTATE) of the associated process is dumped instead of a tuple address. The customer state corresponds to the custat field in a related rIPCBLA tuple.
<i>Bad Data =</i>	LTMSTAT bu_stat attribute value.
<i>Good Data =</i>	Correct value for bu_stat attribute.
<i>Logical Index =</i>	LTMSTAT key.
<i>Dump 1 =</i>	rLTMSTAT tuple.
<i>Dump 2 =</i>	rIDALB tuple.
<i>Dump 3 =</i>	rIBRCSDB tuple.
<i>Dump 4 =</i>	rTEDB tuple.
<i>Dump 5 =</i>	rIDPB tuple.

Table 4 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	A TEDB was found without a DALB linked to it, but the LTMSTAT has call appearances marked as being busy.
Possible Error Effect:	Calls may be invalid set up, or may not be set up correctly, or may be torn down. Communication to the customer premise equipment associated with this TEDB is questionable.
Corrective Action Taken:	Idle out all call appearance bits in the bu_stat attribute of the LTMSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	LTMSTAT bu_stat attribute.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LTMSTAT key.
<i>Dump 1</i> =	rLTMSTAT tuple.
<i>Dump 2</i> =	rDPB tuple.
<i>Dump 3</i> =	rTEDB tuple.

Table 5 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The key of DPB (D-channel port) is not the same as that indicated at RLLPT_PT relation.
Possible Error Effect:	CPE may not be able to communicate with switch through D-channel.
Corrective Action Taken:	This may be DPB is not initialized or key error in DPB. If error continues, then it may be an ODD error in FCPORT_TYPE which does not indicate analog shared DN feature.
Manual Action Required:	If it is an ODD problem, the manual action is required to set up analog shared DN bit in FCPORT_TYPE.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Port in DPB.
<i>Good Data</i> =	Port name in LPT_PT.
<i>Logical Index</i> =	LTMSTAT key (ltm_index).
<i>Dump 1</i> =	rLTMSTAT tuple.
<i>Dump 2</i> =	rDPB tuple.

Table 6 Error Code: KEY

Error Code:	KEY
Error Description:	The key value of PCBLA (pcbla_ptr→procnum) is not equal to the llnk field of dalb_lnk for the DALB (dalb_ptr→dalb_lnk.llnk-DBSFPCBLA) associated with the rLTMSTAT tuple being audited.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Related audits are scheduled from the recovery function to correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the dalb_ptr.
<i>Bad Data</i> =	The procnum field of rPCBLA tuple (pcbla_ptr→procnum).
<i>Good Data</i> =	The correct key value of rdalb tuple. (dalb_ptr→dalb_lnk.llnk-DBSFPCBLA).
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	rDALB tuple.
<i>Dump 2</i> =	rPCBLA tuple.
<i>Dump 3</i> =	rLTMSTAT tuple.

Table 7 Error Code: LTMPORT

Error Code:	LTMPORT
Error Description:	No IS_LINE tuple exists for this LTMSTAT tuple, but the tuple is NOT idle.
Possible Error Effect:	Calls may be invalid set up, or may not be set up correctly, or may be torn down. Communication to the customer premise equipment associated with this LTMSTAT is questionable.
Corrective Action Taken:	Idle out all call appearance bits in the bu_stat attribute of the LTMSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	LTMSTAT bu_stat attribute.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LTMSTAT key.
<i>Dump</i> =	rLTMSTAT tuple.

Table 8 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	A rIDSLEQUIP or rIS_TERM tuple is missing in the ODD.
Possible Error Effect:	Not able to determine if this is a Custom or Standard CPE.
Corrective Action Taken:	None.

Manual Action Required:	Need to insert the missing rIDSLEQUIP or rLIS_TERM tuple. - Insert a rIDSLEQUIP tuple using the port field from RLLPT_PT as the key. - Insert a rLIS_TERM tuple using the lead_lport and lead_termno fields from RLIS_LINE as the key.
Dump Description:	
<i>Error Address</i> =	Address of the rLTMSTAT tuple.
<i>Bad Data</i> =	Relation Identifier of either RIDSLEQUIP or RLIS_TERM.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rLTMSTAT tuple.
<i>Dump 1</i> =	Contents of the rLTMSTAT tuple.
<i>Dump 2</i> =	Contents of the rLIS_LINE tuple if Bad Data is referring to RLIS_TERM.

Table 9 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A DPB did not have a TEDB linked to it, when it should have.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	Schedule DPB audit to link a TEDB to the DBP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DPB owner linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LTMSTAT key.
<i>Dump 1</i> =	rLTMSTAT tuple.
<i>Dump 2</i> =	rDPB tuple.

Table 10 Error Code: PORT

Error Code:	PORT
Error Description:	An inconsistency was found for the port name in the LTMSTAT tuple. An attempt to convert the port name from logical port to physical port was unsuccessful.
Possible Error Effect:	Attempts to make calls on call appearances related to this tuple may not be successful. Existing calls may be torn down.
Corrective Action Taken:	LTMSTAT call appearance bits are idled.
Manual Action Required:	The logical port name was gotten from the static RLIS_LINE relation. The read of LPT_PT to retrieve the physical port name failed. The LPT_PT relation should be investigated and corrected.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The lead_lport value in the RLIS_LINE relation is derived from LTMSTAT tuple index.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LTMSTAT key.
<i>Dump</i> =	rLTMSTAT tuple.

Table 11 Error Code: STATE_IDLE

Error Code:	STATE_IDLE
Error Description:	The LTMSTAT for analog shared key system group should be idle but it is being marked as busy.
Possible Error Effect:	This error will block user from originating or terminating to this line.
Corrective Action Taken:	Idle out all call appearance bits in the bu_stat attribute of the LTMSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	0x8000.
<i>Good Data</i> =	Zero.
<i>Logical Index</i> =	LTMSTAT key.

<i>Dump 1 =</i>	rLTMSTAT.
<i>Dump 2 =</i>	rPORTLA for lead port.

Table 12 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The LTMSTAT for analog shared key system group should be busy but it is not being marked as busy.
Possible Error Effect:	User may have another call which is not consistent with the current shared key system group.
Corrective Action Taken:	Busy out all call appearance bits in the bu_stat attribute of the LTMSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Zero.
<i>Good Data =</i>	0x8000 (busy subaddress).
<i>Logical Index =</i>	LTMSTAT key.
<i>Dump 1 =</i>	rLTMSTAT.
<i>Dump 2 =</i>	rPORTLA for lead port.

Table 13 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	LTMSTAT audit expects the TEDB should have right key, but key in the TEDB is error or TEDB is not initialized.
Possible Error Effect:	This may cause application codes to reference wrong set of data or features in TEDB.
Corrective Action Taken:	TEDB audit is requested as related audit to verify TEDB key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Key of TEDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	LTMSTAT key (ltm_index).
<i>Dump =</i>	The rLTMSTAT and rTEDB tuples.

Table 14 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	The DALB tuple is not properly linked to TEDB.
Possible Error Effect:	The calls will be torn down. If application is accessing DALB, assert will fire.
Corrective Action Taken:	Not applicable for this audit but TEDB and DALB audit will recover the error tuples back to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Zero.
<i>Good Data =</i>	Not applicable.
<i>Logical Index =</i>	LTMSTAT key.
<i>Dump =</i>	rLTMSTAT tuple.

Table 15 Error Code: TYPE

Error Code:	TYPE
Error Description:	A port name could not be found when reading FC_PORTYP.
Possible Error Effect:	Calls may be invalid set up, or may not be set up correctly, or may be torn down. Communication to the customer premise equipment associated with this LTMSTAT is questionable.
Corrective Action Taken:	Idle out all call appearance bits in the bu_stat attribute of the LTMSTAT tuple.
Manual Action Required:	The port name lpt_buf was gotten from LPT_PT. When this information was used to read FC_PORTTYP, no match was found. FC_PORTTYP should be investigated and corrected.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	FC_PORTTYP.port.member.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	LTMSTAT key.
<i>Dump</i> =	rlTLMSTAT tuple.

LTSBPORT

Software Release: 5E10 and later

1. LINE TIME SLOT BRIDGING PORT (LTSBPORT) AUDIT

The LTSBPORT audit verifies the static rLTSBPORT tuple to dynamic rLTSBSTAT tuple existence. It runs segmented in the operating system for distributed switching-switching module (OSDS-SM) environment, and during a selective init.

2. DATA STRUCTURES

The RLLTSLBPORT relation and RLLTSBSTAT relation both are indirect access non-sequential key static relations. The dynamic rLTSBSTAT tuple maintains the current busy/idle status of the two ports that are assigned as a bridged group. The static RLLTSLBPORT relation contains the port information of the group.

3. AUDIT CHECKS

The following checks are performed in the static-to-dynamic function:

- (a) For every rLTSBPORT tuple with a specific attribute indicating that this is a lead port, an associate dynamic rLTSBSTAT tuple should exist.
- (b) If there is not an associate rLTSBSTAT tuple, the audit checks to see if a recent change is in progress.
- (c) If no recent change activity is active, the audit attempts to insert the associate tuple. If it is successful, the audit is completed. Otherwise an assert is reported.

4. ERROR HANDLING

The LTSBPORT audit schedules LTSBSTAT as a related audit if an error occurs.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the dynamic RLLTSBSTAT relation; the corresponding static tuple is in the RLLTSLBPORT relation.
Possible Error Effect:	Application codes will ASSERT and calls originated for this LTSB group will be rejected.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLTSBPORT tuple.
<i>Bad Data =</i>	0--Missing rLTSBSTAT.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of lost tuple.
<i>Dump =</i>	rLTSBPORT tuple.

LTSBSTAT

Software Release: 5E10 and later

1. LINE TIME SLOT BRIDGING STATUS (LTSBSTAT) AUDIT

The LTSBSTAT audit verifies the dynamic rLTSBSTAT tuple to static rLTSBPORT tuple existence, and maintains consistency of the bridged LTSB ports by applying the keep alive mechanism. The audit runs segmented in the operating system for distributed switching-switching module (OSDS-SM) environment.

2. DATA STRUCTURES

The RLLTSBSTAT relation contains the busy/idle status of the two ports that are assigned as a bridged group. For each rLTSBSTAT tuple there must be a corresponding rLTSBPORT tuple with a specific attribute indicating that it is the lead port of the group. The RLLTSBPORT relation and RLLTSBSTAT relation both are indirect access, non-sequential key relations.

3. AUDIT CHECKS

The LTSBSTAT entry function checks to see if recent change is in progress and if so, the audit completes by returning to audit control. The following checks are performed in the dynamic-to-static function:

- (a) The rLTSBSTAT key verification is checked against the key gotten from a dynamic access (DA) subsystem macro.
- (b) For every rLTSBSTAT tuple, there must be a corresponding rLTSBPORT tuple with a specific attribute indicating a lead port.
- (c) If there is a corresponding rLTSBPORT tuple, a specific attribute should indicate the port as lead.
- (d) If any of the checks fail, the audit attempts to delete the extra rLTSBSTAT tuple. If it is successful, the audit is completed. Otherwise an assert is reported.

The keep alive audit function performs the following for the lead port and the non-lead port of the bridged group:

- (a) Check to see if the call status indicates idle and if so, returns without any further checks.
- (b) If the call status indicates busy and the keep alive (KA) bit is set to DBNO, the KA bit sets to DBYES.
- (c) If the call status indicates busy and the KA bit is set to DBYES, set the status bit to idle and the KA bit to DBNO.

4. ERROR HANDLING

The LTSBSTAT audit schedules LTSBPORT as a related audit if an error occurs.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The rLTSBSTAT tuple should not be populated at this SM, because the

	corresponding static rLTSPORT tuple does not exist.
Possible Error Effect:	LTSP calls originated may be erroneously accepted or rejected.
Corrective Action Taken:	The rLTSPSTAT tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rLTSPSTAT tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rLTSPSTAT tuple.
<i>Dump</i> =	rLTSPSTAT tuple.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	An Invalid key value is found in the rLTSPSTAT tuple.
Possible Error Effect:	Incorrectly used resources.
Corrective Action Taken:	A key error will be reported and corrected by the LTSPSTAT audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rLTSPSTAT tuple.
<i>Bad Data</i> =	The error key.
<i>Good Data</i> =	Correct key value.
<i>Logical Index</i> =	Logical key of rLTSPSTAT tuple.
<i>Dump</i> =	rLTSPSTAT tuple.

MALLOC

Software Release: 5E10 and later

1. MEMORY ALLOCATION BLOCK (MALLOC) AUDIT

For a TR303 remote terminal (RT) on integrated digital carrier unit (IDCU) the rIMALLOC tuple is used as an extra storage area for messages that have not been decoded. The rIMALLOC tuple is linked to the rIRTLB tuple when used and the rIDB_HEADC tuple when not needed.

The MALLOC audit maintains the integrity of these linkages.

2. DATA STRUCTURES

The rIMALLOC tuple is linked to the idle list (rIDB_HEADC) or the application process that is decoding the message (rIRTLB). The link_owner field is used to determine the correct linkage.

The dynamic data structures for EOC/TMC message processing on TR303 RT are shown in Figure 1 .

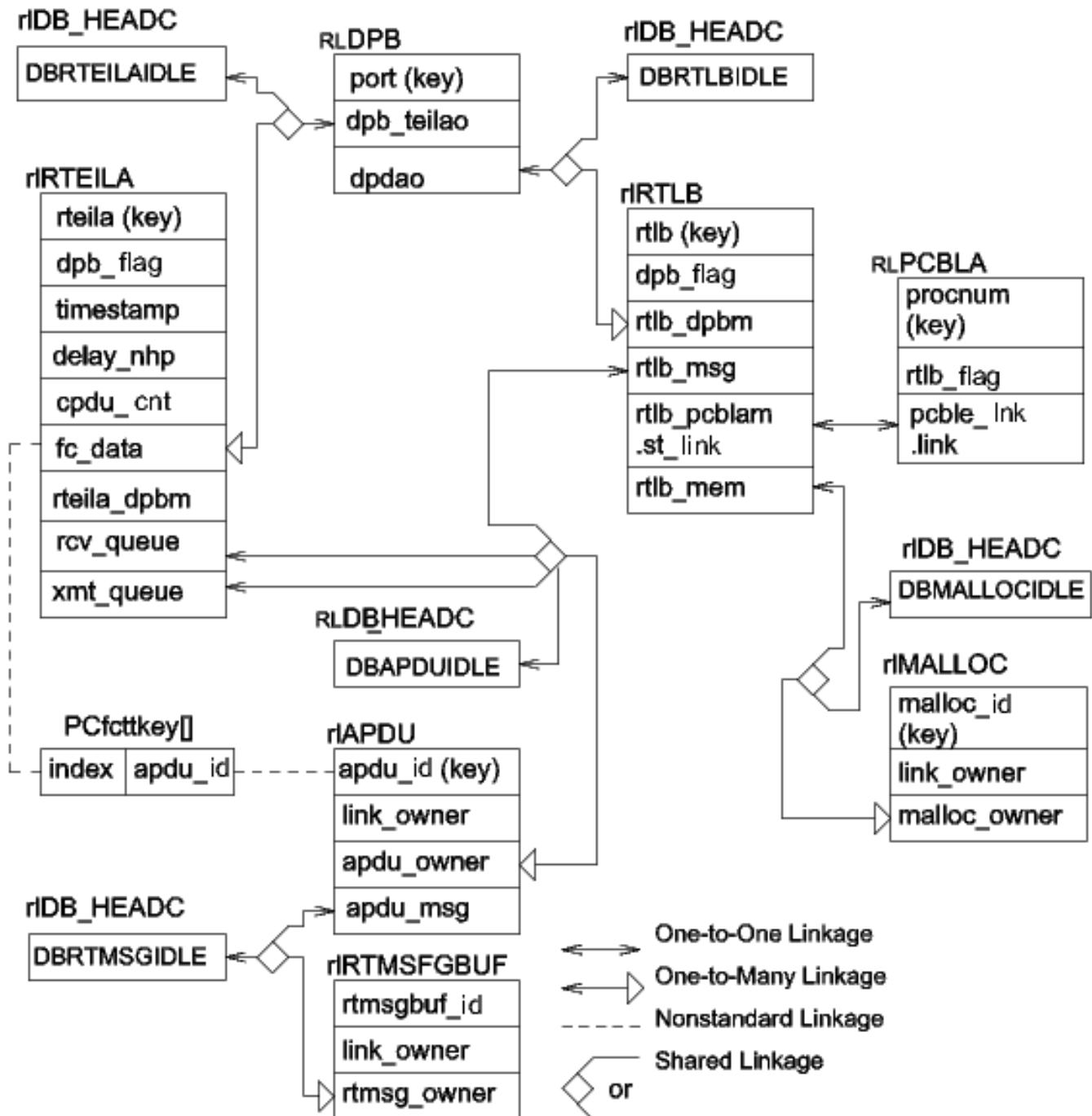


Figure 1 (MALLOC-1) Dynamic Data Structures for EOC/TMC Message Processing on TR30 RT

3. AUDIT CHECKS

The MALLOC audit invokes the DRINASEK driver to make the linkages and semantic checks for each rIMALLOC tuple. The MALLOC audit uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given. For each tuple, the MALLOC audit makes the following checks:

MEMBAD = Check member linkage to rRTLB or rIDB_HEADC.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	This rIMALLOC tuple is not linked to the owner indicated by the flag (rIMALLOC.link owner).
Possible Error Effect:	The messages for this channel could be incomplete.
Corrective Action Taken:	This rIMALLOC tuple will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIMALLOC tuple.
<i>Bad Data =</i>	Key of owner tuple (rIRTLB or rIDB_HEADC).
<i>Good Data =</i>	Flag for owner field (1 for rIRTLB or 0 for rIDB_HEADC).
<i>Logical Index =</i>	Key of rIMALLOC (rIMALLOC.malloc_id)
<i>Dump =</i>	Contents of rIMALLOC tuple.

MATESM

Software Release: 5E10 and later

1. MATE SWITCHING MODULE (MATESM) AUDIT

The MATESM audit is part of the Terminal Maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPCKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The MATESM audit uses the terminal maintenance mate switching module array (TMmatesm) and the switching module port table relation (RLSMPORTBL).

3. AUDIT CHECKS

The audit checks the TMmatesm array against the RLSMPORTBL relation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit has gotten a bad return from DBfrdtup() when reading the rLSMPORTBL relation.
Possible Error Effect:	The ODD could be bad.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should check the rLSMPORTBL relation. The key to the relation is printed in <i>Good Data</i> .
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Return value from DBfrdtup().
<i>Good Data</i> =	The key used to read the relations.
<i>Logical Index</i> =	The index to the TMmatesm[] array.
<i>Dump 1</i> =	The TMmatesm[] array element.
<i>Dump 2</i> =	The rLSMPORTBL tuple.

Table 2 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The TMmatesm[] array does not agree with the rLSMPORTBL relation. One indicates the inter-SM nailup is present and the other says no.
Possible Error Effect:	The inter-SM nailup could be out of order.
Corrective Action Taken:	TMmatesm[] is updated to agree with rLSMPORTBL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Indicator found in TMmatesm[].
<i>Good Data</i> =	Indicator derived from rLSMPORTBL.
<i>Logical Index</i> =	The index to the TMmatesm[] array.
<i>Dump 1</i> =	The TMmatesm[] array element.
<i>Dump 2</i> =	The rLSMPORTBL tuple.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit has been entered by audit control to clean up after a reset.
Possible Error Effect:	Causes resets and or asserts.
Corrective Action Taken:	The audit is restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

MCB

Software Release: 5E10 and later

1. MESSAGE CONTROL BLOCK (MCB) AUDIT

Processes communicate primarily through a message mechanism provided by operating system for distributed switching (OSDS). A sending process prepares a message in a message buffer and then invokes OSDS by means of a message communication primitive, which transmits the message. The name of the receiving process is given as a parameter in the message communication primitive.

To deliver a message to a process, OSDS links a data structure called a message control block to a list owned by the process control block of the receiving process. The message control block contains linkage items, a message header, and the message text. Each message control block in the list owned by a process control block represents a pending message. A fixed number of message control blocks is provided by OSDS per processor.

The MCB audit runs in the OSDS kernel process (OKP) and switch maintenance kernel process (SMKP) environments in the administrative module (AM), in the switching module (SM) and the communications module processor (CMP).

2. DATA STRUCTURES

The fields checked by this audit are the llink, the rlink, and th mowner. In addition to the MCBs themselves (OSmcb[]), the audit verifies and uses the following global data:

- The array of process control blocks (OSbpcb[]) any of which may own a list of MCBs (attached using PCB.msg0 and PCB.msgn).
- The global indexes OSffmcb and OSlfmcb which indicate the first and last members of the doubly linked MCB idle list.

Figure 1 illustrates how message control blocks are linked into per-process message lists and how these lists are linked to a process control block.

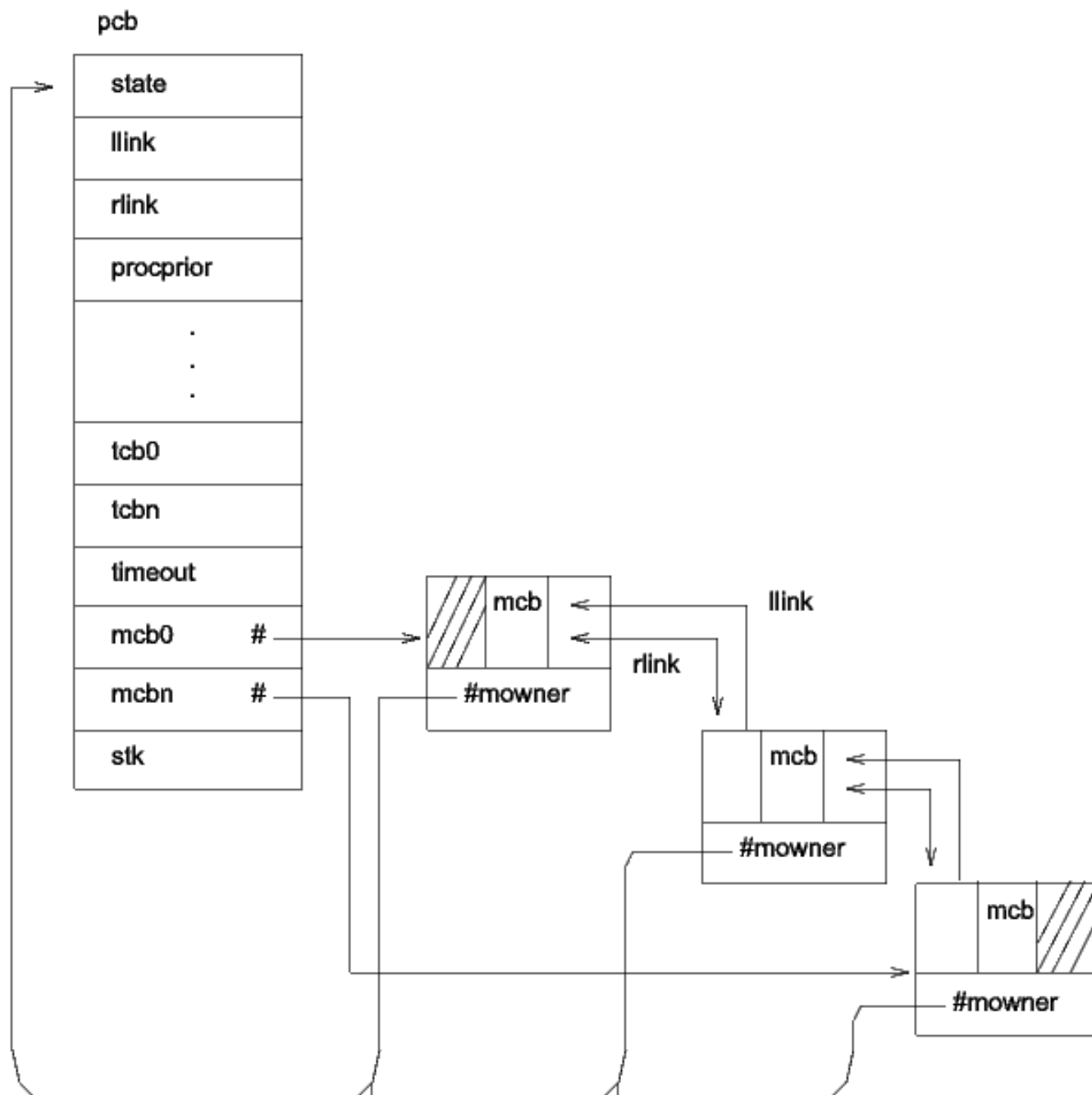


Figure 1 (MCB-1) Per-Process Message Linkage

3. AUDIT CHECKS

The MCB audit does not use any audit driver. All error codes associated with this audit are described. The following list gives a basic indication of the order of the checks. For details regarding each error code, see the comment blocks given.

If the audit is being (re)entered:

ZAP = Check for system interrupt handling.

If the audit is verifying the idle list:

LNK2ALST = Range check OSIfmcb.
 ALNK2QUE = Semantic check of OSIfmcb.
 BLNKINQUE = Semantic check of OSffmcb.
 LNKIALST = Range check idle MCB's llink field.
 PT2PT = Point-to/point-back check between MCBs.
 BADOWNER = Check idle MCB's mowner field.

If the audit is verifying an MCB list attached to a PCB:

LNK2ALST = Range check PCB.msgn.
 ALNK2QUE = Semantic check PCB.msgn.
 BLNKINQUE = Semantic check PCB.msg0.
 LNKIALST = Range check MCB.llink.
 PT2PT = Point-to/point-back check between MCBs.
 BADOWNER = Check MCB's mowner field.

If the audit is looking for lost MCBs:

INVOWNER = Range check potentially lost MCB's mowner field.
 INV = Check for lost MCBs.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALNK2QUE

Error Code:	ALNK2QUE
Error Description:	An index to the last element in a list of MCBs (OSIfmcb if the free list is being checked, otherwise PCB.msgn for some non-idle PCB) indicates an MCB which either has an incorrect owner (not OSNULL for the free list, different from the PCB number in all other cases) or has a non-null (not OSNULL) rlink field.
Possible Error Effect:	The list may not be accessed correctly. For the free list this could mean lost MCBs (leading to overload situations) or non-idle MCBs being taken for new processes (affecting the process to which they rightfully belong). For MCB lists attached to PCBs this problem could cause the owning process to lose messages or falsely believe it has messages which actually belong to

	the idle list or other processes. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the MCB audit in the directed mode. It is also likely that MCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of MCBs and the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying OSffmcb (index to the first free MCB) and, if it is ok, verifying the list from the top down as far as possible. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the MCB.
<i>Bad Data =</i>	The owner of the list we're trying to access (either OSNULL or a PCB index).
<i>Good Data =</i>	Contents of the MCB's mowner field.
<i>Logical Index =</i>	The MCB's index (= OSIfmcb in the free list case, = PCB.msgn otherwise).
<i>Dump 1 =</i>	The MCB.
<i>Dump 2 =</i>	The owning PCB or nothing, if the free list was being checked.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	An MCB's mowner field shows a different owner than the owner of the list it was found in. Note that the MCBs rlink field has already been verified (PT2PT) and its llink has been checked to be in range (BLINKINQUE/LNKIALST).
Possible Error Effect:	If the owning list belongs to a PCB then the associated process could erroneously receive a message. Regardless of where the MCB was found, it is possible that it actually belongs on a list belonging to some process in which case that process will be missing a message. If the MCB was found on the free list, than INV errors are also likely to show up.
Corrective Action Taken:	If this is the free list of MCBs, just relink the list around the bad message and let an INV error fire, to put it back on the idle list, if that's where it really belongs. If the list is attached to a PCB, the associated process will be purged. Although the offending MCB may actually belong somewhere else and the remaining MCBs on this list are ok, this list is still corrupted and can not be trusted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the MCB.
<i>Bad Data =</i>	Contents of the MCB's mowner field.
<i>Good Data =</i>	The owner of the list the MCB was found in (either OSNULL or a PCB index).
<i>Logical Index =</i>	The MCB's index (= OSIfmcb in the free list case, = PCB.msgn otherwise).
<i>Dump =</i>	The MCB.

Table 3 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	While traversing a list (bottom up) an MCB was found with its llink field = OSNULL but was not the first MCB in the list, according to the list owner. Specifically, if the MCB was found on the free list, it was not indicated by OSffmcb (index to the first free MCB).

	If the MCB was found on a list belonging to a PCB, it was not indicated by the PCB's msg0 field (index to the first MCB belonging to the process).
Possible Error Effect:	The list is either broken, or the list owner's index to the first element is corrupt. This could have any of the effects of an ALNK2QUE, LNKIALST or PT2PT error.
Corrective Action Taken:	If this is the free list of MCBs and the audit is running in segmented mode, escalate to the directed mode and rebuild the list. If the audit is already running in the directed mode, then rebuild the list right away. The list will be rebuilt by verifying OSffmcb (index to the first free MCB) and, if it is ok, verifying the list from the top down as far as possible. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the MCB.
<i>Bad Data</i> =	Index of the MCB, the list's owner says is first on the list.
<i>Good Data</i> =	The owner of the list the MCB was found in (either OSNULL or a PCB index).
<i>Logical Index</i> =	The MCB's index.
<i>Dump 1</i> =	The MCB.
<i>Dump 2</i> =	The owning PCB or nothing if the free list was being checked.
<i>Dump 3</i> =	The owning PCB or nothing if the free list was being checked.

Table 4 Error Code: INV

Error Code:	INV
Error Description:	An MCB has been found which is not linked to any PCB or the free list.
Possible Error Effect:	The MCB is lost and inaccessible to the system. If enough MCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	The MCB is cleared and linked to the idle list. In particular, its mowner field is set to OSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the MCB.
<i>Bad Data</i> =	0 (Indicates that the audit did not find the MCB on any list).
<i>Good Data</i> =	1 (Indicates that the audit should have found the MCB on some list).
<i>Logical Index</i> =	The MCB's index.
<i>Dump</i> =	The MCB.

Table 5 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	An MCB has been found which the audit did not see on any list. Before taking an INV error, the audit will verify that the MCB is truly lost. For example, if the audit were running in the segmented mode, the MCB may have changed lists while the audit took a timebreak. In this case, the audit would not have seen the MCB before now. The audit uses the MCB's mowner field to determine which list to check. This error says that the mowner field is out of range (not OSNULL and not a PCB index).
Possible Error Effect:	The MCB is effectively lost and inaccessible to the system. If enough MCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	If possible, the MCB is unlinked from the list its in (the linkages of the MCBs it points to are reset). In any case, the MCB is cleared and linked to the idle list. In particular, its mowner field is set to OSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the MCB.
<i>Bad Data</i> =	Contents of the MCB's mowner field.
<i>Good Data</i> =	OSNULL.
<i>Logical Index</i> =	The MCB's index.
<i>Dump</i> =	The MCB.

Table 6 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Either the tail of an MCB list was null when the head was not or the tail of an MCB list is out of range. The list owner ("Good Data" below) is either a PCB or the free list (OSNULL). If it is a PCB, the out of range value is the PCB's "msgn" field. If the owner is the free list, the out of range value is the last free MCB index (OSfmcmb).
Possible Error Effect:	If the owner of the list is a PCB, that process may fail to receive its messages. The results of this depend on the process and message types. If the free list is involved, this could result in lost MCBs which could, eventually, lead to OSDS MCB overloads.
Corrective Action Taken:	If the MCB list belongs to a PCB, that process will be purged. Otherwise, if the audit is running segmented, it will escalate to directed mode. If the audit is already running in directed mode it will attempt to rebuild the idle list from the top (OSfmcmb) down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The out of range MCB index.
<i>Good Data =</i>	The list owner (a PCB index or OSNULL for the idle list).
<i>Logical Index =</i>	OSNULL.
<i>Dump =</i>	The PCB (if any) which owns the list.

Table 7 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	While traversing a list, an MCB was found with an out of range llink field (not OSNULL and not an MCB index).
Possible Error Effect:	The list is broken. For the free list, this could mean lost MCBs (leading to overload situations). For MCB lists attached to PCBs, this problem could cause the owning process to lose messages. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the MCB audit in the directed mode. It is also likely that MCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of MCBs and the audit is running in segmented mode, it will escalate to the directed mode and rebuild the list. If the audit is already running in the directed mode, it will rebuild the list right away. The list will be rebuilt by verifying OSfmcmb (index to the first free MCB) and, if it is ok, verifying the list from the top down as far as possible. It will then link this segment (top down) with the segment (bottom up) that the audit has already checked. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the MCB.
<i>Bad Data =</i>	The MCB's llink field.
<i>Good Data =</i>	OSMCBNUM (llink should have been OSNULL or less than this value).
<i>Logical Index =</i>	The MCB index.
<i>Dump =</i>	The MCB.

Table 8 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	While traversing a list, an MCB was found whose llink field contained the index of an MCB whose rlink field was not equal to the first MCB's index.

Possible Error Effect:	The list is broken. For the free list, this could mean lost MCBs (leading to overload situations) or non-idle MCBs being taken for new processes (affecting the process to which they rightfully belong). For MCB lists attached to PCBs, this problem could cause the owning process to lose messages or falsely believe it has messages which actually belong to the idle list or other processes. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the MCB audit in the directed mode. It is also likely that MCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of MCBs and the audit is running in the segmented mode, the audit will escalate to the directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by first verifying OSffmcb (index to the first free MCB) and, if it is ok, verifying the list from the top down as far as possible. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the MCB.
<i>Bad Data =</i>	The MCB's llink field.
<i>Good Data =</i>	The rlink field of the MCB indexed by "BAD DATA."
<i>Logical Index =</i>	The MCB's index.
<i>Dump =</i>	The MCB.

Table 9 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, out of range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will idle the last MCB it was looking at (if any), abort and be restarted in directed mode. Otherwise it will just be restarted in directed mode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the MCB (if any).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the MCB (if any).
<i>Dump =</i>	The MCB (if any).

MDB

Software Release: 5E10 and later

1. MODEL DATA BLOCK (MDB) AUDIT

Model Data Blocks are used by any BRCS feature to define the associated models (MFAR1, MFAR2, and so forth) of a BRCS call. The model data block audit verifies the semantic data and linkages of the model data block relation. It runs segmented and during selective initializations.

2. DATA STRUCTURES

The RLLMDB relation is a dynamic indexed sequential relation that is used to record FEX model data for BRCS calls. When the rIMDB tuple is busy it is linked to an rICLDB tuple involved in a stable call. All idle rIMDB tuples are linked to an idle list represented by a specific tuple in the RLCLDB relation.

A representation of the general scenario is shown in Figure 1 .

Some rIMDB tuples may be used for call park feature models. For this scenario, the **model_id** will be marked PARK_FM, and only one rICLDB tuple will be linked off of RLPCBLA. This rICLDB tuple serves as the owner of the rIMDB park feature model tuple. Although more than one rIMDB tuple may be linked off of RLCLDB, for call park there will usually be only one RLMDDB. For cases where there are more than one RLMDDB linked off of RLCLDB, the call park feature model rIMDB tuple will usually be the first one on the list.

All parked calls are represented by an entry in the RLPARK_LOT dynamic relation. This relation contains a tuple entry for each DN on a given SM that has a call parked against it. This relation is hashed. Tuple entries are created as calls get parked, and are subsequently deleted as calls get unparked. Each tuple contains data related to the parked DN. This data includes **locdn** (key) which is the parked DN, **park_pid** which is the call park terminal process, and **touch** which is the keep alive touch bit.

Call park processes and RLPARK_LOT table entries need not reside on the same SM. For instances where the RLPARK_LOT table entry and the call park process reside on the same SM, there exists an implied linkage between the **pk_dn** field represented in the RLMDDB PARK_DATA feature data block and the **locdn** (key) field in RLPARK_LOT, between **pk_mod** in the RLMDDB PARK_DATA feature data block and **park_pid.pcrd** in RLPARK_LOT, and between **park_pid.procno** in RLPARK_LOT and rIPCBLA.procnum in the associated rIPCBLA tuple.

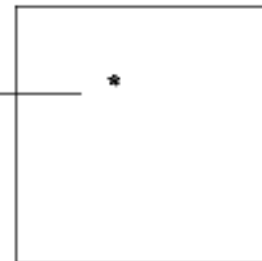
All of these implied linkages are audited by the MDB audit for cases where the call park terminal process and RLPARK_LOT table entries are local to the same SM, and by call park keep alive for cases where this data resides on different SMs. A **parked** bit in RLPCBLA indicates if the rIPCBLA tuple is being used for a call park process, and is always marked DBYES for call park feature models.

A representation of the call park feature model scenario is shown in Figure 2 .

rPCBLA

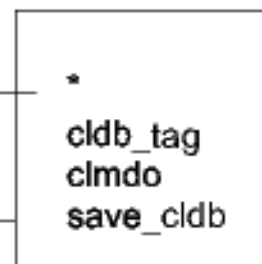
PROCESS NUMBER (procnum)
CLDB_FLAG (apb or cldb)
CALL WAITING IN PROGRESS (cw inpro)
.
.
.
LINK TO rAPB OR rCLDB
.
.
.
.

rAPB



rCLDB

OR



rMDB



—— * = ONE-TO-MANY LINKAGE

OR = SHARED LINKAGE

Figure 1 (MDB-1) Data Structures of MDB Audit

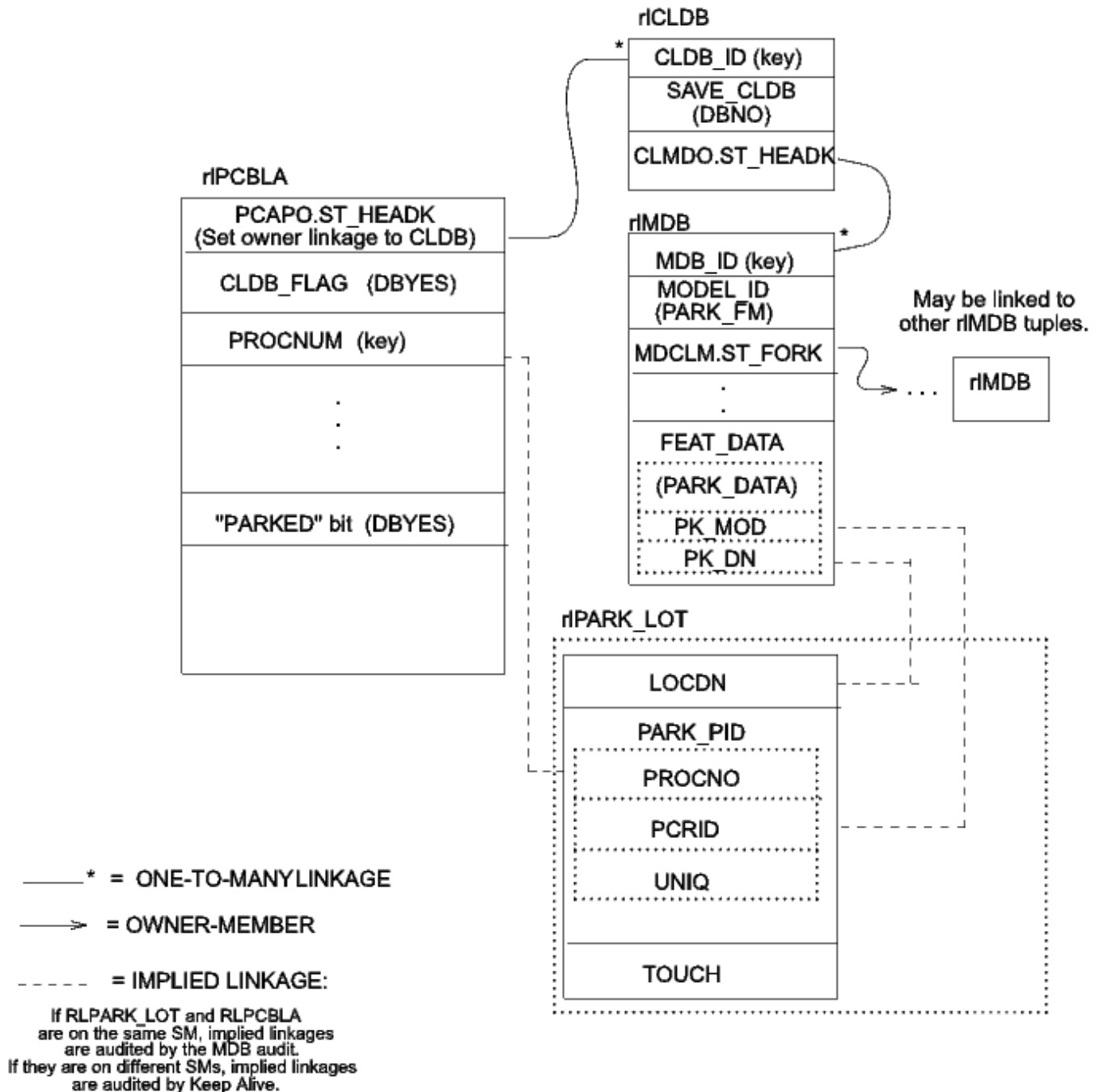


Figure 2 (MDB-2) Data Structures of the MDB Audit for Call Park Feature Model Scenario

3. AUDIT CHECKS

The MDB audit steps through the RLMDb relation using the DRINASEK audit driver, verifying linkages with rCLDB.

4. ERROR HANDLING

If linkage or semantic errors are detected, the RLMDb is recovered to the idle list, and any associated terminal processes are purged. Related audits are run to recover other data structures.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	An associated RLPARK_LOT relation table entry could not be accessed. For rIMDB tuples used as call park feature models, (rIMDB.spdb.model_id = PARK_FM), if the park feature model data block indicates that a RLPARK_LOT table entry should exist on this SM (rIMDB.feats_data.pk_mod = GI_pcrd) then a corresponding rIPARK_LOT tuple should exist where rIPARK_LOT.locdn = rIMDB.feats_data.pk_dn.
Possible Error Effect:	Possible failures or asserts when call processing attempts to use this rIMDB tuple to park a call.
Corrective Action Taken:	The rIMDB tuple is recovered to an idle state along with the rCLDB and rPCBLA tuples, if accessible. Any process found associated with the rIMDB tuple will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMDB tuple found in error.
<i>Bad Data =</i>	NA
<i>Good Data =</i>	The status of the parked bit in RLPCBLA (DBYES or DBNO).
<i>Logical Key =</i>	The logical key of the rIMDB tuple (rIMDB.mdb_id).
<i>Dump 1 =</i>	The contents of the rIMDB tuple.
<i>Dump 2 =</i>	The contents of the PARK_DATA feature data block union in the rIMDB tuple ((PARK_DATA)rIMDB.feats_data).

Table 2 Error Code: CAMPSTATE

Error Code:	CAMPSTATE
Error Description:	The MDB does not find entry on the camp-on list.
Possible Error Effect:	The process would be in a stuck state.
Corrective Action Taken:	The corresponding process is purged from the system and the mdb is returned to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of mdb.
<i>Bad Data =</i>	State.
<i>Good Data =</i>	NA.
<i>Logical Key =</i>	mdb key.
<i>Dump 1 =</i>	rIMDB tuple.
<i>Dump 2 =</i>	rPCBLA tuple.

Table 3 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	For a given rIMDB tuple which is associated with a call park feature model (rIMDB.spdb.model_id == PARK_FM), the corresponding rPCBLA tuple for the call park terminal process could not be accessed. A relationship exists such that for every rIMDB tuple associated with a PARK feature model, there should exist a corresponding rPCBLA tuple, where rPCBLA.procnum = rCLDB.clpcm.st_ownd - DBSFCLDB, for the rCLDB tuple which is the PARK_FM rIMDB tuple's owner (rCLDB.cldb_id = rIMDB.mdcldm.st_ownd - DBSFMDB).
Possible Error Effect:	An rIMDB tuple cannot be used, and is a lost resource. This may cause possible call failures if the system exhausts the number of available rIMDB tuples.
Corrective Action Taken:	The rIMDB tuple is recovered to an idle state, and the CLDB and PCBLA related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rIMDB tuple.
<i>Bad Data</i> =	Logical key of the rIPCBLA tuple found.
<i>Good Data</i> =	Logical key of the rIPCBLA tuple that was marked in the rICLDB tuple found linked to the rIMDB tuple under audit.
<i>Logical Key</i> =	The logical key of the rIMDB tuple.
<i>Dump 1</i> =	The contents of the rIMDB tuple.
<i>Dump 2</i> =	The contents of the rICLDB tuple.
<i>Dump 3</i> =	The contents of the SPDB union in the rIMDB tuple.

Table 4 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A tuple of the model data block dynamic relation "mdb_tag" is out-of-range.
Possible Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMDB tuple.
<i>Bad Data</i> =	Value of mdb_tag.
<i>Good Data</i> =	NA.
<i>Logical Key</i> =	Key of rIMDB tuple.
<i>Dump 1</i> =	rIMDB tuple.
<i>Dump 2</i> =	Owning rICLDB tuple.

Table 5 Error Code: PCRID

Error Code:	PCRID
Error Description:	The SM marked in the PARK_DATA feature data block union in the rIMDB tuple ((PARK_DATA)rIMDB.feats_data.pk_mod) is either out of range or not in operation. This field should be in the legal range of SM values (between GL1STSM and GLLASTSM inclusive) and should reference an SM that is equipped and in operation.
Possible Error Effect:	The RLPARK_LOT table entry for the rIMDB tuple cannot be accessed. Calls cannot be parked or unparked against the DN marked in the feature data block ((PARK_DATA)rIMDB.feats_data.pk_dn).
Corrective Action Taken:	The rIMDB tuple is recovered to an idle state along with the rICLDB and rIPCBLA tuples, if accessible. Any process found associated with the rIMDB tuple will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of rIMDB tuple found in error.
<i>Bad Data</i> =	The SM represented in pk_mod in the PARK_DATA feature data block union.
<i>Good Data</i> =	NA.
<i>Logical Key</i> =	The logical key (rIMDB.mdb_id) of the rIMDB tuple.
<i>Dump 1</i> =	The contents of the rIMDB tuple.
<i>Dump 2</i> =	The contents of the PARK_DATA feature data block union in the rIMDB tuple ((PARK_DATA)rIMDB.feats_data).

Table 6 Error Code: TAG

Error Code:	TAG
Error Description:	A tuple of the model data block dynamic relation "mdb_tag" is incorrect.
Error Effect:	Incorrect use of resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is zeroed, and recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMDB tuple.
<i>Bad Data</i> =	Value of mdb_tag.
<i>Good Data</i> =	Value of associated rICLDB tag.
<i>Logical Key</i> =	Key of rIMDB tuple.
<i>Dump 1</i> =	rIMDB tuple.
<i>Dump 2</i> =	rICLDB tuple.

Table 7 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	A tuple of the model data block dynamic relation is linked to an incorrect list.
Possible Error Effect:	Lost resources vital to BRCS call processing.
Corrective Action Taken:	The dynamic tuple is recovered to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMDB tuple.
<i>Bad Data</i> =	Key of incorrect list.
<i>Good Data</i> =	Key of correct idle list.
<i>Logical Key</i> =	Key of rIMDB tuple.
<i>Dump 1</i> =	rIMDB tuple.
<i>Dump 2</i> =	Owning rCLDB tuple.

MDRQ

Software Release: 5E10 and later

1. MESSAGE DETAIL RECORDING QUEUE (MDRQ) AUDIT

Message detail recording (MDR) of private facility calls using the revenue accounting office provides automated message accounting (AMA) records, on a per outgoing call basis, on private facilities selected by the customer. Records may be requested for WATS facilities (by band) for foreign exchange lines, TIE trunks, tandem TIE trunks, CCSA access lines, ETS trunks, and so on. The per call information allows customers to perform such tasks as cost accounting and engineering.

The MDR information is stored in the RLMDRQ relation, and is audited by the MDRQ audit. The MDRQ audit runs in routine and elevated segmented modes.

2. DATA STRUCTURES

The MDRQ audit checks rIMDRQ tuples to determine if the tuples are validly marked busy or idle. The audit makes these checks based on the following data:

- mdrqkey* = The logical key of rIMDRQ tuples.
- mdrqlnk.st_ownk* = Owner linkage field to the RLDB_HEADC idle list head cell. Idle rIMDRQ tuples should have this field marked to DBMDRQIDLE + DBSFDB_HEA. For busy rIMDRQ tuples, this field should always be DBMDRQBUSY + DBSFDB_HEA.
- ap_id* = Site identifier. For busy rIMDRQ tuples, this value should not be zero. For idle rIMDRQ tuples, this value should be zero.

3. AUDIT CHECKS

For each tuple in the RLMDRQ relation, the MDRQ audit makes the following checks: IF the MDRQ audit get entered:

- NOTZERO = Check that the site identifier (rIMDRQ.ap_id) is zero for idle rIMDRQ tuples.
- NOTBUSY = Check that the site identifier (rIMDRQ.ap_id) is not zero for busy rIMDRQ tuples.
- BUSYIDLE = Check that all rIMDRQ tuple are linked to a busy list or idle list head cell (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA == DBMDRQIDLE or DBMDRQBUSY).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	An rIMDRQ tuple was found unlinked to a busy list or idle list head cell. All rIMDRQ tuples must be linked to a busy or idle list head cell in the RLDB_HEADC relation (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA == DBMDRQIDLE or DBMDRQBUSY).

Possible Error Effect:	Possible exhaustion of MDR resources, and incorrect measurements data provided to customers.
Corrective Action Taken:	The rIMDRQ tuple is recovered to the RLDB_HEADC idle list (rIMDRQ.mdrqlnk.st_ownk set to DBMDRQIDLE + DBSFDB_HEA).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMDRQ tuple.
<i>Bad Data =</i>	The rIMDRQ tuple's head cell linkage (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIMDRQ tuple (rIMDRQ.mdrqkey).
<i>Dump =</i>	The contents of the rIMDRQ tuple.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rIMDRQ tuple was found linked to a busy list head cell, but the site identifier was not marked to any given value. All rIMDRQ tuples linked to the busy list head cell in the RLDB_HEADC relation (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA == DBMDRQBUSY) must have the site identifier (rIMDRQ.ap_id) marked to a non-zero value.
Possible Error Effect:	Possible exhaustion of MDR resources, and incorrect measurements data provided to customers.
Corrective Action Taken:	The rIMDRQ tuple is recovered to the RLDB_HEADC idle list (rIMDRQ.mdrqlnk.st_ownk set to DBMDRQIDLE + DBSFDB_HEA).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMDRQ tuple.
<i>Bad Data =</i>	The contents of the RLMDRQ site identifier (rIMDRQ.ap_id).
<i>Good Data =</i>	The owner linkage to the RLDB_HEADC head cell relation (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA).
<i>Logical Index =</i>	The key of the rIMDRQ tuple (rIMDRQ.mdrqkey).
<i>Dump =</i>	The contents of the rIMDRQ tuple.

Table 3 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	An rIMDRQ tuple was found linked to a idle list head cell, but the site identifier was not marked to zero. All rIMDRQ tuples linked to the idle list head cell in the RLDB_HEADC relation (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA == DBMDRQIDLE) must have the site identifier (rIMDRQ.ap_id) marked to zero.
Possible Error Effect:	Possible exhaustion of MDR resources, and incorrect measurements data provided to customers.
Corrective Action Taken:	The rIMDRQ tuple is recovered to the RLDB_HEADC idle list (rIMDRQ.mdrqlnk.st_ownk set to DBMDRQIDLE + DBSFDB_HEA).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMDRQ tuple.
<i>Bad Data =</i>	The contents of the RLMDRQ site identifier (rIMDRQ.ap_id).
<i>Good Data =</i>	The owner linkage to the RLDB_HEADC head cell relation (rIMDRQ.mdrqlnk.st_ownk - DBSFDB_HEA).
<i>Logical Index =</i>	The key of the rIMDRQ tuple (rIMDRQ.mdrqkey).
<i>Dump =</i>	The contents of the rIMDRQ tuple.

MELNK-A**Software Release:** 5E12 only**1. MCTSI-BASED ETHERNET LINK (MELNK) AUDIT**

The MELNK audit is responsible for checking dynamic data maintained by the ethernet link fault recovery system process (ELSP). This data includes the maintenance status of MELNKs and MCTSI-based ethernet pipes (MEPIPEs). It also includes dynamic copies of information found in the static relations RLENETPIPE and RLENETLINK. Consistency between related dynamic data structures and consistency with static ODD are both verified. All audited data is SM-2000 SMP-resident. Other software mechanisms are used to ensure that the SMP-resident data is consistent with MEPIPE/MELNK status data stored in the SM-2000 message handler.

2. DATA STRUCTURES

The following static data is used by the MELNK audit:

RLENETPIPE = Database relation containing MEPIPE data.

RLENETLINK = Database relation containing MELNK data.

Glprtc = Global parameter identifying the global billing SM.

The following dynamic data is checked by the MELNK audit:

SMelstat[] = Dynamic data maintained by ELSP for MEPIPEs and MELNKs.

SMelmcsc = Data related to ELSP's MCTSI/MH escalation strategy.

SMelmrbuf = Buffer for data about in progress manual MELNK remove/restore requests.

3. AUDIT CHECKS

The MELNK audit makes the following checks:

ACTION = Check MEPIPE/MELNK maintenance and recovery action data.

AVAIL_CNT = Check the number of equipped child MELNKs for an MEPIPE.

BADTYPE = Check the MEPIPE/MELNK recovery timer types.

BILLFLG = Check the value of the DDRTCDELNK data delivery key.

DYN_STAT = Check for consistency between MEPIPE/MELNK static data and dynamic copies of that data.

INV_STATE = Check for valid MEPIPE/MELNK maintenance states.

INV_TIMER = Check existence of valid timer indexes for MEPIPE/MELNK maintenance actions.

LNKS_MISSING = Check that a child MELNK is being restored when MEPIPE is in the INIT state.

LNK_CONFIG = Check manual request buffer for consistency with MELNK dynamic data.

MISSTUP = Check MEPIPE/MELNK equipage state in dynamic data against static data (when there is no MEPIPE/MELNK tuple in static data).

PARENT	= Check MEPIPE/MELNK parent/child relationships.
PORT_BAD	= Check for incorrect MELNK active port counts.
STATE_VAR	= Check state of ELSP's MCTSI/MH escalation/recovery strategy.
STATUS_BAD	= Check the value of the DDMELNK data delivery key.
STUCK	= Check for stuck out-of-service MEPIPE/MELNK states.
TID	= Check for existence of unexpected timer indexes.
TIMERLST	= Check for existence of a pending timer for each valid timer index.
UNEQUIP	= Check MEPIPE/MELNK equipage state in dynamic data against static data (when there is an MEPIPE/MELNK tuple in static data).
ZAP	= Check for system interrupt handling.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ACTION

Error Code:	ACTION
Error Description:	Error is due to one of the following: 0 = The MELNK action is set to SMELASSW but the MEPIPE action does not match. 1 = The MELNK action is invalid. 2 = The MELNK recovery action is invalid. 3 = The MEPIPE action is invalid. 4 = The MEPIPE recovery action is invalid.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 = The audit clears the MELNK recovery action and recovery count fields.

	3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 = The audit clears the MEPIPE recovery action and recovery count fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3 or 4 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 2 = MELNK recovery action (SMelstat[mepipe].elnk[melnk].rcvy_act). 3 = MEPIPE action (SMelstat[mepipe].action). 4 = MEPIPE recovery action (SMelstat[mepipe].rcvy_act).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE action (SMelstat[mepipe].action). 1 and 2 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 3 and 4 = MEPIPE state (SMelstat[mepipe].state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 2 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT
Error Description:	The audit did not find the expected number of equipped child MELNKs.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data.
Manual Action Required:	None.

Dump Description:	
<i>Error Address =</i>	0 = The MEPIPE is not in the INIT state. 1 = The MEPIPE is in the INIT state.
<i>Bad Data =</i>	The number of equipped MELNKs found in MELNK dynamic data.
<i>Good Data =</i>	The number of equipped MELNKs found in MEPIPE dynamic data (SMelstat[mepipe].eq elnks).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	Error is due to one of the following: 0 = The MELNK recovery timer type field is invalid. 1 = The MEPIPE recovery timer type field is invalid.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 1 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows:

	0 = The MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 1 = The MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype).
<i>Good Data</i> =	Corresponds to the Error Description as follows: 0 = The MELNK recovery timer index (SMelstat[mepipe].elnk[melnk].rcvy_timer). 1 = The MEPIPE recovery timer index (SMelstat[mepipe].rcvy_timer).
<i>Logical Index</i> =	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump</i> =	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 4 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	Error is due to one of the following: 0 = The value of the DDRTCDELNK data delivery (DD) key is inconsistent with MELNK status (ACTIVE). 1 = The value of the DDRTCDELNK DD key is inconsistent with MELNK status (DEGRADED). 2 = The value of the DDRTCDELNK DD key is inconsistent with MELNK status (OOS).
Possible Error Effect:	Status information that controls whether real time billing data should be sent to the global billing SM is incorrect. Billing data may be inappropriately buffered (or not buffered) in an originating SM's real time billing memory (RTBM).
Corrective Action Taken:	The audit corrects the DD key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data</i> =	The incorrect value of the DD key. Corresponds to the Error Description as follows: 0 and 1 = DDOFFNORMAL. 2 = DDNORMAL.
<i>Good Data</i> =	The correct value of the DD key. Corresponds to the Error Description as follows: 0 and 1 = DDNORMAL. 2 = DDOFFNORMAL.
<i>Logical Index</i> =	A four byte value indicating the MEPIPE and MELNK numbers associated with

	the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 5 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	Error is due to one of the following: 0 = The MEPIPE is in a valid equipped state and there is a tuple in the RLENETPIPE relation, but there is an inconsistency between static and dynamic data (related to MHEIB and/or hub addresses). 1 = The MELNK is in a valid equipped state and there is a tuple in the RLENETLINK relation, but there is an inconsistency between static and dynamic data (related to far end internet protocol [IP] and hardware [HW] addresses, protocol information, or MELNK service type). 2 = Static data in the RLENETLINK relation indicates that this MELNK should be simplex but dynamic data does not agree.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data. 1 = The audit re-initializes MELNK dynamic data based on information derived from static ODD. If this information indicates that the MELNK should be equipped, the audit asks ELSP to restore it to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data. 2 = The audit corrects the dynamic data and then asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE state (SMelstat[mepipe].state).

	1 = Identification of data found to be inconsistent: 1 = Far-end IP address. 2 = Far-end HW address. 3 = Protocol/service data. 4 = Far-end IP address and protocol/service data. 5 = Far-end HW address and protocol/service data. 2 = MELNK address number (SMelstat[mepipe].elnk[melnk].addrnum).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = Not applicable. 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 2 = Correct MELNK address number (SMEL0ADDR).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	Corresponds to the Error Description as follows: 0 = Contents of RLENETPIPE tuple. 1 and 2 = Contents of RLENETLINK tuple.

Table 6 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	Error is due to one of the following: 0 and 2 = The MEPIPE state is invalid. 1 = The MELNK state is invalid.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows:

	0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data</i> =	Corresponds to the Error Description as follows: 0 and 2 = MEPIPE state (SMelstat[mepipe].state). 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state).
<i>Good Data</i> =	Corresponds to the Error Description as follows: 0 = Not applicable. 1 = MEPIPE state (SMelstat[mepipe].state). 2 = Number of equipped child MELNKs (SMelstat[mepipe].eq_elinks).
<i>Logical Index</i> =	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1</i> =	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2</i> =	Corresponds to the Error Description as follows: 0 = Contents of RLENETPIPE tuple. 1 = Contents of RLENETLINK tuple. 2 = Not applicable.

Table 7 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	Error is due to one of the following: 0 = An MELNK action is in progress without a valid index for an associated timer. 1 = An MELNK recovery timer should be pending but there is no valid timer index. 2 = An MEPIPE action is in progress without a valid index for an associated timer. 3 = An MEPIPE recovery timer should be pending but there is no valid timer index. 4 = An MCTSI/MH long term recovery timer should be pending but there is no valid timer index. 5 = An MCTSI/MH short term recovery timer should be pending but there is no valid timer index.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using

	asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 and 5 = The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3, 4 or 5 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action timer (SMelstat[mepipe].elnk[melnk].timer). 1 = MELNK recovery timer (SMelstat[mepipe].elnk[melnk].rcvy_timer). 2 = MEPIPE action timer (SMelstat[mepipe].timer). 3 = MEPIPE recovery timer (SMelstat[mepipe].rcvy_timer). 4 = MCTSI/MH long-term recovery timer index (SMelmcesc.longt_tid). 5 = MCTSI/MH short-term recovery timer index (SMelmcesc.shortt_tid).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 1 = MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 2 = MEPIPE action (SMelstat[mepipe].action). 3 = MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype). 4 and 5 = MCTSI/MH recovery/escalation state (SMelmcesc.state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with

	the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	Corresponds to the Error Description as follows: 0, 1, 2 and 3 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data). 4 and 5 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 8 Error Code: LNKS_MISSING

Error Code:	LNKS_MISSING
Error Description:	The MEPIPE is in the INIT state but the audit did not find any child MELNK being restored.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The audit then asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	MEPIPE state (SMelstat[mepipe].state).
<i>Bad Data =</i>	Number of child MELNKs that are being restored (0).
<i>Good Data =</i>	Number of equipped child MELNKs (SMelstat[mepipe].eq_elinks).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 9 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	Error is due to one of the following: 0 = MELNK dynamic data indicates that a manual request is in progress but ELSP's manual request buffer is idle. 1 = MELNK dynamic data indicates that no manual request is in progress, but ELSP's manual request buffer indicates that there should be.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit clears the MELNK's manual request indicator.

	1 = The audit clears the usage indicator in the manual request buffer.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Corresponds to the Error Description as follows: 0 or 1 = The number of the Error Description associated with this error report.
<i>Bad Data</i> =	The manual request buffer usage indicator (SMelmrbuf.in_use).
<i>Good Data</i> =	Corresponds to the Error Description as follows: 0 = MELNK manual request indicator (SMelstat[mepipe].elnk[melnk].man_req). 1 = Not applicable.
<i>Logical Index</i> =	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump</i> =	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 10 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Error is due to one of the following: 0 = The MEPIPE is in an equipped state, but the associated RLENETPIPE tuple is not present. 1 = No RLENETLINK tuple exists for the MELNK, but MELNK dynamic data is not consistent with an unequipped MELNK. 2 = The MEPIPE is in the unequipped state, but other dynamic MEPIPE data is not consistent with an unequipped MEPIPE.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 2 = The audit updates dynamic data to indicate that the MEPIPE and child MELNKs are not equipped. 1 = The audit updates dynamic data to indicate that the MELNK is not equipped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data</i> =	Corresponds to the Error Description as follows:

	0 = MEPIPE state (SMelstat[mepipe].state). 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 2 = Number of equipped child MELNKs (SMelstat[mepipe].eq_elinks).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = The UNEQ state. 2 = MEPIPE state (SMelstat[mepipe].state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 11 Error Code: PARENT

Error Code:	PARENT
Error Description:	Error is due to one of the following: 0 = An MELNK is in a state that requires an ACTIVE parent MEPIPE, but the MEPIPE is not ACTIVE. 1 = An MELNK is in the out-of-service family-of-equipment (OOSFE) state, but the parent MEPIPE is ACTIVE. 2 = The MEPIPE state is not consistent with the number of manually removed child MELNKs. If all child MELNKs are in the OOSMAN state, the MEPIPE should also be OOSMAN. 3 = The MEPIPE is in a stable OOS state, but a child MELNK is in the out-of-service due to fault recovery (OOSFLT) state. This state should not have been preserved when the MEPIPE went OOS.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually

	removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2 or 3 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 2 and 3 = MEPIPE state (SMelstat[mepipe].state).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = MEPIPE state (SMelstat[mepipe].state). 2 = Number of child MELNKs in the OOSMAN state. 3 = Number of child MELNKs in the OOSFLT state.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 12 Error Code: PORT_BAD

Error Code:	PORT_BAD
Error Description:	The error is due to one of the following: 1 = Incorrect in-service port count for an ACTIVE MELNK. 2 = Incorrect in-service port count for a DEGRADED MELNK. 3 = Incorrect in-service port count for an OOS MELNK.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	MELNK state (SMelstat[mepipe].elnk[melnk].state).
<i>Bad Data =</i>	Actual MELNK in-service port count (SMelstat[mepipe].elnk[melnk].act_ports).
<i>Good Data =</i>	Expected MELNK in-service port count.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 13 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	Error is due to one of the following: 0 = ELSP's MCTSI/MH recovery strategy is in an invalid state. 1 = ELSP's MCTSI/MH recovery strategy is active, even though there is an in-service MELNK and the safe interval timer is not running. 2 = ELSP's MCTSI/MH recovery strategy is active, even though all MELNKs are manually removed from service.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	MCTSI/MH recovery/escalation state (SMelmcesc.state).
<i>Good Data =</i>	Not applicable.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	Corresponds to the Error Description as follows: 0 and 2 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data). 1 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	Corresponds to the Error Description as follows: 0 and 2 = Not applicable. 1 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 14 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	The value of the DDMELNK data delivery (DD) key is incorrect.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed.
Corrective Action Taken:	The audit corrects the DD key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not applicable.
<i>Bad Data =</i>	Incorrect value of the DDMELNK key.
<i>Good Data =</i>	Correct value of the DDMELNK key.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[0] (MEPIPE/MELNK dynamic data for MEPIPE 0).

Table 15 Error Code: STUCK

Error Code:	STUCK
Error Description:	Error is due to one of the following: 0 = An MELNK is stuck in the stable OOSFLT state without a recovery timer running. 1 = An MEPIPE is stuck in the stable OOSFLT or OOSL1 state without a recovery timer running.
Possible Error Effect:	MELNK functionality is lost while the MEPIPE/MELNK is stuck in an OOS state. Since no automatic recovery action is pending, recovery will be delayed until the problem is corrected or a manual MELNK restore is requested.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The audit then asks ELSP to restore the MELNK to service. 1 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 1 = MEPIPE state (SMelstat[mepipe].state).

<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE state (SMelstat[mepipe].state). 1 = Last MEPIPE recovery action (SMelstat[mepipe].rcvy_act).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 16 Error Code: TID

Error Code:	TID
Error Description:	Error is due to one of the following: 0 = No MELNK action requiring a timer is pending but a timer is apparently running. 1 = No MELNK recovery timer should be pending but a timer is apparently running. 2 = No MEPIPE action requiring a timer is pending but a timer is apparently running. 3 = No MEPIPE recovery timer should be pending but a timer is apparently running. 4 = ELSP's MCTSI/MH recovery strategy is in the short-term recovery state but there is apparently a long-term recovery timer running. 5 = ELSP's MCTSI/MH recovery strategy is not active, but there is apparently at least one MCTSI/MH recovery timer running.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in

	the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 = The long term recovery timer is canceled. 5 = The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3, 4 or 5 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action timer (SMelstat[mepipe].elnk[melnk].timer). 1 = MELNK recovery timer (SMelstat[mepipe].elnk[melnk].rcvy_timer). 2 = MEPIPE action timer (SMelstat[mepipe].timer). 3 = MEPIPE recovery timer (SMelstat[mepipe].rcvy_timer). 4 = MCTSI/MH recovery long-term timer index (SMelmcesc.longt_tidx). 5 = MCTSI/MH recovery short-term timer index (SMelmcesc.shortt_tidx).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 1 = MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 2 = MEPIPE action (SMelstat[mepipe].action). 3 = MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype). 4 = MCTSI/MH recovery/escalation state (SMelmcesc.state). 5 = MCTSI/MH long-term recovery timer index (SMelmcesc.longt_tidx).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	Corresponds to the Error Description as follows: 0, 1, 2 and 3 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data). 4 and 5 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 17 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	Error is due to one of the following: 0 = An MELNK action requiring a timer is pending but no timer is running. 1 = An MELNK recovery timer should be pending but no timer is running. 2 = An MEPIPE action requiring a timer is pending but no timer is running. 3 = An MEPIPE recovery timer should be pending but no timer is running. 4 = A long-term recovery timer should be pending but no timer is running. 5 = A short-term recovery timer should be pending but no timer is running.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 and 5 = The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3, 4 or 5 = Indicates error description, error effect, corrective action and dump.
<i>Bad Data =</i>	Timer tag value of the missing timer.
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 1 = MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype).

	2 = MEPIPE action (SMelstat[mepipe].action). 3 = MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype). 4 and 5 = MCTSI/MH recovery/escalation state (SMelmcesc.state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	Corresponds to the Error Description as follows: 0, 1, 2 and 3 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data). 4 and 5 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 18 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	Error is due to one of the following: 0 = An MELNK is in the unequipped state but there is an associated tuple in RLENETLINK. 1 = An MEPIPE is in the unequipped state but there is an equipped child MELNK.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data. 1 = The audit updates dynamic data to indicate that the MEPIPE and all child MELNKs are unequipped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = Indicates error description, error effect, corrective action and dump.
<i>Bad Data =</i>	MELNK state (SMelstat[mepipe].elnk[melnk].state).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = Not applicable.

	1 = MEPIPE state (SMelstat[mepipe].state).
<i>Logical Index</i> =	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1</i> =	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2</i> =	Corresponds to the Error Description as follows: 0 = Contents of RLENETLINK tuple. 1 = Not applicable.

Table 19 Error Code: ZAP

Error Code:	ZAP
Error Description:	This error occurs as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Re-initialize all data that could have made the audit cause a system reset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not applicable.
<i>Bad Data</i> =	Not applicable.
<i>Good Data</i> =	Not applicable.
<i>Logical Index</i> =	Not applicable.
<i>Dump</i> =	Not applicable.

MELNK-B

Software Release: 5E13 and later

1. MCTSI-BASED ETHERNET LINK (MELNK) AUDIT

The MELNK audit is responsible for checking dynamic data maintained by the ethernet link fault recovery system process (ELSP). This data includes the maintenance status of MELNKs and MCTSI-based ethernet pipes (MEPIPEs). It also includes dynamic copies of information found in the static relations RLENETPIPE and RLENETLINK. Consistency between related dynamic data structures and consistency with static ODD are both verified. All audited data is SM-2000 SMP-resident. Other software mechanisms are used to ensure that the SMP-resident data is consistent with MEPIPE/MELNK status data stored in the SM-2000 message handler.

2. DATA STRUCTURES

The following static data is used by the MELNK audit:

RLENETPIPE = Database relation containing MEPIPE data.

RLENETLINK = Database relation containing MELNK data.

Glprtc = Global parameter identifying the global billing SM.

The following dynamic data is checked by the MELNK audit:

SMelstat[] = Dynamic data maintained by ELSP for MEPIPEs and MELNKs.

SMelmcsc = Data related to ELSP's MCTSI/MH escalation strategy.

SMelmrbuf = Buffer for data about in progress manual MELNK remove/restore requests.

For details on the fields contained in the RLENETPIPE and RLENETLINK relations, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ACTION

Error Code:	ACTION
Error Description:	Error is due to one of the following: 0 = The MELNK action is set to SMELASSW but the MEPIPE action does not match. 1 = The MELNK action is invalid. 2 = The MELNK recovery action is invalid. 3 = The MEPIPE action is invalid.

	4 = The MEPIPE recovery action is invalid.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 = The audit clears the MELNK recovery action and recovery count fields. 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 = The audit clears the MEPIPE recovery action and recovery count fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3 or 4 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 2 = MELNK recovery action (SMelstat[mepipe].elnk[melnk].rcvy_act). 3 = MEPIPE action (SMelstat[mepipe].action). 4 = MEPIPE recovery action (SMelstat[mepipe].rcvy_act).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE action (SMelstat[mepipe].action). 1 and 2 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 3 and 4 = MEPIPE state (SMelstat[mepipe].state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The

	MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 2 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT
Error Description:	The audit did not find the expected number of equipped child MELNKs.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0 = The MEPIPE is not in the INIT state. 1 = The MEPIPE is in the INIT state.
<i>Bad Data =</i>	The number of equipped MELNKs found in MELNK dynamic data.
<i>Good Data =</i>	The number of equipped MELNKs found in MEPIPE dynamic data (SMelstat[mepipe].eq elnks).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	Error is due to one of the following: 0 = The MELNK recovery timer type field is invalid. 1 = The MEPIPE recovery timer type field is invalid.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service.

	1 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = The MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 1 = The MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = The MELNK recovery timer index (SMelstat[mepipe].elnk[melnk].rcvy_timer). 1 = The MEPIPE recovery timer index (SMelstat[mepipe].rcvy_timer).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 4 Error Code: BILLFLG

Error Code:	BILLFLG
Error Description:	Error is due to one of the following: 0 = The value of the DDRTCDELNK data delivery (DD) key is inconsistent with MELNK status (ACTIVE). 1 = The value of the DDRTCDELNK DD key is inconsistent with MELNK status (DEGRADED). 2 = The value of the DDRTCDELNK DD key is inconsistent with MELNK status (OOS).
Possible Error Effect:	Status information that controls whether real time billing data should be sent to the global billing SM is incorrect. Billing data may be inappropriately buffered (or not buffered) in an originating SM's real time billing memory (RTBM).
Corrective Action Taken:	The audit corrects the DD key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	The incorrect value of the DD key. Corresponds to the Error Description as follows: 0 and 1 = DDOFFNORMAL.

	2 = DDNORMAL.
<i>Good Data =</i>	The correct value of the DD key. Corresponds to the Error Description as follows: 0 and 1 = DDNORMAL. 2 = DDOFFNORMAL.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 5 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	Error is due to one of the following: 0 = The MEPIPE is in a valid equipped state and there is a tuple in the RLENETPIPE relation, but there is an inconsistency between static and dynamic data (related to MHEIB and/or hub addresses). 1 = The MELNK is in a valid equipped state and there is a tuple in the RLENETLINK relation, but there is an inconsistency between static and dynamic data (related to far end internet protocol [IP] and hardware [HW] addresses, protocol information, or MELNK service type). 2 = Static data in the RLENETLINK relation indicates that this MELNK should be simplex but dynamic data does not agree.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data. 1 = The audit re-initializes MELNK dynamic data based on information derived from static ODD. If this information indicates that the MELNK should be equipped, the audit asks ELSP to restore it to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data. 2 = The audit corrects the dynamic data and then asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows:

	0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE state (SMelstat[mepipe].state). 1 = Identification of data found to be inconsistent: 1 = Far-end IP address. 2 = Far-end HW address. 3 = Protocol/service data. 4 = Far-end IP address and protocol/service data. 5 = Far-end HW address and protocol/service data. 2 = MELNK address number (SMelstat[mepipe].elnk[melnk].addrnum).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = Not applicable. 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 2 = Correct MELNK address number (SMEL0ADDR).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	Corresponds to the Error Description as follows: 0 = Contents of RLENETPIPE tuple. 1 and 2 = Contents of RLENETLINK tuple.

Table 6 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	Error is due to one of the following: 0 and 2 = The MEPIPE state is invalid. 1 = The MELNK state is invalid.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with

	SMP-resident static and dynamic data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 and 2 = MEPIPE state (SMelstat[mepipe].state). 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = Not applicable. 1 = MEPIPE state (SMelstat[mepipe].state). 2 = Number of equipped child MELNKs (SMelstat[mepipe].eq_elinks).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	Corresponds to the Error Description as follows: 0 = Contents of RLENETPIPE tuple. 1 = Contents of RLENETLINK tuple. 2 = Not applicable.

Table 7 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	Error is due to one of the following: 0 = An MELNK action is in progress without a valid index for an associated timer. 1 = An MELNK recovery timer should be pending but there is no valid timer index. 2 = An MEPIPE action is in progress without a valid index for an associated timer. 3 = An MEPIPE recovery timer should be pending but there is no valid timer index. 4 = An MCTSI/MH long term recovery timer should be pending but there is no valid timer index. 5 = An MCTSI/MH short term recovery timer should be pending but there is no valid timer index.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.

Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 and 5 = The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3, 4 or 5 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action timer (SMelstat[mepipe].elnk[melnk].timer). 1 = MELNK recovery timer (SMelstat[mepipe].elnk[melnk].rcvy_timer). 2 = MEPIPE action timer (SMelstat[mepipe].timer). 3 = MEPIPE recovery timer (SMelstat[mepipe].rcvy_timer). 4 = MCTSI/MH long-term recovery timer index (SMelmcesc.longt_tidx). 5 = MCTSI/MH short-term recovery timer index (SMelmcesc.shortt_tidx).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 1 = MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 2 = MEPIPE action (SMelstat[mepipe].action). 3 = MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype). 4 and 5 = MCTSI/MH recovery/escalation state (SMelmcesc.state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The

	MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	Corresponds to the Error Description as follows: 0, 1, 2 and 3 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data). 4 and 5 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 8 Error Code: LNKS_MISSING

Error Code:	LNKS_MISSING
Error Description:	The MEPIPE is in the INIT state but the audit did not find any child MELNK being restored.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The audit then asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	MEPIPE state (SMelstat[mepipe].state).
<i>Bad Data =</i>	Number of child MELNKs that are being restored (0).
<i>Good Data =</i>	Number of equipped child MELNKs (SMelstat[mepipe].eq_elnks).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 9 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	Error is due to one of the following: 0 = MELNK dynamic data indicates that a manual request is in progress but ELSP's manual request buffer is idle. 1 = MELNK dynamic data indicates that no manual request is in progress, but ELSP's manual request buffer indicates that there should be.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit clears the MELNK's manual request indicator. 1 = The audit clears the usage indicator in the manual request buffer.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = The number of the Error Description associated with this error report.

<i>Bad Data =</i>	The manual request buffer usage indicator (SMelmrbuf.in_use).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK manual request indicator (SMelstat[mepipe].elnk[melnk].man_req). 1 = Not applicable.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 10 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Error is due to one of the following: 0 = The MEPIPE is in an equipped state, but the associated RLENETPIPE tuple is not present. 1 = No RLENETLINK tuple exists for the MELNK, but MELNK dynamic data is not consistent with an unequipped MELNK. 2 = The MEPIPE is in the unequipped state, but other dynamic MEPIPE data is not consistent with an unequipped MEPIPE.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 2 = The audit updates dynamic data to indicate that the MEPIPE and child MELNKs are not equipped. 1 = The audit updates dynamic data to indicate that the MELNK is not equipped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE state (SMelstat[mepipe].state). 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 2 = Number of equipped child MELNKs (SMelstat[mepipe].eq_elinks).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = The UNEQ state. 2 = MEPIPE state (SMelstat[mepipe].state).

<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 11 Error Code: PARENT

Error Code:	PARENT
Error Description:	Error is due to one of the following: 0 = An MELNK is in a state that requires an ACTIVE parent MEPIPE, but the MEPIPE is not ACTIVE. 1 = An MELNK is in the out-of-service family-of-equipment (OOSFE) state, but the parent MEPIPE is ACTIVE. 2 = The MEPIPE state is not consistent with the number of manually removed child MELNKs. If all child MELNKs are in the OOSMAN state, the MEPIPE should also be OOSMAN. 3 = The MEPIPE is in a stable OOS state, but a child MELNK is in the out-of-service due to fault recovery (OOSFLT) state. This state should not have been preserved when the MEPIPE went OOS.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2 or 3 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows:

	0 and 1 = MELNK state (SMelstat[mepipe].elnk[melnk].state).
	2 and 3 = MEPIPE state (SMelstat[mepipe].state).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 and 1 = MEPIPE state (SMelstat[mepipe].state). 2 = Number of child MELNKs in the OOSMAN state. 3 = Number of child MELNKs in the OOSFLT state.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 12 Error Code: PORT_BAD

Error Code:	PORT_BAD
Error Description:	The error is due to one of the following: 1 = Incorrect in-service port count for an ACTIVE MELNK. 2 = Incorrect in-service port count for a DEGRADED MELNK. 3 = Incorrect in-service port count for an OOS MELNK.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	MELNK state (SMelstat[mepipe].elnk[melnk].state).
<i>Bad Data =</i>	Actual MELNK in-service port count (SMelstat[mepipe].elnk[melnk].act_ports).
<i>Good Data =</i>	Expected MELNK in-service port count.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).

Table 13 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	Error is due to one of the following: 0 = ELSP's MCTSI/MH recovery strategy is in an invalid state.

	1 = ELSP's MCTSI/MH recovery strategy is active, even though there is an in-service MELNK and the safe interval timer is not running. 2 = ELSP's MCTSI/MH recovery strategy is active, even though all MELNKs are manually removed from service.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows:
	0, 1 or 2 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	MCTSI/MH recovery/escalation state (SMelmcesc.state).
<i>Good Data =</i>	Not applicable.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	Corresponds to the Error Description as follows:
	0 and 2 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).
	1 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	Corresponds to the Error Description as follows:
	0 and 2 = Not applicable.
	1 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 14 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	The value of the DDMELNK data delivery (DD) key is incorrect.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed.
Corrective Action Taken:	The audit corrects the DD key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not applicable.
<i>Bad Data =</i>	Incorrect value of the DDMELNK key.
<i>Good Data =</i>	Correct value of the DDMELNK key.
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	The contents of SMelstat[0] (MEPIPE/MELNK dynamic data for MEPIPE 0).

Table 15 Error Code: STUCK

Error Code:	STUCK
Error Description:	Error is due to one of the following:

	0 = An MELNK is stuck in the stable OOSFLT state without a recovery timer running. 1 = An MEPIPE is stuck in the stable OOSFLT or OOSL1 state without a recovery timer running.
Possible Error Effect:	MELNK functionality is lost while the MEPIPE/MELNK is stuck in an OOS state. Since no automatic recovery action is pending, recovery will be delayed until the problem is corrected or a manual MELNK restore is requested.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The audit then asks ELSP to restore the MELNK to service. 1 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK state (SMelstat[mepipe].elnk[melnk].state). 1 = MEPIPE state (SMelstat[mepipe].state).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MEPIPE state (SMelstat[mepipe].state). 1 = Last MEPIPE recovery action (SMelstat[mepipe].rcvy_act).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 16 Error Code: TID

Error Code:	TID
Error Description:	Error is due to one of the following: 0 = No MELNK action requiring a timer is pending but a timer is apparently running. 1 = No MELNK recovery timer should be pending but a timer is apparently running. 2 = No MEPIPE action requiring a timer is pending but a timer is apparently running.

	3 = No MEPIPE recovery timer should be pending but a timer is apparently running. 4 = ELSP's MCTSI/MH recovery strategy is in the short-term recovery state but there is apparently a long-term recovery timer running. 5 = ELSP's MCTSI/MH recovery strategy is not active, but there is apparently at least one MCTSI/MH recovery timer running.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service. 2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 = The long term recovery timer is canceled. 5 = The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3, 4 or 5 = The number of the Error Description associated with this error report.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action timer (SMelstat[mepipe].elnk[melnk].timer). 1 = MELNK recovery timer (SMelstat[mepipe].elnk[melnk].rcvy_timer). 2 = MEPIPE action timer (SMelstat[mepipe].timer). 3 = MEPIPE recovery timer (SMelstat[mepipe].rcvy_timer). 4 = MCTSI/MH recovery long-term timer index (SMelmcesc.longt_tidx).

	5 = MCTSI/MH recovery short-term timer index (SMelmcesc.shortt_tidx).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 1 = MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 2 = MEPIPE action (SMelstat[mepipe].action). 3 = MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype). 4 = MCTSI/MH recovery/escalation state (SMelmcesc.state). 5 = MCTSI/MH long-term recovery timer index (SMelmcesc.longt_tidx).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	Corresponds to the Error Description as follows: 0, 1, 2 and 3 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data). 4 and 5 = The contents of SMelmcesc (MCTSI/MH recovery/escalation data).

Table 17 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	Error is due to one of the following: 0 = An MELNK action requiring a timer is pending but no timer is running. 1 = An MELNK recovery timer should be pending but no timer is running. 2 = An MEPIPE action requiring a timer is pending but no timer is running. 3 = An MEPIPE recovery timer should be pending but no timer is running. 4 = A long-term recovery timer should be pending but no timer is running. 5 = A short-term recovery timer should be pending but no timer is running.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.
Corrective Action Taken:	Corresponds to the Error Description as follows: 0 and 1 = The audit re-initializes MELNK dynamic data that is not derived from static ODD, ensuring that the MELNK is in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MELNK in the OOSMAN state (because it was manually removed) or the OOSFE state (because the parent MEPIPE was removed due to hardware errors), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MELNK to service.

	2 and 3 = The audit re-initializes all MEPIPE (and child MELNK) dynamic data that is not derived from static ODD, ensuring that the MEPIPE and MELNK(s) are in a stable out-of-service state. The OOSMAN (manually removed) state is preserved. If this data initialization leaves the MEPIPE in the OOSMAN state (because all child MELNKs were manually removed), the audit asks ELSP to initiate an MEPIPE/MELNK status audit to ensure data consistency between the SMP and MH. Otherwise, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. 4 and 5 = The audit causes ELSP to exit MCTSI/MH recovery and re-initialize the associated data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0, 1, 2, 3, 4 or 5 = Indicates error description, error effect, corrective action and dump.
<i>Bad Data =</i>	Timer tag value of the missing timer.
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = MELNK action (SMelstat[mepipe].elnk[melnk].action). 1 = MELNK recovery timer type (SMelstat[mepipe].elnk[melnk].rcvy_ttype). 2 = MEPIPE action (SMelstat[mepipe].action). 3 = MEPIPE recovery timer type (SMelstat[mepipe].rcvy_ttype). 4 and 5 = MCTSI/MH recovery/escalation state (SMelmcsc.state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump =</i>	Corresponds to the Error Description as follows: 0, 1, 2 and 3 = The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data). 4 and 5 = The contents of SMelmcsc (MCTSI/MH recovery/escalation data).

Table 18 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	Error is due to one of the following: 0 = An MELNK is in the unequipped state but there is an associated tuple in RLENETLINK. 1 = An MEPIPE is in the unequipped state but there is an equipped child MELNK.
Possible Error Effect:	MEPIPE/MELNK maintenance data is incorrect or inconsistent. As a result, incorrect status may be reported and/or displayed. Attempts to configure the MELNKs may fail. MEPIPE/MELNK data problems may be reported using asserts and REPT TRBL output messages. If an MEPIPE or MELNK is out-of-service, recovery may be delayed until the problem is corrected.

Corrective Action Taken:	Corresponds to the Error Description as follows: 0 = The audit re-initializes MEPIPE (and child MELNK) dynamic data based on information derived from static ODD. If this information indicates that the MEPIPE should be equipped, the audit asks ELSP to restore the MEPIPE and all child MELNKs to service. Otherwise, the audit just refreshes MH-resident data to ensure consistency with SMP-resident static and dynamic data. 1 = The audit updates dynamic data to indicate that the MEPIPE and all child MELNKs are unequipped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Corresponds to the Error Description as follows: 0 or 1 = Indicates error description, error effect, corrective action and dump.
<i>Bad Data =</i>	MELNK state (SMelstat[mepipe].elnk[melnk].state).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 0 = Not applicable. 1 = MEPIPE state (SMelstat[mepipe].state).
<i>Logical Index =</i>	A four byte value indicating the MEPIPE and MELNK numbers associated with the error. The MEPIPE number is reported in the most significant two bytes and may be SMELNULLEPIPE (0x3) if the error is not associated with a specific MEPIPE. The MELNK number is reported in the least significant two bytes and may be SMELNULLELNK (0xf) if the error is not associated with a specific MELNK.
<i>Dump 1 =</i>	The contents of SMelstat[mepipe] (MEPIPE/MELNK dynamic data).
<i>Dump 2 =</i>	Corresponds to the Error Description as follows: 0 = Contents of RLENETLINK tuple. 1 = Not applicable.

Table 19 Error Code: ZAP

Error Code:	ZAP
Error Description:	This error occurs as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Re-initialize all data that could have made the audit cause a system reset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not applicable.
<i>Bad Data =</i>	Not applicable.
<i>Good Data =</i>	Not applicable.
<i>Logical Index =</i>	Not applicable.
<i>Dump =</i>	Not applicable.

MEM

Software Release: 5E10 and later

1. MULTILINE HUNT GROUP MEMBER STATUS (MEM) AUDIT

To process a call during the stand alone (SA), the routing and terminal allocation (RTA) routing process in the remote switching module (RSM) and in the optically remote module (ORM) must determine which ports are available for terminating calls. Information about a small (up to 64 members) multi-line hunt group (MLHG) is kept on bit maps, with each bit representing a member of the MLHG. The bit map is included in each tuple in the MLHG status relation (RLMEM). When a member is populated for a MLHG, the corresponding bit must be set to "1" to indicate that the member is active. If the member is not active, the bit must be set to "0". The relation RLMEM is created during the SA condition in a RSM/ORM for all global MLHG members for which that switching module (SM) is the hunting SM and for all global members that have ports in that SM.

The MLHG member status audit (MEM) checks tuples in the dynamic relation RLMEM for an extra tuple. It also performs a check on the status bit for each group. The MEM audit verifies with the static relations RLGROUP_PORT/RLPSGRP_PRT that for each MLHG member that is set to "1", is a valid active member.

In the RSM/ORM, the audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode. This audit is scheduled only during the SA mode.

2. DATA STRUCTURES

The data structures checked by the MEM audit are the group and member number to the port relation (RLGROUP_PORT), and the packet switching multiline/trunk group and member number to the packet switching port relation (RLPSGP_PRT).

The RLGROUP_PORT and RLPSGP_PRT relations are static (write-protected) relations that contain information about all global MLHG ports. For every member of the global MLHG, a rIGROUP_PORT/rIPSGP_PRT tuple must exist in the SM which owns the port, in the designated hunting SM, and in the designated routing SM. The rIPSGP_PRT is populated only for RTH_PSUCD and RTH_PSLGUCD hunt types (packet switching ports), while rIGROUP_PORT is populated for all other MLHG hunt types.

3. AUDIT CHECKS

The MEM audit uses the static driver (DRSTATIC) to step through the RLMEM relation and sequentially retrieve tuples. The DRSTATIC driver invokes a semantic check function that verifies each rIMEM tuple against the rIGROUP_PORT/rIPSGP_PRT tuples by using the key (group number) of rIMEM to read the static relations. The audit looks specifically for members that are marked as active in the RLMEM relation, but for which there is no corresponding tuple in the static relation.

4. ERROR HANDLING

Excess tuples, tuples for which all members are not active, are deleted from the relation. Bits that are set to "1" (active) for which there are no MLHG members associated with it, are corrected to "0" (not active). If a tuple of the member status relation contains an invalid data in its key, the key is corrected.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP

Error Description:	All members of audited rMEM tuple are invalid (0) and therefore the tuple should not exist.
Possible Error Effect:	An ASSERT will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of pointer to actual tuple.
<i>Bad Data</i> =	Key of the bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the bad tuple.
<i>Dump</i> =	rMEM tuple.

Table 2 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	Audit cannot find static rRTMHG tuple using dynamic rMEM key.
Possible Error Effect:	An error will report if this tuple is not found.
Corrective Action Taken:	The dynamic rMEM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rRT MHG tuple.
<i>Bad Data</i> =	The key of rMEM tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 3 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	The corresponding member in the dynamic relations rMEM was incorrectly initialized to valid ("1").
Possible Error Effect:	Initialization was wrong and calls will fail to that member.
Corrective Action Taken:	The member will be set to invalid ("0").
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rMEM tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	0 (invalid member).
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rMEM tuple.

Table 4 Error Code: KEY

Error Code:	KEY
Error Description:	A DAGETMEM macro returns a pointer whose key does not match the tuple buffer key of rMEM tuple.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rLMEM tuple having the wrong key.
<i>Bad Data</i> =	The key value of pointer to tuple.
<i>Good Data</i> =	The key value of tuple buffer pointer.
<i>Logical Index</i> =	The correct value of the key.
<i>Dump</i> =	The contents of the rMEM tuple with the wrong key.

Table 5 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The status of SM indicate normal and this dynamic tuple rMEM should not exist in this situation.
Possible Error Effect:	Incorrect busy/idle status may exist for this Multiline Hunt Group.
Corrective Action Taken:	The dynamic rLGEM tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rMEM tuple buffer.

<i>Bad Data</i> =	The key of rILGMEM tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 6 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type as indicated in rIRT_MHG tuple for the group number in rIMEM relation.
Possible Error Effect:	An error will report if this group's hunt type is not of the type used in rIMEM (that is, RTH_REGULAR, RTH_CIRCULAR, RTH_UCD, RTH_PSUCD, RTH_PPSMHG, or RTH_ISAT).
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIMEM tuple.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of rIMEM tuple used to access rIRT_MHG tuple.
<i>Dump</i> =	rIRTMHG tuple.

MEMMAN-A

Software Release: 5E10 - 5E12

1. MEMORY MANAGEMENT (MEMMAN) AUDIT

The software demand paging (SDP) package, a database memory manager, allocates memory within the write-protected office-dependent database (ODD), disk ODD, and the unprotected database based on the availability information stored in bit maps. The SDP bit maps keep track of the space within the ODD, disk ODD, and unprotected database. Each bit corresponds to a block of memory. The runtime access dictionary entries contain pointers (block IDs) to data pages in the ODD, disk ODD, and unprotected database, and to head tables of relations. One access dictionary entry is allocated for each version of a relation in the database. Each entry contains access information for the associated relation version, including a head table index and information about the relation. In Figure 1, for example, the block ID of the entry for relation 1 points to a data page in the ODD; relation 1 is therefore a static one-level indexed relation. The data pages for static two-level indexed, static hash, and static sequential relations such as relation 2 also reside in the ODD. Dynamic relations have their head tables in the ODD and their data pages in unprotected memory. Relation 3 is a dynamic hash or dynamic indexed relation. The indexed head table contains indices that define the data pages that make up the relation version. Relation 4 is a static three-level indexed relation. The head table has indices to all the intermediate data pages (IDPs). The IDPs have indices to data pages.

Memory management of the ODD must ensure that no data pages are lost to the system. The memory management audit (MEMMAN) verifies that the software demand paging package record of memory block usage corresponds to the actual memory blocks in use. If SDP has an unused memory block marked as used, the block is unavailable or lost. If SDP has a used memory block marked as unused, it may overwrite the existing data with new data. The audit detects these inconsistencies and corrects the SDP bit map and internal ODD free lists as appropriate. It also performs several checks on the transaction dictionary blocks because the local log pages of update transactions are in the ODD.

The MEMMAN audit is part of the Data Base audits subgroup. It runs in the OKP and operating system for distributed switching-switching module (OSDS-SM) environments in the segmented mode.

2. DATA STRUCTURES

The memory used to store relations is organized into units called *blocks* and *pages*. A block contains 256 contiguous bytes of memory (note that the term "block" has a different meaning when used in this context than when used synonymously with "tuple" as in an access dictionary block). A page is a set of 1, 2, 4, 8, 16, 32, or 64 contiguous blocks. All the data pages that make up a given relation have the same number of blocks. The number of blocks per page is contained in the access dictionary. This information is used by the audit in conjunction with the relation's head table (if any) to determine which blocks are used by the relation.

The MEMMAN audit checks eight bit maps and two dictionaries to ensure the consistency of availability information about the ODD. The eight bitmaps are divided into two sets, one set for the memory ODD and another for the DISK ODD. The memory ODD includes the static ODD and the dynamic database. Each set has two bitmaps which are complements of each other that are used by the audit to note relation ownership of each block. Another bit map in each set, maintained by SDP, contains one bit per block that stores the allocated/available status for each block. SDP uses this bit map to allocate and deallocate blocks for use as head tables or data pages. In addition, the memory ODD portion of the audit creates two bitmaps that audit the free page lists built into the free ODD pages. Figure 1, shows the layout of system memory, including the ODD. The MEMMAN audit initializes two audit-owned bit maps based on information in the access dictionary and another two bit maps for the memory ODD free page list audit. All these bit maps are checked for consistency with each other and the actual available memory space.

The two dictionaries used by the MEMMAN audit are the runtime access dictionary and the transaction dictionary. The access dictionary contains information about all the relations within ODD. There is an access block in the access dictionary for each version of each relation; it provides the block ID of a head table in ODD. The head table

contains an array of block IDs that point to data pages. The transaction dictionary manages all transactions; the transactions are used primarily by the Recent Change/Verify and Maintenance subsystems.

3. AUDIT CHECKS

The steps of the MEMMAN audit are as follows.

- (a) In its first real-time segment, the audit initializes its four relation ownership bit maps (two relating to the memory ODD and two relating to the disk ODD). The first bit map of each set is initialized to zeros and the second to ones. The first and second bit maps are thus one's complements of each other; this setup provides an internal consistency check for the audit.
- (b) The audit uses the head table pointer of the current access dictionary block to locate the associated head table. It then uses the head table to determine which data pages or IDPs are used to store the associated relation. The number of blocks per page for the relation is in the access dictionary. This information allows the audit to determine which blocks are used to store the relation. Next, the audit compares this block usage information with the block usage information contained in the SDP bit map. There are two possible results: the status of a used block claimed by the audit agrees with the bit map, or the audit claims that a block is used and the bit map claims that it is unused. The second case is an error and the audit simply marks the bit in the SDP bit map to show the block as used. It is also possible that the bit map claims that a block is used when it is not, but the audit cannot determine this until all the access blocks have been read through. During this processing, the audit marks its bit maps for blocks that are found to be used.
- (c) The audit then performs garbage collection. Garbage results from purged processes not cleaned up by the transaction manager or other database audits. Garbage may also result from pages lost because of interrupted transactions. After processing all the access blocks, the audit compares its auxiliary bit maps to the SDP bit map. If a bit is found to be marked as used in the SDP bit map but not in the auxiliary bit map, the audit clears the bit in the SDP bit map.
- (d) The next phase of the audit concerns auditing the integrity of the free lists. The free lists are organized as follows: Each free page of a given size is placed into a linked list of other pages of that size. The forward and previous links to other blocks on the list are stored within the ODD at the page's physical location. SDP maintains a list of the head blocks of each size's list. Blocks are allocated and freed according to a Buddy Block strategy in which adjacent blocks may be coalesced or split depending on the block size requested and the current block distribution. The audit checks that each page is assigned to the appropriate list, that the linkages are consistent and that every block of memory is accounted for.
- (e) The audit then checks the usage of the tuple cache pages. These are pages of the ODD seized by the tuple cache manager to provide space for temporary copies of tuples in memory prior to committal of a transaction. The audit verifies that the internal cache bitmaps are consistent with the contents of the cache.
- (f) The audit checks the global office parameters by marking them according to information in the ODD. The audit performs the checks on one access block per real-time segment. When comparing the auxiliary and SDP bit maps, it processes two bytes of the bit map per segment. The state of the audit is saved before taking a segment break to allow the audit to resume at the correct point when control returns from OSDs.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ABID

Error Code:	ABID
Error Description:	The MEMMAN audit finds an out-of-range block ID (trn_log).
Possible Error Effect:	This update transaction loses its local log.
Corrective Action Taken:	None.
Manual Action Required:	Run the TRNDC audit.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	NA.

Table 2 Error Code: ALIGNMENT

Error Code:	ALIGNMENT
Error Description:	A free page begins on an illegal boundary for its page size. For example, a 2-block page cannot begin at block 1 but may begin at block 2 or 4.
Possible Error Effect:	Memory fragmentation.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the page in error. The removed pages are subsequently reclaimed during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the byte in the free page bitmap where the misalignment was detected.
<i>Bad Data</i> =	Incorrect block ID stored in the pages to/from link.
<i>Good Data</i> =	Block ID of the last good page on the list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 3 Error Code: BUDDY

Error Code:	BUDDY
Error Description:	The buddy page for any given available page is also free. The buddy page is defined as one which is: - the same size (for example, both 256 bytes, 512 bytes, and so on). - adjacent (for example, block 0 and block 1 for 256 byte pages). - properly aligned (for example, block 0 and block 1, block 2 and block 3, but not block 1 and block 2) to a given page. This is an error condition because it violates the constraints of the memory management algorithms.
Possible Error Effect:	This can unnecessarily fragment office-dependent data (ODD) memory. For example, if two 8K buddies exist on the 8K free list and a request for a 16-Kbyte page is made, and no page exists on the 16K free list, new memory is allocated for the ODD section before the request is satisfied. If new memory cannot be allocated, then the request cannot be satisfied.
Corrective Action Taken:	Free buddies are coalesced and put on the free lists for their combined page size.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the free buddy was located.
<i>Bad Data</i> =	Block ID of the free page's buddy.
<i>Good Data</i> =	Block ID of the original free page.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512

	list, and so forth.
<i>Dump</i> =	NA.

Table 4 Error Code: CHBKAU

Error Code:	CHBKAU
Error Description:	The audit of the tuple cache bit map (DBchbkmp) has found a cache block incorrectly marked as used. This bit map is used to mark tuple caching blocks as used or not (1 - used, 0 - not in use).
Possible Error Effect:	The cache block is unavailable for use. This is a loss of resource.
Corrective Action Taken:	The appropriate bit (which is in error) representing a cache block is set to zero (marking it as free).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of four byte section of the cache block bit map which contains the error (part of the area pointed to by DBchbkmp).
<i>Bad Data</i> =	Four byte section of cache block bit map containing error.
<i>Good Data</i> =	Four byte section of cache block bit map containing error.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	32 cache blocks corresponding to the 4 byte section of DBchbkmp which had the bad status.

Table 5 Error Code: CNVACCBID

Error Code:	CNVACCBID
Error Description:	An access dictionary block ID of any relation in the RODD is detected by MEMMAN as not being converted by the ODD BST PROCESS.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The access dictionary block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid access dictionary block ID the RODD relation.
<i>Bad Data</i> =	The invalid access dictionary block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The whole access dictionary entry for the RODD relation that has had its access dictionary block ID converted.

Table 6 Error Code: CNVHDBID

Error Code:	CNVHDBID
Error Description:	A head table block ID of a two-level or a three-level indexing relation in the RODD is detected by MEMMAN as not being converted by the ODD BST process.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The head table block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid head table block ID of a 2 or a 3-level indexing relation.
<i>Bad Data</i> =	The invalid head table block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The whole access dictionary entry for the two-level or three-level indexing relation that has had its head table block ID converted.

Table 7 Error Code: CNVIPBID

Error Code:	CNVIPBID
Error Description:	An intermediate page block ID of a three-level indexing relation in the RODD is detected by MEMMAN as not being converted by the ODD BST PROCESS.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.

Corrective Action Taken:	The intermediate page block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid intermediate page block ID of a three-level indexing relation.
<i>Bad Data</i> =	The invalid intermediate page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The whole access dictionary entry for the three-level indexing relation that has had its intermediate page converted.

Table 8 Error Code: CONFLICT

Error Code:	CONFLICT
Error Description:	A data page is marked as being in use in the official software demand paging (SDP) bitmap, and is also marked as being available on one of the free lists.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However, the list is corrupt and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The free list is terminated at the entry immediately preceding the page in conflict, and the remainder of the list that is legitimately free is restored to the free list during garbage collection. The corrective action for this error can cause an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the SDP bitmap in which the block is found to be set as in use.
<i>Bad Data</i> =	Block ID of the block in conflict.
<i>Good Data</i> =	Block ID of the last good block in the free list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 9 Error Code: DATABID

Error Code:	DATABID
Error Description:	An out-of-range data page block ID of a memory-resident relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of a data page for a memory resident relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired along with this audit error report to help find out which relation has lost a data page.
Dump Description:	
<i>Error Address</i> =	Address of the invalid data page block ID of a memory-resident relation.
<i>Bad Data</i> =	Invalid data page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the memory-resident relation that has lost a data page. NOTE: Any bad block ID of one of the pages on the free list will trigger a MEMMAN DATABID error. However, since the pages on the free list are not associated with a relation, there will be no access dictionary entry and therefore no dump will be provided.

Table 10 Error Code: DISKBID

Error Code:	DISKBID
Error Description:	An out-of-range data page block ID for a disk relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of a data page of a disk relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired with this audit error report to help find out

	what disk relation has lost a data page.
Dump Description:	
<i>Error Address</i> =	Address of the out-of-range data page block ID on the head table for a disk relation.
<i>Bad Data</i> =	Invalid block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the disk relation that lost a data page.

Table 11 Error Code: DUPLI

Error Code:	DUPLI
Error Description:	The free page list for pages of a certain size contains a page with a block ID that is also claimed as part of the free page list for pages of a different size.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However the list is corrupt, and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the duplicated page, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the duplicate was located.
<i>Bad Data</i> =	Block ID of the duplicated block.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 12 Error Code: GRBG

Error Code:	GRBG
Error Description:	The database SDP system incorrectly believes that a data page is in use.
Possible Error Effect:	If too many data pages are believed to be seized, recent change will not be able to make updates.
Corrective Action Taken:	The data page is marked as being free in the bit map maintained by the software demand paging system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	Corrected byte of the SDP bit map.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 13 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	An out-of-range head Table block ID is found in the access dictionary.
Possible Error Effect:	This causes the loss of a relation, and it may also cause DB to assert.
Corrective Action Taken:	This head table block ID is zeroed out in the access dictionary.
Manual Action Required:	A DB manual action assert is fired with this error code to catch the attention of maintenance personnel. The manual action assert dump and the audit error dump should be used to figure out which relation is bad.
Dump Description:	
<i>Error Address</i> =	Address of the head table block ID in the access dictionary.
<i>Bad Data</i> =	Out-of-range block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry to show the relation ID, and so forth.

Table 14 Error Code: IPBID

Error Code:	IPBID
Error Description:	An out-of-range intermediate page block ID of a three-level indexing relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of several data pages for a three-level indexing relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired along with this audit error report to catch the attention of maintenance personnel to find out which three-level indexing relation has lost an intermediate page.
Dump Description:	
<i>Error Address</i> =	Address of the invalid intermediate page block ID of a three-level indexing relation.
<i>Bad Data</i> =	Invalid intermediate page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the three-level indexing relation that has lost an intermediate page.

Table 15 Error Code: LINKS

Error Code:	LINKS
Error Description:	The <i>from</i> link in a free page list element is incorrect.
Possible Error Effect:	Some free page list elements may be inaccessible. This contributes to memory fragmentation.
Corrective Action Taken:	The free list is terminated at the entry immediately preceding the element in error, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the free list element.
<i>Bad Data</i> =	Incorrect block ID stored in the from link
<i>Good Data</i> =	Block ID of the last good page on the list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 16 Error Code: LOGNULL

Error Code:	LOGNULL
Error Description:	The MEMMAN audit finds an inconsistent update transaction entry.
Possible Error Effect:	Either this update transaction loses its local log, or this transaction is not an update transaction.
Corrective Action Taken:	None.
Manual Action Required:	Run TRNDC audit.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	NA.

Table 17 Error Code: LOOP

Error Code:	LOOP
Error Description:	The free page list contains two references to the same block ID for the same size free page, causing a cycle.
Possible Error Effect:	Any free pages of the same size that would have followed the cycling page are now inaccessible and thus unavailable for allocation.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the cycling page, and the remainder of the list is restored to the free list during garbage collection.

	This corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the cycle occurs.
<i>Bad Data</i> =	Block ID of the page in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 18 Error Code: MVER

Error Code:	MVER
Error Description:	The database SD) system believes that a data page is free when the data page is actually in use.
Possible Error Effect:	If the data page is allocated for another relation, the two relations will overwrite each other on the same data page.
Corrective Action Taken:	The data page is marked as being seized in the bit map maintained by the software demand paging system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	Corrected byte of the SDP bit map.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 19 Error Code: SIZE

Error Code:	SIZE
Error Description:	The page size that is stored within a free page is not consistent with the free list on which the page is found.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However, the list is corrupt and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the corrupted list entry, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error will cause an UNCLAIMED report during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the page with the mismatched size.
<i>Bad Data</i> =	Incorrect page size, as found within the page.
<i>Good Data</i> =	Correct page size as appropriate for the list.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 20 Error Code: TPBKAU

Error Code:	TPBKAU
Error Description:	The audit of the tuple block bit map (DBtpbkmp) has found a tuple block incorrectly marked as used. This bit map is used to mark tuple blocks as used or not (1 - used, 0 - not in use).
Possible Error Effect:	The tuple block is unavailable for use. This is a loss of resource.
Corrective Action Taken:	The appropriate bit (which is in error) representing a tuple block is set to zero (marking it as free).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of four byte section of the tuple block bit map which contains the error (part of the area pointed to by DBtpbkmp).
<i>Bad Data</i> =	Four byte section of tuple block bit map containing error.
<i>Good Data</i> =	Four byte section corrected.

<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 21 Error Code: UNCLAIMED

Error Code:	UNCLAIMED
Error Description:	The database SDP system bitmap has one or more blocks marked as free that do not appear on the free page lists.
Possible Error Effect:	Pages that are not on the free lists will never be put to use. The blocks are then unusable. If too many free pages are not on the free lists, the ODD may have to be grown unnecessarily, or possibly recent change updates may be blocked.
Corrective Action Taken:	The blocks are placed on the appropriate free lists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect available audit bit map byte.
<i>Bad Data</i> =	Contents of the incorrect byte of the available bit map.
<i>Good Data</i> =	Contents of what the available bit map byte should be.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 22 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after a system interrupt (such as write protect, out-of-range read, and so forth.)
Possible Error Effect:	May cause further resets, audits and or asserts.
Corrective Action Taken:	None, all data that would have caused the reset is reinitialized on the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

MEMMAN-B

Software Release: 5E13 only

1. MEMORY MANAGEMENT (MEMMAN) AUDIT

The software demand paging (SDP) package, a database memory manager, allocates memory within the write-protected office-dependent database (ODD), disk ODD, and the unprotected database based on the availability information stored in bit maps. The SDP bit maps keep track of the space within the ODD, disk ODD, and unprotected database. Each bit corresponds to a block of memory. The runtime access dictionary entries contain pointers (block IDs) to data pages in the ODD, disk ODD, and unprotected database, and to head tables of relations. One access dictionary entry is allocated for each version of a relation in the database. Each entry contains access information for the associated relation version, including a head table index and information about the relation. In Figure 1, for example, the block ID of the entry for relation 1 points to a data page in the ODD; relation 1 is therefore a static one-level indexed relation. The data pages for static two-level indexed, static hash, and static sequential relations such as relation 2 also reside in the ODD. Dynamic relations have their head tables in the ODD and their data pages in unprotected memory. Relation 3 is a dynamic hash or dynamic indexed relation. The indexed head table contains indices that define the data pages that make up the relation version. Relation 4 is a static three-level indexed relation. The head table has indices to all the intermediate data pages (IDPs). The IDPs have indices to data pages.

Memory management of the ODD must ensure that no data pages are lost to the system. The memory management audit (MEMMAN) verifies that the software demand paging package record of memory block usage corresponds to the actual memory blocks in use. If SDP has an unused memory block marked as used, the block is unavailable or lost. If SDP has a used memory block marked as unused, it may overwrite the existing data with new data. The audit detects these inconsistencies and corrects the SDP bit map and internal ODD free lists as appropriate. It also performs several checks on the transaction dictionary blocks because the local log pages of update transactions are in the ODD.

The MEMMAN audit is part of the Data Base audits subgroup. It runs in the OKP and operating system for distributed switching-switching module (OSDS-SM) environments in the segmented mode.

2. DATA STRUCTURES

The memory used to store relations is organized into units called *blocks* and *pages*. A block contains 256 contiguous bytes of memory (note that the term "block" has a different meaning when used in this context than when used synonymously with "tuple" as in an access dictionary block). A page is a set of 1, 2, 4, 8, 16, 32, or 64 contiguous blocks. All the data pages that make up a given relation have the same number of blocks. The number of blocks per page is contained in the access dictionary. This information is used by the audit in conjunction with the relation's head table (if any) to determine which blocks are used by the relation.

The MEMMAN audit checks eight bit maps and two dictionaries to ensure the consistency of availability information about the ODD. The eight bitmaps are divided into two sets, one set for the memory ODD and another for the DISK ODD. The memory ODD includes the static ODD and the dynamic database. Each set has two bitmaps which are complements of each other that are used by the audit to note relation ownership of each block. Another bit map in each set, maintained by SDP, contains one bit per block that stores the allocated/available status for each block. SDP uses this bit map to allocate and deallocate blocks for use as head tables or data pages. In addition, the memory ODD portion of the audit creates two bitmaps that audit the free page lists built into the free ODD pages. Figure 1, shows the layout of system memory, including the ODD. The MEMMAN audit initializes two audit-owned bit maps based on information in the access dictionary and another two bit maps for the memory ODD free page list audit. All these bit maps are checked for consistency with each other and the actual available memory space.

The two dictionaries used by the MEMMAN audit are the runtime access dictionary and the transaction dictionary. The access dictionary contains information about all the relations within ODD. There is an access block in the access dictionary for each version of each relation; it provides the block ID of a head table in ODD. The head table

contains an array of block IDs that point to data pages. The transaction dictionary manages all transactions; the transactions are used primarily by the Recent Change/Verify and Maintenance subsystems.

3. AUDIT CHECKS

The steps of the MEMMAN audit are as follows.

- (a) In its first real-time segment, the audit initializes its four relation ownership bit maps (two relating to the memory ODD and two relating to the disk ODD). The first bit map of each set is initialized to zeros and the second to ones. The first and second bit maps are thus one's complements of each other; this setup provides an internal consistency check for the audit.
- (b) The audit uses the head table pointer of the current access dictionary block to locate the associated head table. It then uses the head table to determine which data pages or IDPs are used to store the associated relation. The number of blocks per page for the relation is in the access dictionary. This information allows the audit to determine which blocks are used to store the relation. Next, the audit compares this block usage information with the block usage information contained in the SDP bit map. There are two possible results: the status of a used block claimed by the audit agrees with the bit map, or the audit claims that a block is used and the bit map claims that it is unused. The second case is an error and the audit simply marks the bit in the SDP bit map to show the block as used. It is also possible that the bit map claims that a block is used when it is not, but the audit cannot determine this until all the access blocks have been read through. During this processing, the audit marks its bit maps for blocks that are found to be used.
- (c) The audit then performs garbage collection. Garbage results from purged processes not cleaned up by the transaction manager or other database audits. Garbage may also result from pages lost because of interrupted transactions. After processing all the access blocks, the audit compares its auxiliary bit maps to the SDP bit map. If a bit is found to be marked as used in the SDP bit map but not in the auxiliary bit map, the audit clears the bit in the SDP bit map.
- (d) The next phase of the audit concerns auditing the integrity of the free lists. The free lists are organized as follows: Each free page of a given size is placed into a linked list of other pages of that size. The forward and previous links to other blocks on the list are stored within the ODD at the page's physical location. SDP maintains a list of the head blocks of each size's list. Blocks are allocated and freed according to a Buddy Block strategy in which adjacent blocks may be coalesced or split depending on the block size requested and the current block distribution. The audit checks that each page is assigned to the appropriate list, that the linkages are consistent and that every block of memory is accounted for.
- (e) The audit then checks the usage of the tuple cache pages. These are pages of the ODD seized by the tuple cache manager to provide space for temporary copies of tuples in memory prior to committal of a transaction. The audit verifies that the internal cache bitmaps are consistent with the contents of the cache.
- (f) The audit checks the global office parameters by marking them according to information in the ODD. The audit performs the checks on one access block per real-time segment. When comparing the auxiliary and SDP bit maps, it processes two bytes of the bit map per segment. The state of the audit is saved before taking a segment break to allow the audit to resume at the correct point when control returns from OSDs.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ABID

Error Code:	ABID
Error Description:	The MEMMAN audit finds an out-of-range block ID (trn_log).
Possible Error Effect:	This update transaction loses its local log.
Corrective Action Taken:	None.
Manual Action Required:	Run the TRNDC audit.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	NA.

Table 2 Error Code: ALIGNMENT

Error Code:	ALIGNMENT
Error Description:	A free page begins on an illegal boundary for its page size. For example, a 2-block page cannot begin at block 1 but may begin at block 2 or 4.
Possible Error Effect:	Memory fragmentation.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the page in error. The removed pages are subsequently reclaimed during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the byte in the free page bitmap where the misalignment was detected.
<i>Bad Data</i> =	Incorrect block ID stored in the pages to/from link.
<i>Good Data</i> =	Block ID of the last good page on the list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 3 Error Code: ARRAY-ENTRY-BAD

Error Code:	ARRAY-ENTRY-BAD
Error Description:	Tuples in the ODD data dictionary relations must have corresponding entries in runtime data dictionaries dbrelidx and/or dbdrlidx. A runtime dictionary entry could not be located or was found on another location for the ODD data dictionary relation identified in the error address.
Possible Error Effect:	Failed or incomplete transactions or initializations or a possible overwrite has occurred.
Corrective Action Taken:	The faulty runtime data dictionary has been updated to contain the missing ODD data dictionary information followed by this report.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Relation ID of the ODD data dictionary relation in error.
<i>Bad Data</i> =	Key of the mismatched ODD data dictionary relation tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: BUDDY

Error Code:	BUDDY
Error Description:	The buddy page for any given available page is also free. The buddy page is defined as one which is: - the same size (for example, both 256 bytes, 512 bytes, and so on). - adjacent (for example, block 0 and block 1 for 256 byte pages). - properly aligned (for example, block 0 and block 1, block 2 and block 3,

	but not block 1 and block 2) to a given page.
	This is an error condition because it violates the constraints of the memory management algorithms.
Possible Error Effect:	This can unnecessarily fragment office-dependent data (ODD) memory. For example, if two 8K buddies exist on the 8K free list and a request for a 16-Kbyte page is made, and no page exists on the 16K free list, new memory is allocated for the ODD section before the request is satisfied. If new memory cannot be allocated, then the request cannot be satisfied.
Corrective Action Taken:	Free buddies are coalesced and put on the free lists for their combined page size.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the free buddy was located.
<i>Bad Data</i> =	Block ID of the free page's buddy.
<i>Good Data</i> =	Block ID of the original free page.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 5 Error Code: CHBKAU

Error Code:	CHBKAU
Error Description:	The audit of the tuple cache bit map (DBchbkmp) has found a cache block incorrectly marked as used. This bit map is used to mark tuple caching blocks as used or not (1 - used, 0 - not in use).
Possible Error Effect:	The cache block is unavailable for use. This is a loss of resource.
Corrective Action Taken:	The appropriate bit (which is in error) representing a cache block is set to zero (marking it as free).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of four byte section of the cache block bit map which contains the error (part of the area pointed to by DBchbkmp).
<i>Bad Data</i> =	Four byte section of cache block bit map containing error.
<i>Good Data</i> =	Four byte section of cache block bit map containing error.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	32 cache blocks corresponding to the 4 byte section of DBchbkmp which had the bad status.

Table 6 Error Code: CNVACCBID

Error Code:	CNVACCBID
Error Description:	An access dictionary block ID of any relation in the RODD is detected by MEMMAN as not being converted by the ODD BST PROCESS.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The access dictionary block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid access dictionary block ID the RODD relation.
<i>Bad Data</i> =	The invalid access dictionary block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The whole access dictionary entry for the RODD relation that has had its access dictionary block ID converted.

Table 7 Error Code: CNVHDBID

Error Code:	CNVHDBID
Error Description:	A head table block ID of a two-level or a three-level indexing relation in the RODD is detected by MEMMAN as not being converted by the ODD BST process.

Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The head table block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address =</i>	Address of the invalid head table block ID of a 2 or a 3-level indexing relation.
<i>Bad Data =</i>	The invalid head table block ID.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The whole access dictionary entry for the two-level or three-level indexing relation that has had its head table block ID converted.

Table 8 Error Code: CNVIPBID

Error Code:	CNVIPBID
Error Description:	An intermediate page block ID of a three-level indexing relation in the RODD is detected by MEMMAN as not being converted by the ODD BST PROCESS.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The intermediate page block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address =</i>	Address of the invalid intermediate page block ID of a three-level indexing relation.
<i>Bad Data =</i>	The invalid intermediate page block ID.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The whole access dictionary entry for the three-level indexing relation that has had its intermediate page converted.

Table 9 Error Code: CONFLICT

Error Code:	CONFLICT
Error Description:	A data page is marked as being in use in the official software demand paging (SDP) bitmap, and is also marked as being available on one of the free lists.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However, the list is corrupt and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The free list is terminated at the entry immediately preceding the page in conflict, and the remainder of the list that is legitimately free is restored to the free list during garbage collection. The corrective action for this error can cause an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Byte in the SDP bitmap in which the block is found to be set as in use.
<i>Bad Data =</i>	Block ID of the block in conflict.
<i>Good Data =</i>	Block ID of the last good block in the free list.
<i>Logical Index =</i>	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump =</i>	NA.

Table 10 Error Code: DATABID

Error Code:	DATABID
Error Description:	An out-of-range data page block ID of a memory-resident relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of a data page for a memory resident relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired along with this audit error report to help find out which relation has lost a data page.
Dump Description:	

<i>Error Address</i> =	Address of the invalid data page block ID of a memory-resident relation.
<i>Bad Data</i> =	Invalid data page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the memory-resident relation that has lost a data page. NOTE: Any bad block ID of one of the pages on the free list will trigger a MEMMAN DATABID error. However, since the pages on the free list are not associated with a relation, there will be no access dictionary entry and therefore no dump will be provided.

Table 11 Error Code: DICTCORRUPT

Error Code:	DICTCORRUPT
Error Description:	A runtime version and an ODD version of the access dictionary exists. Every entry in the runtime version must have a corresponding entry in the ODD version. An entry in the runtime version is missing its corresponding ODD entry. The static ODD relations DBODDACC and DBRODDAC represent the ODD version. Either one of these relations is missing a tuple for the relation ID provided in the dump error address or the runtime entry for this relation is corrupt.
Possible Error Effect:	A faulty processor initialization, an overwrite of the runtime data dictionary entry, or some activity that corrupted a tuple in the static ODD access specific relations has occurred corrupting the data contained in the runtime version of the access dictionary or its corresponding static ODD tuple.
Corrective Action Taken:	None.
Manual Action Required:	Identify the missing data using the relation ID provided in the error address. Determine if this tuple resides in the appropriate ODD access relation. If it does not exist it should be added using the information provided in the <i>Good Data</i> section of the Dump Description. If it does exist then the entry in the runtime version of the access dictionary (dbdoddacc) is corrupt.
Dump Description:	
<i>Error Address</i> =	Relation ID of the corrupt data dictionary relation.
<i>Bad Data</i> =	Key of the corrupt data dictionary relation.
<i>Good Data</i> =	Expected data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 12 Error Code: DICTINCONS

Error Code:	DICTINCONS
Error Description:	Specific data dictionary information requires corresponding tuples in multiple static ODD data dictionary relations. An inconsistency between those data dictionary relations has been detected. A tuple in one data dictionary relation is missing its corresponding tuple in one of the other data dictionary relations.
Possible Error Effect:	Static ODD corruption.
Corrective Action Taken:	None.
Manual Action Required:	Restore consistency across data dictionary relations. Using the information provided in the Good Data section of the Dump Description determine if the missing tuple(s) exists in the same dictionary relation on another processor. If so, insert that missing tuple into the static ODD on the processor where the inconsistency occurred.
Dump Description:	
<i>Error Address</i> =	Relation ID of the data dictionary relation where the inconsistency exists.
<i>Bad Data</i> =	Key of the data dictionary relation tuple not found.
<i>Good Data</i> =	Expected tuple data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 13 Error Code: DICTLINKERR

Error Code:	DICTLINKERR
Error Description:	The ODD data dictionaries are constructed using linked lists. Corruption has been detected in one of these linked lists.
Possible Error Effect:	Any event that could corrupt a linked list.
Corrective Action Taken:	None.
Manual Action Required:	The static ODD data dictionary relation containing the corruption is identified in the error address portion of this assert. The key of the tuple where the corruption was detected is provided in the bad data portion of this assert. Examine the tuple(s) positioned just prior to and immediately following the failing tuple to determine the proper link values. Refer to Figure 2.
Dump Description:	
<i>Error Address</i> =	Relation ID of the data dictionary relation where the inconsistency exists.
<i>Bad Data</i> =	Key of the data dictionary relation tuple not found.
<i>Good Data</i> =	Expected tuple data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 14 Error Code: DISKBID

Error Code:	DISKBID
Error Description:	An out-of-range data page block ID for a disk relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of a data page of a disk relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired with this audit error report to help find out what disk relation has lost a data page.
Dump Description:	
<i>Error Address</i> =	Address of the out-of-range data page block ID on the head table for a disk relation.
<i>Bad Data</i> =	Invalid block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the disk relation that lost a data page.

Table 15 Error Code: DUPLI

Error Code:	DUPLI
Error Description:	The free page list for pages of a certain size contains a page with a block ID that is also claimed as part of the free page list for pages of a different size.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However the list is corrupt, and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the duplicated page, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the duplicate was located.
<i>Bad Data</i> =	Block ID of the duplicated block.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 16 Error Code: GRBG

Error Code:	GRBG
Error Description:	The database SDP system incorrectly believes that a data page is in use.
Possible Error Effect:	If too many data pages are believed to be seized, recent change will not be able to make updates.
Corrective Action Taken:	The data page is marked as being free in the bit map maintained by the software

	demand paging system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	Corrected byte of the SDP bit map.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 17 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	An out-of-range head Table block ID is found in the access dictionary.
Possible Error Effect:	This causes the loss of a relation, and it may also cause DB to assert.
Corrective Action Taken:	This head table block ID is zeroed out in the access dictionary.
Manual Action Required:	A DB manual action assert is fired with this error code to catch the attention of maintenance personnel. The manual action assert dump and the audit error dump should be used to figure out which relation is bad.
Dump Description:	
<i>Error Address</i> =	Address of the head table block ID in the access dictionary.
<i>Bad Data</i> =	Out-of-range block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry to show the relation ID, and so forth.

Table 18 Error Code: IPBID

Error Code:	IPBID
Error Description:	An out-of-range intermediate page block ID of a three-level indexing relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of several data pages for a three-level indexing relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired along with this audit error report to catch the attention of maintenance personnel to find out which three-level indexing relation has lost an intermediate page.
Dump Description:	
<i>Error Address</i> =	Address of the invalid intermediate page block ID of a three-level indexing relation.
<i>Bad Data</i> =	Invalid intermediate page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the three-level indexing relation that has lost an intermediate page.

Table 19 Error Code: LINKS

Error Code:	LINKS
Error Description:	The <i>from</i> link in a free page list element is incorrect.
Possible Error Effect:	Some free page list elements may be inaccessible. This contributes to memory fragmentation.
Corrective Action Taken:	The free list is terminated at the entry immediately preceding the element in error, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the free list element.
<i>Bad Data</i> =	Incorrect block ID stored in the from link
<i>Good Data</i> =	Block ID of the last good page on the list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 20 Error Code: LOGNULL

Error Code:	LOGNULL
Error Description:	The MEMMAN audit finds an inconsistent update transaction entry.
Possible Error Effect:	Either this update transaction loses its local log, or this transaction is not an update transaction.
Corrective Action Taken:	None.
Manual Action Required:	Run TRNDC audit.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	NA.

Table 21 Error Code: LOOP

Error Code:	LOOP
Error Description:	The free page list contains two references to the same block ID for the same size free page, causing a cycle.
Possible Error Effect:	Any free pages of the same size that would have followed the cycling page are now inaccessible and thus unavailable for allocation.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the cycling page, and the remainder of the list is restored to the free list during garbage collection. This corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the cycle occurs.
<i>Bad Data</i> =	Block ID of the page in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 22 Error Code: MVER

Error Code:	MVER
Error Description:	The database SD) system believes that a data page is free when the data page is actually in use.
Possible Error Effect:	If the data page is allocated for another relation, the two relations will overwrite each other on the same data page.
Corrective Action Taken:	The data page is marked as being seized in the bit map maintained by the software demand paging system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	Corrected byte of the SDP bit map.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 23 Error Code: SIZE

Error Code:	SIZE
Error Description:	The page size that is stored within a free page is not consistent with the free list on which the page is found.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However, the list is corrupt and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the corrupted list entry, and the remainder of the list is restored to the free list during garbage collection.

	The corrective action for this error will cause an UNCLAIMED report during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the page with the mismatched size.
<i>Bad Data</i> =	Incorrect page size, as found within the page.
<i>Good Data</i> =	Correct page size as appropriate for the list.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 24 Error Code: TPBKAU

Error Code:	TPBKAU
Error Description:	The audit of the tuple block bit map (DBtpbkmp) has found a tuple block incorrectly marked as used. This bit map is used to mark tuple blocks as used or not (1 - used, 0 - not in use).
Possible Error Effect:	The tuple block is unavailable for use. This is a loss of resource.
Corrective Action Taken:	The appropriate bit (which is in error) representing a tuple block is set to zero (marking it as free).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of four byte section of the tuple block bit map which contains the error (part of the area pointed to by DBtpbkmp).
<i>Bad Data</i> =	Four byte section of tuple block bit map containing error.
<i>Good Data</i> =	Four byte section corrected.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 25 Error Code: UNCLAIMED

Error Code:	UNCLAIMED
Error Description:	The database SDP system bitmap has one or more blocks marked as free that do not appear on the free page lists.
Possible Error Effect:	Pages that are not on the free lists will never be put to use. The blocks are then unusable. If too many free pages are not on the free lists, the ODD may have to be grown unnecessarily, or possibly recent change updates may be blocked.
Corrective Action Taken:	The blocks are placed on the appropriate free lists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect available audit bit map byte.
<i>Bad Data</i> =	Contents of the incorrect byte of the available bit map.
<i>Good Data</i> =	Contents of what the available bit map byte should be.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 26 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after a system interrupt (such as write protect, out-of-range read, and so forth.)
Possible Error Effect:	May cause further resets, audits and or asserts.
Corrective Action Taken:	None, all data that would have caused the reset is reinitialized on the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

MEMMAN-C

Software Release: 5E14 and later

1. MEMORY MANAGEMENT (MEMMAN) AUDIT

The software demand paging (SDP) package, a database memory manager, allocates memory within the write-protected office-dependent database (ODD), disk ODD, and the unprotected database based on the availability information stored in bit maps. The SDP bit maps keep track of the space within the ODD, disk ODD, and unprotected database. Each bit corresponds to a block of memory. The runtime access dictionary entries contain pointers (block IDs) to data pages in the ODD, disk ODD, and unprotected database, and to head tables of relations. One access dictionary entry is allocated for each version of a relation in the database. Each entry contains access information for the associated relation version, including a head table index and information about the relation. In Figure 1, for example, the block ID of the entry for relation 1 points to a data page in the ODD; relation 1 is therefore a static one-level indexed relation. The data pages for static two-level indexed, static hash, and static sequential relations such as relation 2 also reside in the ODD. Dynamic relations have their head tables in the ODD and their data pages in unprotected memory. Relation 3 is a dynamic hash or dynamic indexed relation. The indexed head table contains indices that define the data pages that make up the relation version. Relation 4 is a static three-level indexed relation. The head table has indices to all the intermediate data pages (IDPs). The IDPs have indices to data pages.

Memory management of the ODD must ensure that no data pages are lost to the system. The memory management audit (MEMMAN) verifies that the software demand paging package record of memory block usage corresponds to the actual memory blocks in use. If SDP has an unused memory block marked as used, the block is unavailable or lost. If SDP has a used memory block marked as unused, it may overwrite the existing data with new data. The audit detects these inconsistencies and corrects the SDP bit map and internal ODD free lists as appropriate. It also performs several checks on the transaction dictionary blocks because the local log pages of update transactions are in the ODD.

The MEMMAN audit is part of the Data Base audits subgroup. It runs in the OKP and operating system for distributed switching-switching module (OSDS-SM) environments in the segmented mode.

2. DATA STRUCTURES

The memory used to store relations is organized into units called *blocks* and *pages*. A block contains 256 contiguous bytes of memory (note that the term "block" has a different meaning when used in this context than when used synonymously with "tuple" as in an access dictionary block). A page is a set of 1, 2, 4, 8, 16, 32, or 64 contiguous blocks. Page size bitmaps keep track of page IDs, which are page sizes identified by the block ID of the first block of the page. All the data pages that make up a given relation have the same number of blocks. The number of blocks per page is contained in the access dictionary. This information is used by the audit in conjunction with the relation's head table (if any) to determine which blocks are used by the relation.

The MEMMAN audit checks eight bit maps and two dictionaries to ensure the consistency of availability information about the ODD. The eight bitmaps are divided into two sets, one set for the memory ODD and another for the DISK ODD. The memory ODD includes the static ODD and the dynamic database. Each set has two bitmaps which are complements of each other that are used by the audit to note relation ownership of each block. Another bit map in each set, maintained by SDP, contains one bit per block that stores the allocated/available status for each block. SDP uses this bit map to allocate and deallocate blocks for use as head tables or data pages. In addition, the memory ODD portion of the audit creates two bitmaps that audit the free page lists built into the free ODD pages. Figure 1, shows the layout of system memory, including the ODD. The MEMMAN audit initializes two audit-owned bit maps based on information in the access dictionary and another two bit maps for the memory ODD free page list audit. All these bit maps are checked for consistency with each other and the actual available memory space.

The two dictionaries used by the MEMMAN audit are the runtime access dictionary and the transaction dictionary. The access dictionary contains information about all the relations within ODD. There is an access block in the

access dictionary for each version of each relation; it provides the block ID of a head table in ODD. The head table contains an array of block IDs that point to data pages. The transaction dictionary manages all transactions; the transactions are used primarily by the Recent Change/Verify and Maintenance subsystems.

The MEMMAN audit checks that used block IDs have their associated page ID marked in the official page size bitmap.

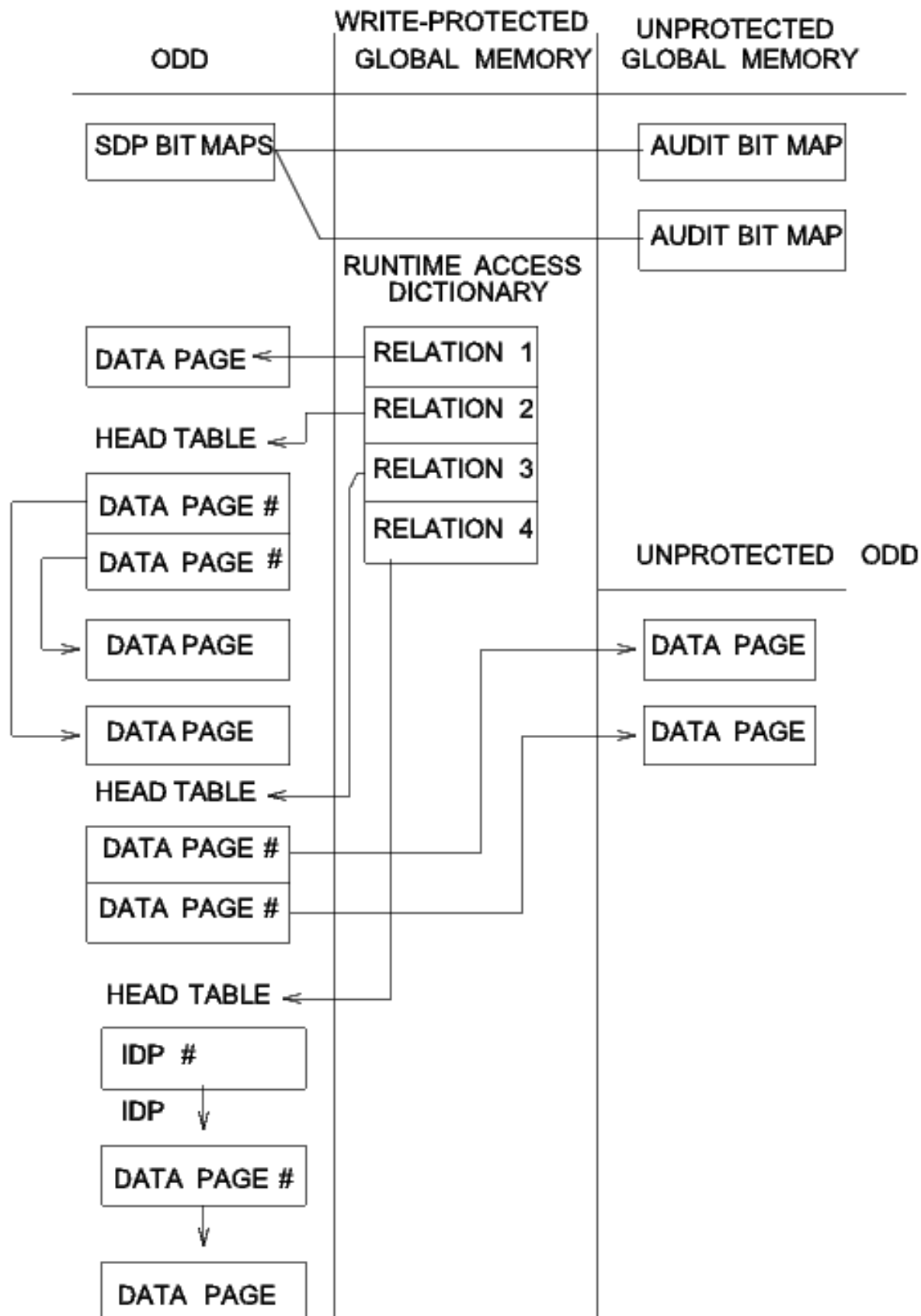


Figure 1 (MEMMAN-1) Layout of Memory

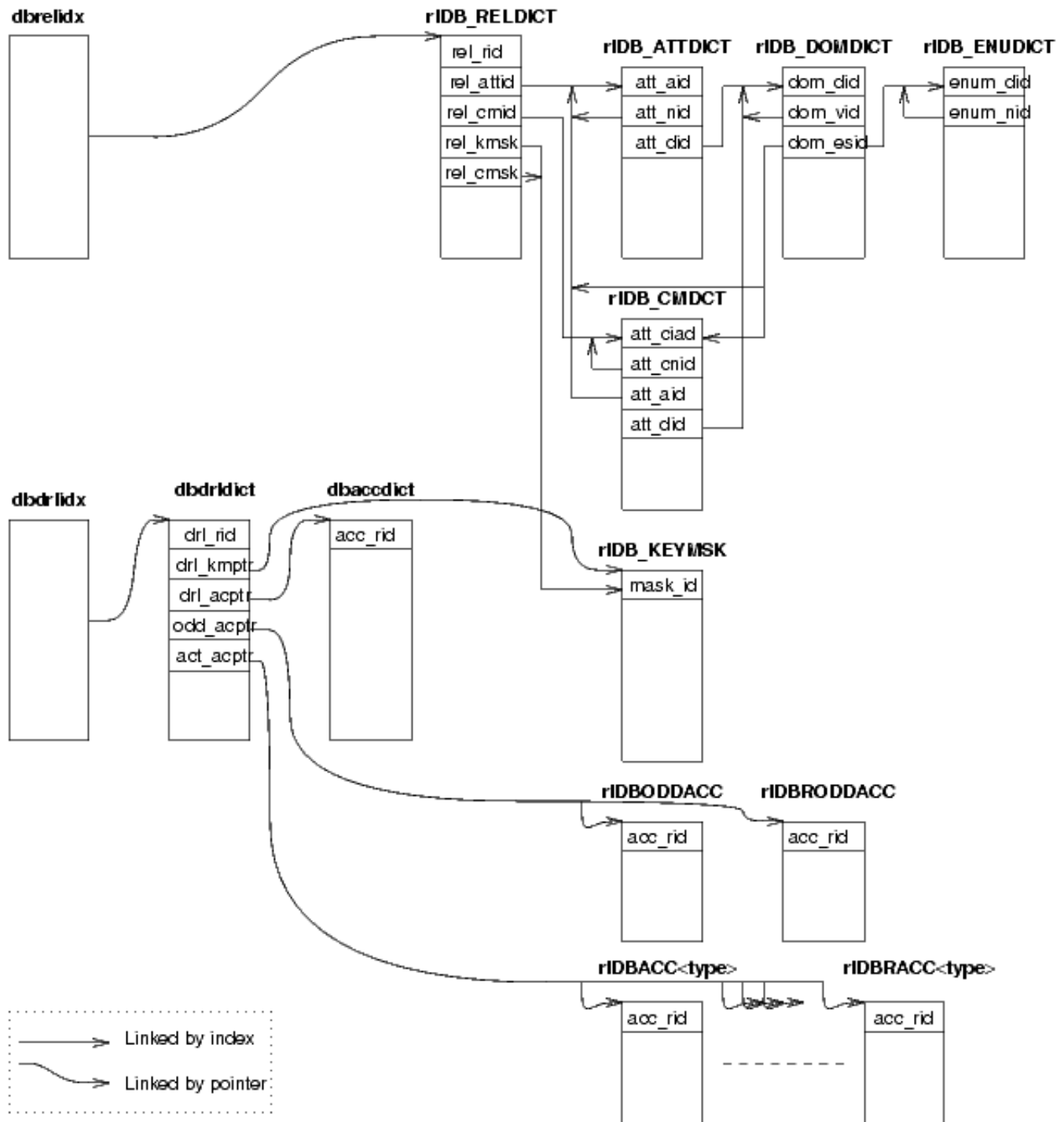


Figure 2 (MEMMAN-2) Dictionary Link Diagram

3. AUDIT CHECKS

The steps of the MEMMAN audit are as follows.

- In its first real-time segment, the audit initializes its four relation ownership bit maps (two relating to the

memory ODD and two relating to the disk ODD). The first bit map of each set is initialized to zeros and the second to ones. The first and second bit maps are thus one's complements of each other; this setup provides an internal consistency check for the audit.

- (b) The audit uses the head table pointer of the current access dictionary block to locate the associated head table. It then uses the head table to determine which data pages or IDPs are used to store the associated relation. The number of blocks per page for the relation is in the access dictionary. This information allows the audit to determine which blocks are used to store the relation. Next, the audit compares this block usage information with the block usage information contained in the SDP bit map. There are two possible results: the status of a used block claimed by the audit agrees with the bit map, or the audit claims that a block is used and the bit map claims that it is unused. The second case is an error and the audit simply marks the bit in the SDP bit map to show the block as used. It is also possible that the bit map claims that a block is used when it is not, but the audit cannot determine this until all the access blocks have been read through. During this processing, the audit marks its bit maps for blocks that are found to be used.
- (c) The audit then performs garbage collection. Garbage results from purged processes not cleaned up by the transaction manager or other database audits. Garbage may also result from pages lost because of interrupted transactions. After processing all the access blocks, the audit compares its auxiliary bit maps to the SDP bit map. If a bit is found to be marked as used in the SDP bit map but not in the auxiliary bit map, the audit clears the bit in the SDP bit map.
- (d) The next phase of the audit concerns auditing the integrity of the free lists. The free lists are organized as follows: Each free page of a given size is placed into a linked list of other pages of that size. The forward and previous links to other blocks on the list are stored within the ODD at the page's physical location. SDP maintains a list of the head blocks of each size's list. Blocks are allocated and freed according to a Buddy Block strategy in which adjacent blocks may be coalesced or split depending on the block size requested and the current block distribution. The audit checks that each page is assigned to the appropriate list, that the linkages are consistent and that every block of memory is accounted for.
- (e) The audit then checks the usage of the tuple cache pages. These are pages of the ODD seized by the tuple cache manager to provide space for temporary copies of tuples in memory prior to committal of a transaction. The audit verifies that the internal cache bitmaps are consistent with the contents of the cache.
- (f) The audit checks the global office parameters by marking them according to information in the ODD. The audit performs the checks on one access block per real-time segment. When comparing the auxiliary and SDP bit maps, it processes two bytes of the bit map per segment. The state of the audit is saved before taking a segment break to allow the audit to resume at the correct point when control returns from OSDs.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ABID

Error Code:	ABID
Error Description:	The MEMMAN audit finds an out-of-range block ID (trn_log).
Possible Error Effect:	This update transaction loses its local log.
Corrective Action Taken:	None.
Manual Action Required:	Run the TRNDC audit.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	NA.

Table 2 Error Code: ALIGNMENT

Error Code:	ALIGNMENT
Error Description:	A free page begins on an illegal boundary for its page size. For example, a 2-block page cannot begin at block 1 but may begin at block 2 or 4.
Possible Error Effect:	Memory fragmentation.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the page in error. The removed pages are subsequently reclaimed during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the byte in the free page bitmap where the misalignment was detected.
<i>Bad Data</i> =	Incorrect block ID stored in the pages to/from link.
<i>Good Data</i> =	Block ID of the last good page on the list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 3 Error Code: ARRAY-ENTRY-BAD

Error Code:	ARRAY-ENTRY-BAD
Error Description:	Tuples in the ODD data dictionary relations must have corresponding entries in runtime data dictionaries dbrelidx and/or dbdrlidx. A runtime dictionary entry could not be located or was found on another location for the ODD data dictionary relation identified in the error address.
Possible Error Effect:	Failed or incomplete transactions or initializations or a possible overwrite has occurred.
Corrective Action Taken:	The faulty runtime data dictionary has been updated to contain the missing ODD data dictionary information followed by this report.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Relation ID of the ODD data dictionary relation in error.
<i>Bad Data</i> =	Key of the mismatched ODD data dictionary relation tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 4 Error Code: BUDDY

Error Code:	BUDDY
Error Description:	The buddy page for any given available page is also free. The buddy page is defined as one which is: - the same size (for example, both 256 bytes, 512 bytes, and so on). - adjacent (for example, block 0 and block 1 for 256 byte pages). - properly aligned (for example, block 0 and block 1, block 2 and block 3, but not block 1 and block 2) to a given page. This is an error condition because it violates the constraints of the memory management algorithms.
Possible Error Effect:	This can unnecessarily fragment office-dependent data (ODD) memory. For example, if two 8K buddies exist on the 8K free list and a request for a 16-Kbyte page is made, and no page exists on the 16K free list, new memory is allocated

	for the ODD section before the request is satisfied. If new memory cannot be allocated, then the request cannot be satisfied.
Corrective Action Taken:	Free buddies are coalesced and put on the free lists for their combined page size.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the free buddy was located.
<i>Bad Data</i> =	Block ID of the free page's buddy.
<i>Good Data</i> =	Block ID of the original free page.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 5 Error Code: CHBKAU

Error Code:	CHBKAU
Error Description:	The audit of the tuple cache bit map (DBchbkmp) has found a cache block incorrectly marked as used. This bit map is used to mark tuple caching blocks as used or not (1 - used, 0 - not in use).
Possible Error Effect:	The cache block is unavailable for use. This is a loss of resource.
Corrective Action Taken:	The appropriate bit (which is in error) representing a cache block is set to zero (marking it as free).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of four byte section of the cache block bit map which contains the error (part of the area pointed to by DBchbkmp).
<i>Bad Data</i> =	Four byte section of cache block bit map containing error.
<i>Good Data</i> =	Four byte section of cache block bit map containing error.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	32 cache blocks corresponding to the 4 byte section of DBchbkmp which had the bad status.

Table 6 Error Code: CNVACCBID

Error Code:	CNVACCBID
Error Description:	An access dictionary block ID of any relation in the RODD is detected by MEMMAN as not being converted by the ODD BST PROCESS.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The access dictionary block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid access dictionary block ID the RODD relation.
<i>Bad Data</i> =	The invalid access dictionary block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The whole access dictionary entry for the RODD relation that has had its access dictionary block ID converted.

Table 7 Error Code: CNVHDBID

Error Code:	CNVHDBID
Error Description:	A head table block ID of a two-level or a three-level indexing relation in the RODD is detected by MEMMAN as not being converted by the ODD BST process.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The head table block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid head table block ID of a 2 or a 3-level indexing relation.
<i>Bad Data</i> =	The invalid head table block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.

<i>Dump</i> =	The whole access dictionary entry for the two-level or three-level indexing relation that has had its head table block ID converted.

Table 8 Error Code: CNVIPBID

Error Code:	CNVIPBID
Error Description:	An intermediate page block ID of a three-level indexing relation in the RODD is detected by MEMMAN as not being converted by the ODD BST PROCESS.
Possible Error Effect:	Since this was caught and converted by the MEMMAN audit, it has been corrected. This would only be a warning.
Corrective Action Taken:	The intermediate page block ID is converted.
Manual Action Required:	None. See Corrective Action Taken .
Dump Description:	
<i>Error Address</i> =	Address of the invalid intermediate page block ID of a three-level indexing relation.
<i>Bad Data</i> =	The invalid intermediate page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The whole access dictionary entry for the three-level indexing relation that has had its intermediate page converted.

Table 9 Error Code: CONFLICT

Error Code:	CONFLICT
Error Description:	A data page is marked as being in use in the official software demand paging (SDP) bitmap, and is also marked as being available on one of the free lists.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However, the list is corrupt and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The free list is terminated at the entry immediately preceding the page in conflict, and the remainder of the list that is legitimately free is restored to the free list during garbage collection. The corrective action for this error can cause an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the SDP bitmap in which the block is found to be set as in use.
<i>Bad Data</i> =	Block ID of the block in conflict.
<i>Good Data</i> =	Block ID of the last good block in the free list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 10 Error Code: DATABID

Error Code:	DATABID
Error Description:	An out-of-range data page block ID of a memory-resident relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of a data page for a memory resident relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired along with this audit error report to help find out which relation has lost a data page.
Dump Description:	
<i>Error Address</i> =	Address of the invalid data page block ID of a memory-resident relation.
<i>Bad Data</i> =	Invalid data page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the memory-resident relation that has lost a data page. NOTE: Any bad block ID of one of the pages on the free list will trigger a MEMMAN DATABID error. However, since the pages on the free

	list are not associated with a relation, there will be no access dictionary entry and therefore no dump will be provided.

Table 11 Error Code: DICTCORRUPT

Error Code:	DICTCORRUPT
Error Description:	A runtime version and an ODD version of the access dictionary exists. Every entry in the runtime version must have a corresponding entry in the ODD version. An entry in the runtime version is missing its corresponding ODD entry. The static ODD relations DBODDACC and DBRODDAC represent the ODD version. Either one of these relations is missing a tuple for the relation ID provided in the dump error address or the runtime entry for this relation is corrupt.
Possible Error Effect:	A faulty processor initialization, an overwrite of the runtime data dictionary entry, or some activity that corrupted a tuple in the static ODD access specific relations has occurred corrupting the data contained in the runtime version of the access dictionary or its corresponding static ODD tuple.
Corrective Action Taken:	None.
Manual Action Required:	Identify the missing data using the relation ID provided in the error address. Determine if this tuple resides in the appropriate ODD access relation. If it does not exist it should be added using the information provided in the <i>Good Data</i> section of the Dump Description. If it does exist then the entry in the runtime version of the access dictionary (dbdoddacc) is corrupt.
Dump Description:	
<i>Error Address</i> =	Relation ID of the corrupt data dictionary relation.
<i>Bad Data</i> =	Key of the corrupt data dictionary relation.
<i>Good Data</i> =	Expected data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 12 Error Code: DICTINCONS

Error Code:	DICTINCONS
Error Description:	Specific data dictionary information requires corresponding tuples in multiple static ODD data dictionary relations. An inconsistency between those data dictionary relations has been detected. A tuple in one data dictionary relation is missing its corresponding tuple in one of the other data dictionary relations.
Possible Error Effect:	Static ODD corruption.
Corrective Action Taken:	None.
Manual Action Required:	Restore consistency across data dictionary relations. Using the information provided in the Good Data section of the Dump Description determine if the missing tuple(s) exists in the same dictionary relation on another processor. If so, insert that missing tuple into the static ODD on the processor where the inconsistency occurred.
Dump Description:	
<i>Error Address</i> =	Relation ID of the data dictionary relation where the inconsistency exists.
<i>Bad Data</i> =	Key of the data dictionary relation tuple not found.
<i>Good Data</i> =	Expected tuple data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 13 Error Code: DICTLINKERR

Error Code:	DICTLINKERR
Error Description:	The ODD data dictionaries are constructed using linked lists. Corruption has been detected in one of these linked lists.
Possible Error Effect:	Any event that could corrupt a linked list.
Corrective Action Taken:	None.
Manual Action Required:	The static ODD data dictionary relation containing the corruption is identified in the error address portion of this assert. The key of the tuple where the corruption was detected is provided in the bad data portion of this assert. Examine the

	tuple(s) positioned just prior to and immediately following the failing tuple to determine the proper link values. Refer to Figure 2.
Dump Description:	
<i>Error Address</i> =	Relation ID of the data dictionary relation where the inconsistency exists.
<i>Bad Data</i> =	Key of the data dictionary relation tuple not found.
<i>Good Data</i> =	Expected tuple data.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 14 Error Code: DISKBID

Error Code:	DISKBID
Error Description:	An out-of-range data page block ID for a disk relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of a data page of a disk relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired with this audit error report to help find out what disk relation has lost a data page.
Dump Description:	
<i>Error Address</i> =	Address of the out-of-range data page block ID on the head table for a disk relation.
<i>Bad Data</i> =	Invalid block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the disk relation that lost a data page.

Table 15 Error Code: DUPLI

Error Code:	DUPLI
Error Description:	The free page list for pages of a certain size contains a page with a block ID that is also claimed as part of the free page list for pages of a different size.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However the list is corrupt, and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the duplicated page, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the duplicate was located.
<i>Bad Data</i> =	Block ID of the duplicated block.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 16 Error Code: GRBG

Error Code:	GRBG
Error Description:	The database SDP system incorrectly believes that a data page is in use.
Possible Error Effect:	If too many data pages are believed to be seized, recent change will not be able to make updates.
Corrective Action Taken:	The data page is marked as being free in the bit map maintained by the software demand paging system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	Corrected byte of the SDP bit map.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 17 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	An out-of-range head Table block ID is found in the access dictionary.
Possible Error Effect:	This causes the loss of a relation, and it may also cause DB to assert.
Corrective Action Taken:	This head table block ID is zeroed out in the access dictionary.
Manual Action Required:	A DB manual action assert is fired with this error code to catch the attention of maintenance personnel. The manual action assert dump and the audit error dump should be used to figure out which relation is bad.
Dump Description:	
<i>Error Address</i> =	Address of the head table block ID in the access dictionary.
<i>Bad Data</i> =	Out-of-range block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry to show the relation ID, and so forth.

Table 18 Error Code: IPBID

Error Code:	IPBID
Error Description:	An out-of-range intermediate page block ID of a three-level indexing relation is detected by MEMMAN.
Possible Error Effect:	This may cause the loss of several data pages for a three-level indexing relation.
Corrective Action Taken:	None.
Manual Action Required:	A DB manual action assert is fired along with this audit error report to catch the attention of maintenance personnel to find out which three-level indexing relation has lost an intermediate page.
Dump Description:	
<i>Error Address</i> =	Address of the invalid intermediate page block ID of a three-level indexing relation.
<i>Bad Data</i> =	Invalid intermediate page block ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Whole access dictionary entry for the three-level indexing relation that has lost an intermediate page.

Table 19 Error Code: LINKS

Error Code:	LINKS
Error Description:	The <i>from</i> link in a free page list element is incorrect.
Possible Error Effect:	Some free page list elements may be inaccessible. This contributes to memory fragmentation.
Corrective Action Taken:	The free list is terminated at the entry immediately preceding the element in error, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the free list element.
<i>Bad Data</i> =	Incorrect block ID stored in the from link
<i>Good Data</i> =	Block ID of the last good page on the list.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 20 Error Code: LOGNULL

Error Code:	LOGNULL
Error Description:	The MEMMAN audit finds an inconsistent update transaction entry.
Possible Error Effect:	Either this update transaction loses its local log, or this transaction is not an update transaction.
Corrective Action Taken:	None.
Manual Action Required:	Run TRNDC audit.
Dump Description:	

<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	NA.

Table 21 Error Code: LOOP

Error Code:	LOOP
Error Description:	The free page list contains two references to the same block ID for the same size free page, causing a cycle.
Possible Error Effect:	Any free pages of the same size that would have followed the cycling page are now inaccessible and thus unavailable for allocation.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the cycling page, and the remainder of the list is restored to the free list during garbage collection. This corrective action for this error causes an UNCLAIMED error report to occur during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Byte in the free page bitmap where the cycle occurs.
<i>Bad Data</i> =	Block ID of the page in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index which represents the page size on the list. 0 for the 256 list, 1 for the 512 list, and so forth.
<i>Dump</i> =	NA.

Table 22 Error Code: MVER

Error Code:	MVER
Error Description:	The database SD) system believes that a data page is free when the data page is actually in use.
Possible Error Effect:	If the data page is allocated for another relation, the two relations will overwrite each other on the same data page.
Corrective Action Taken:	The data page is marked as being seized in the bit map maintained by the software demand paging system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	Corrected byte of the SDP bit map.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 23 Error Code: NOT_IN_PGTBL

Error Code:	NOT_IN_PGTBL
Error Description:	A page ID associated with a used block ID is not marked in the appropriate page size table.
Possible Error Effect:	Possibility of shared data pages and/or bad block ID.
Corrective Action Taken:	The appropriate bits which are incorrectly cleared in the page size tables are set to one.
Manual Action Required:	Contact customer technical support immediately to investigate the possibility of shared data pages and/or bad block IDs. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Dump Description:	
<i>Error Address</i> =	Address of the block ID location in the page size table.
<i>Bad Data</i> =	Block ID.
<i>Good Data</i> =	Contents of the block ID location in the page size table.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 24 Error Code: SIZE

Error Code:	SIZE
Error Description:	The page size that is stored within a free page is not consistent with the free list on which the page is found.
Possible Error Effect:	The memory management code ensures that the same block ID will never be allocated twice. However, the list is corrupt and the code will fire an assert and fail the request should there be an attempt to seize an already seized block.
Corrective Action Taken:	The list is terminated at the entry immediately preceding the corrupted list entry, and the remainder of the list is restored to the free list during garbage collection. The corrective action for this error will cause an UNCLAIMED report during garbage collection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the page with the mismatched size.
<i>Bad Data</i> =	Incorrect page size, as found within the page.
<i>Good Data</i> =	Correct page size as appropriate for the list.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 25 Error Code: TPBKAU

Error Code:	TPBKAU
Error Description:	The audit of the tuple block bit map (DBtpbkmp) has found a tuple block incorrectly marked as used. This bit map is used to mark tuple blocks as used or not (1 - used, 0 - not in use).
Possible Error Effect:	The tuple block is unavailable for use. This is a loss of resource.
Corrective Action Taken:	The appropriate bit (which is in error) representing a tuple block is set to zero (marking it as free).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of four byte section of the tuple block bit map which contains the error (part of the area pointed to by DBtpbkmp).
<i>Bad Data</i> =	Four byte section of tuple block bit map containing error.
<i>Good Data</i> =	Four byte section corrected.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 26 Error Code: UNCLAIMED

Error Code:	UNCLAIMED
Error Description:	The database SDP system bitmap has one or more blocks marked as free that do not appear on the free page lists.
Possible Error Effect:	Pages that are not on the free lists will never be put to use. The blocks are then unusable. If too many free pages are not on the free lists, the ODD may have to be grown unnecessarily, or possibly recent change updates may be blocked.
Corrective Action Taken:	The blocks are placed on the appropriate free lists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect available audit bit map byte.
<i>Bad Data</i> =	Contents of the incorrect byte of the available bit map.
<i>Good Data</i> =	Contents of what the available bit map byte should be.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 27 Error Code: WRONG_PGTBL

Error Code:	WRONG_PGTBL
Error Description:	A page ID associated with a used block ID is marked in a page size table it should not be marked in.
Possible Error Effect:	Possibility of shared data pages and/or bad block ID.
Corrective Action Taken:	The appropriate bits which are incorrectly marked in the page size tables are set to zero.
Manual Action Required:	Contact customer technical support immediately to investigate the possibility of

	shared data pages and/or bad block IDs. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Dump Description:	
<i>Error Address</i> =	Address of the block ID location in the page size table.
<i>Bad Data</i> =	Block ID.
<i>Good Data</i> =	Contents of the block ID location in the page size table.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 28 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after a system interrupt (such as write protect, out-of-range read, and so forth.)
Possible Error Effect:	May cause further resets, audits and or asserts.
Corrective Action Taken:	None, all data that would have caused the reset is reinitialized on the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the incorrect SDP bit map byte.
<i>Bad Data</i> =	Incorrect byte of the SDP bit map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

MMGR

Software Release: 5E10 and later

1. MEMORY MANAGER (MMGR) AUDIT

The operating system memory manager (MMGR) administers all memory allocation requests and manipulates the memory management resources accordingly. The resources managed by the MMGR include those associated with the administrative modules main memory (segments and pages) and swap space on disk. In order to ensure continuous operating system operation, it is essential that these resources be protected from loss and corruption. The memory manager audits are designed to ensure the integrity of memory manager resources and prevent their permanent loss.

There are six MMGR audits, numbered 1, 2, 3, 4, 5, and 9. The first three audits are designed to run in order and repeatedly. For example, a lost page table is recovered by running the MMGR 2 audit, followed by the MMGR 1 audit, followed by the MMGR 2 audit, each audit in correcting mode. A similar sequence is required for the recovery of lost pages. Therefore, each of the MMGR 2 and MMGR 3 audits should be rerun only after the MMGR 1 audit has run. The routine scheduling of the MMGR audits follows this sequence. Also, the first three audits should all be allowed or all be inhibited.

The MMGR 1 audit:

- Validates the list of free segment descriptor table entries (SDEs).
- Recovers lost SDEs that are not allocated and not on the free list.
- For each SDE for an in-memory segment, validates the linkages to its page table and the page descriptor entries (PDEs) associated with its pages.

The MMGR 2 audit:

- Validates the list of free page tables.
- Recovers lost page tables that are not validly linked to an SDE and are not on the free list.

The MMGR 3 audit:

- Validates the list of free pages.
- Recovers lost pages that are not linked to a segment and are not on the free list.

The MMGR 4 audit:

- Detects and completes an interrupted breakdown of a segment.
- Releases the segment with its associated page table and pages.

The MMGR 5 audit:

- Detects old IO locked segments. A segment slotted for release but locked in memory for an IO job is saved until the job completes and the segment is unlocked. It is possible that under certain conditions this segment may never be unlocked. After 30 minutes, it is assumed that the segment is forgotten.

The MMGR 9 audit:

- Compacts disk swap space.
- Rearranges allocated swap space to provide a large free space consisting of contiguous blocks. This audit is run only when manually invoked, usually in response to repeated REPT:MMGR004 output messages. These messages signal that the available free swap space fragmented into sets of contiguous free blocks too small to satisfy a request and that an attempt has been made to compact the swap space cheaply.

The compaction done by the audit, however, is not cheap. While this compaction is running, the memory manager is not able to swap supervisor processes into memory. As a result, the only supervisor processes that run during this interval are those already in memory. The real-time interval varies with system activity, usually several seconds.

1.1 Requesting the MMGR Audits

Although the MMGR 1 through 5 audits are scheduled to run routinely, they can be requested by using the following input message. The MMGR 9 audit is run only when invoked manually.

```
AUD:MMGR=a[:b|:c|:b,c];
```

where,

a = Specifies member number of the audit. Valid values are:

1	= Segment descriptor table audit.
2	= Page table audit.
3	= Page descriptor table audit.
4	= Segment release audit.
5	= Segment unlock audit.
9	= Swap space compaction audit.

b = Indicates whether audit should correct errors. Valid values are:

CORR	= Audit should correct errors (default).
NCG	= Audit should not correct errors.

c = Type of report generated. Valid values are:

DETL	= Generates a raw data report for each error found followed by an audit completion message (default). A limited number of reports are generated.
SUM	= Generates only an audit completion message when the audit terminates. Raw data reports are not created for any errors that are found.

1.2 Interpreting AUD:MMGR Output Messages and Error Reports

The AUD MMGR output message reports any errors found during the execution of MMGR audits. For a detailed description of these errors, refer to the AUD:MMGR output message in the Output Messages manual (235-600-750).

Table 1 lists the MMGR audit error messages and their associated categories.

The MMGR audit uses six error categories:

- Category 0 - Used only for the swap space compaction audit (AUD:MMGR9).
- Category 1 - Resets page table (PGT), PDE, and SDE status flags, corrects global free accounts, and corrects SDE structure values.
- Category 2 - Adds unreferenced PDEs/PGTs to free lists.
- Category 3 - Corrects global pointers and counts, releases SDEs and associated resources, corrects PDE structure values.
- Category 4 - Rebuilds SDE, PGT, and PDE free lists. For segments with invalid page entries, page overlaps, and for segment overlaps, the condition is cleared and associated killable processes are terminated.
- Category 5 - Assigned to messages which are not currently in use. It has no real meaning at this time.

Table 1 MMGR AUDIT ERROR MESSAGES

Category	Error Number	Meaning
0	1	Indicates that the MMGR 9 audit has completed the compaction of swap space. Not a real error.
0	2	The MMGR 9 audit was invoked in noncorrecting mode which prevents the audit from doing anything.
0	3	The MMGR 9 audit was unable to move a particular segment. Therefore, it was unable to complete the swap space compaction.
3	10	One or more of the following globals contain incorrect values: pointer to first segment descriptor, pointer to last segment descriptor, number of segment descriptors.
3	20	One or more of the following globals contain incorrect values: pointer to first page table, pointer to last page table, number of page tables.
3	30	One or more of the following globals contain incorrect values: pointer to first page descriptor, pointer to last page descriptor, number of page descriptors.
4	101	The SDE free list contains an invalid link.
4	102	An SDE free list member is allocated.
1	103	The global count of free SDEs contains an incorrect value.
4	104	The SDE free list is circular.
1	105	The on-free-list bit is not set in an SDE that is on the free list.
3	106	An SDE with zero status and not linked to any page table is not on the free list.
1	107	An SDE with non-zero status is not marked allocated.
1	108	An SDE is linked to an empty page table.
1	109	A page table linked to an SDE is not marked allocated.
1	110	An SDE contains an invalid page table pointer.
3	112	The PDE for a page linked to an SDE does not contain the correct segment index.
3	113	The PDE for a page linked to an SDE does not contain the correct page index.
1	114	The in-memory page count in an SDE is incorrect.
1	115	The PDE for a page linked to an SDE is not marked allocated.
4	116	The page table for a segment contains an invalid page address. No process that has this segment in its address space is nonkillable.
4	118	A segment contains the same physical page of memory at two different virtual addresses. No process that has this segment in its address space is nonkillable.
4	120	Two or more in-memory segments contain the same physical page of memory. No process that has the segment(s) in its address space is nonkillable.
4	122	Two or more in-memory segments have the same page table and, therefore, contain the same pages of memory. No process that has the segment(s) in its address space is nonkillable.

4	201	The list of free page tables contains an invalid link.
4	202	A page table on the free list is allocated.
1	203	The global count of free page tables contains an incorrect value.
4	204	The linked list of free page tables is circular.
1	205	The on-free-list bit is not set in a page table that is on the free list.
2	206	A page table is neither linked to an SDE nor is on the free list.
4	301	The list of free pages contains an invalid link index.
4	302	The PDE for a page on the free list is allocated.
1	303	The global count of free pages contains an incorrect value.
4	304	The free page list is circular.
1	305	The on-free-list bit is not set in the PDE for a page that is on the free list.
2	306	A page is not linked to an SDE nor is on the free list.
3	401	The breakdown (release) of a segment did not complete.
3	501	A segment has been IO-locked for 30 minutes or more.

1.3 Responding to MMGR Audit Reports

All the MMGR audits are correcting audits. Unless run in noncorrecting mode, they correct all the errors they find with the exception of those in category 5.

The errors in category 5 involve corruption of memory segments accessed by nonkillable processes. The only way to remove these errors is a level 2 or above phase. This severe recovery action may not be desirable for an application or at all periods of time. Therefore, it is recommended that, in the database audit records, the error threshold be set at 1 (a single such error), and the recovery action be set to RPTAIM (report the audit results to ASM). The AIM process can then choose a recovery appropriate to the current conditions. It may ignore the error, call a phase 2, or postpone a phase to a convenient time.

MODHBLF

Software Release: 5E10 and later

1. SWITCHING MODULE LINK HEADER BUSY LINE FIELD (MODHBLF) AUDIT

The switching module link header busy line field (MODHBLF) relation, RLMODH_BLF, contains one tuple for a given routing module for each 100's group being actively monitored by one or more attendant console.

The MODHBLF audit verifies the idle and busy list linkages for the RLMODH_BLF relation. The MODHBLF audit cycles through the dynamic relation RLMODH_BLF, one tuple at a time using the DRINASEK audit driver. For each tuple, the linkages to the idle or the busy list are checked since the tuple must be linked to one of these lists. When the tuple is on the busy list, linked to a rIMODM_BLF tuple, the owner linkage is verified. If the tuple is on the idle list (linked to rIDBHEADC), the owner linkage is expected to be zero. In addition, the key stored within the tuple will be checked for correctness. If any of the checks fail an error will be printed, and the RLMODH_BLF tuple will be placed on the idle list.

The MODHBLF audit resides in the AUatfeat module. It will run in the OSDS-M environment.

2. DATA STRUCTURES

For details on the fields contained in the RLMODH_BLF relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

The MODHBLF audit verifies owner-member linkages between the RLMODH_BLF and RLMODM_BLF relations or between the RLMODH_BLF and RLDBHEADC, the tuple must be on one of these lists. The audit also checks for a corresponding RLREPLY_BLF for each RLMODH_BLF relation.

4. ERROR HANDLING

The semantic function checks the owner of the rIMODH_BLF tuple. If the owner is the idle tuple of RLDBHEADC, a check is made to determine if the setmember field in RLMODH_BLF is not zero. If it is not zero, an error is printed and the idle function is called to return the correct idle list to the driver and schedule the MODHBLF and DB_HEADC audits.

The setowner linkage is also checked and if the tuple is on the busy list, the head and tail keys are checked for non-zero. If either is equal to zero, an error is printed and the idle function is called, which will return the correct idle list to the driver for idling. The MODHBLF and DB_HEADC audits will be scheduled.

If the associated rIREPLY_BLF tuple is not present, the audit calls the idle function to return to the driver the correct idle list on which to idle the rIMODH_BLF tuple.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALINK

Error Code:	ALINK
Error Description:	The hgid in the associated rIREPLYBLF tuple is not consistent with what is in rIMODH_BLF.
Possible Error Effect:	The rIREPLYBLF tuple has been garbaged and may show invalid status for this monitored 100's group.

Corrective Action Taken:	Report the error on the ROP and return to the driver to put the tuple on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIMODH_BLF.
<i>Bad Data</i> =	hgid of rIREPLYBLF.
<i>Good Data</i> =	Key of rIMODH_BLF.
<i>Logical Index</i> =	Logical Key of rIMODH_BLF.
<i>Dump</i> =	rIMODH_BLF tuple.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	This tuple is not on the idle list, the owner linkage is zero but the member linkage is also zero.
Possible Error Effect:	The rIMODH_BLF tuple has been garbaged, you cannot trust the data in this tuple.
Corrective Action Taken:	Report the error on the ROP and return to the driver to put the tuple on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIMODH_BLF.
<i>Bad Data</i> =	Member linkage to rIMODH_BLF.
<i>Good Data</i> =	Owner linkage.
<i>Logical Index</i> =	Logical key of rIMODH_BLF.
<i>Dump</i> =	rIMODH_BLF tuple.

Table 3 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rIMODH_BLF is supposed to be idle but is not linked to the rIDBHEADC idle list.
Possible Error Effect:	The rIMODH_BLF is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rIMODH_BLF tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMODH_BLF.
<i>Bad Data</i> =	Owner link of rIMODH_BLF.
<i>Good Data</i> =	Physical key of idle head cell.
<i>Logical Index</i> =	Logical key of rIMODH_BLF.
<i>Dump</i> =	rIMODH_BLF tuple.

Table 4 Error Code: PTR1

Error Code:	PTR1
Error Description:	The pointer to the associated rIREPLYBLF tuple is invalid.
Possible Error Effect:	The MODH_BLF audit will get the wrong data.
Corrective Action Taken:	Report the error on the ROP and return to the driver to put the tuple on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIMODH_BLF.
<i>Bad Data</i> =	Key of rIREPLYBLF.
<i>Good Data</i> =	Owner linkage.
<i>Logical Index</i> =	Logical key of rIMODH_BLF.
<i>Dump</i> =	rIMODH_BLF tuple.

Table 5 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rIMODH_BLF is in the idle state but some of its linkages involved in the idle list mappoint are nonzero.
Possible Error Effect:	A resource is linked to a rIMODH_BLF on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rIMODH_BLF tuple linkages involved.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMODH_BLF.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIMODH_BLF.
<i>Dump</i> =	rIMODH_BLF tuple.

MODMBLF

Software Release: 5E10 and later

1. SWITCHING MODULE LINK MEMBER BUSY LINE FIELD (MODMBLF) AUDIT

The RLMODM-BLF relation contains one tuple for each attendant console residing module actively requesting 100's group busy lamp feature status in the 100's group DCP routing module.

The switching module link member busy line field (MODMBLF) audit verifies setmember linkages between the RLMODMBLF and the busy RLMODHBLF relations or between RLMODMBLF and the idle RLMODHBLF relation.

The MODMBLF audit cycles through the dynamic relation RLMODM_BLF on a tuple-by-tuple basis. For each tuple the audit will ensure the correctness of the linkages to the RLMODH_BLF. In addition, the key stored within the tuple will be checked for correctness. If any of the checks fails an error will be printed, and the rIMODM_BLF tuple will be placed on the idle list.

The MODMBLF audit cycles through the dynamic relation RLMODM_BLF on a tuple-by-tuple basis. For each tuple the audit will ensure the correctness of the linkages to the RLMODH_BLF. In addition, the key stored within the tuple will be checked for correctness. If any of the checks fails an error will be printed, and the rIMODM_BLF tuple will be placed on the idle list.

The MODMBLF audit resides in the AUatfeat module. It will run in the OSDS-M environment, will be part of the BASE package.

2. DATA STRUCTURES

For details on the fields contained in the RLMODM_BLF relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

The MODMBLF audit calls the audit driver, AUdrinasek using the interface function. The driver performs the linkage checks and calls the idle function to place the tuple on the proper idle list if an error is found. If no error is found the next tuple is audited.

4. ERROR HANDLING

The MODMBLF audit calls the AUdrinaek driver to check the linkages to the busy and idle rIMODHBLF tuples.

When the setowner linkage is checked and the rIMODMBLF tuple is linked to the busy tuple of the RLMODHBLF relation or is linked to an idle tuple of the rIMODHBLF tuple and an error is found, the idle function will be called and the rIMODHBLF tuple will be recovered to the idle list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

22. MP - N

MPHDATA

Software Release: 5E10 and later

1. MSU PATH DATA (MPHDATA) AUDIT

The metallic service units (MSUs) provide a collection of functions that may be shared among a number of switching modules (SMs).

Metallic interconnection buses (MTIBs) interconnect one MSU to another MSU in a different SM. Since the MTIBs are global to all SMs, they are allocated from the communications module processor (CMP). The MSU path data relation (RLMPHDATA), which resides in the CMP, is used to globally allocate the MTIBs. The MPHDATA audit ensures the integrity of RLMPHDATA.

This audit is part of the peripheral control audits subgroup. It runs in the CMP in the segmented mode.

2. DATA STRUCTURES

The metallic service unit path data relation (rIMPHDATA) contains information about the status of the metallic interconnection bus. Figure 1 shows the relationship between rIMPHDATA and the administrative module equipment status table relation (RLCPEST). The RLCPEST relation contains the in-service or out-of-service status of the MTIB. It is audited by the CPEST audit.

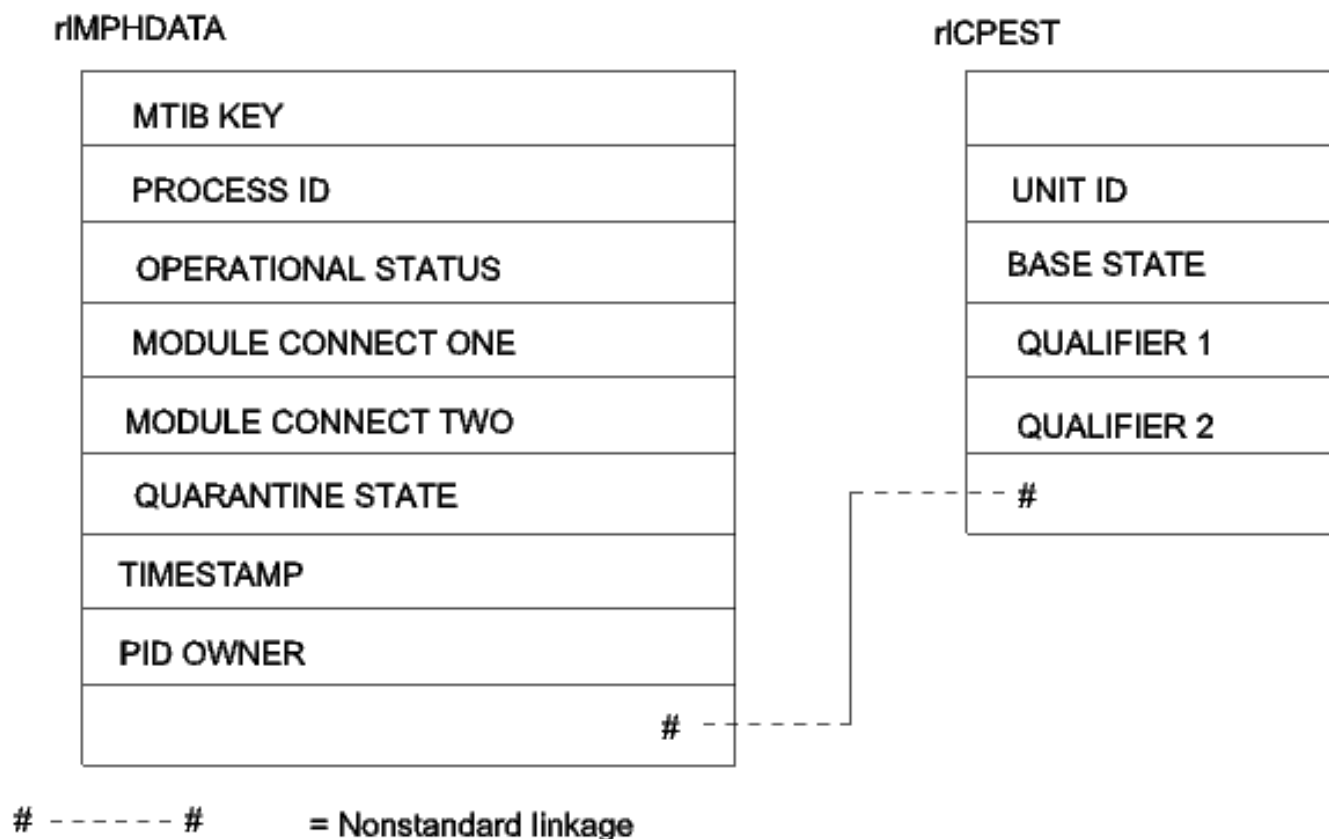


Figure 1 (MPHDATA-1) Metallic Path Data Relation

3. AUDIT CHECKS

The metallic service unit path data audit uses the dynamic linkage driver (DRLINK) to step through the

RLMPHDATA relation. The DRLINK driver calls a semantic check function to perform the following audit checks: If the rIMPHDATA tuple is idle (op_status = AUMPHIDLE):

IDLE_BUSY = Check if all fields are set to an idle state.

If the rIMPHDATA tuple is busy:

TIMESTAMP = Check if a metallic connection has been up to long.

MOD_1CONN = Check if a rIMUMTIB tuple exists associated with the mod_1conn field.

MOD_2CONN = Check if a rIMUMTIB tuple exists associated with the mod_2conn field.

BUSYIDLE = Check if a valid process owns this rIMPHDATA tuple.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The operational status (op_status) of the rIMPHDATA tuple is busy (not AUMPHIDLE) but the owner pid field (pid_owner) is null (OSnullpid).
Possible Error Effect:	The rIMPHDATA tuple will be lost to the system and may lead to degradation of metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMPHDATA tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple in error.
<i>Dump =</i>	rIMPHDATA tuple found in error.

Table 2 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The operational status (op_status) of the rIMPHDATA tuple is idle (AUMPHIDLE) but either the time stamp field (timestamp) is not set to 0, the owning process identification (pid_owner) is not set to OSnullpid, the queueing process identification (queproc) is not set to OSnullpid, the module 1 connect field (mod_1conn) is not set to 0, the module 2 connect field is not set to 0, the queue bit (quefor) is not set to DBNO, or the queueing module connect field (qmod_conn2) is not set to 0.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMPHDATA tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple in error.
<i>Dump =</i>	rIMPHDATA tuple found in error.

Table 3 Error Code: MOD_1CONN

Error Code:	MOD_1CONN
Error Description:	The operational status (op_status) of the rIMPHDATA tuple is busy (not AUMPHIDLE) but the module connect one field (mod_1conn) is either idle or contains an SM that does not contain a MSU that is connected to a MTIB.
Possible Error Effect:	The rIMPHDATA tuple will be lost to the system and may lead to degradation of metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The module connect one field (mod_1conn) from the rIMPHDATA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple in error.
<i>Dump =</i>	rIMPHDATA tuple found in error.

Table 4 Error Code: MOD_2CONN

Error Code:	MOD_2CONN
Error Description:	The operational status (op_status) of the rIMPHDATA tuple is busy (not AUMPHIDLE) but the module connect two field (mod_2conn) is either idle or contains an SM that does not contain a MSU that is connected to a MTIB.
Possible Error Effect:	The rIMPHDATA tuple will be lost to the system and may lead to degradation of metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The module connect two field (mod_2conn) from the rIMPHDATA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple in error.
<i>Dump =</i>	rIMPHDATA tuple found in error.

Table 5 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The timestamp field (timestamp) in the rIMPHDATA tuple indicates that the metallic connection associated has been up for longer than 3 hours.
Possible Error Effect:	The rIMPHDATA tuple will be lost to the system and may lead to degradation of metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIMPHDATA tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple in error.
<i>Dump 1 =</i>	rIMPHDATA tuple found in error.
<i>Dump 2 =</i>	Current time in the system (OSnow).

MSALWSG

Software Release: 5E10 and later

1. MEASUREMENTS ALLOW STUDY GROUP (MSALWSG) AUDIT

The MSALWSG audit verifies the extra tuples in the dynamic relation RLms_alwsg and also does data comparisons between the dynamic RLMS_ALWSG tuple and the static measurements line unit study (RLMS_LUSDY) tuple. This dynamic relation is used for the integrated switch line unit/line unit (ISLU/LU) load administration measurements feature. This feature will assist the telephone companies in effective load administration and load balancing. It will provide traffic usage counts on all operational switch groups (SG) and line group controllers (LGC) on telephone company specified operational LUs and ISLUs. The counts will be generated in the switching modules (SMs) and reported on two new sections of the 30-minute traffic report (TRFC30); section 144 will report usage on per-LGC basis and section 145 will report usage on per-SG basis. A new recent change/verify view will be provided to enable the telephone company to specify the LUs and ISLUs to be studied. This will be part of the BASE package.

2. DATA STRUCTURES

The primary data structure for this audit is the RLMS_ALWSG relation. This relation identifies switching module (SM) line units (LUs) [integrated switch line units (ISLUs)] allowed for study. This is a dynamic copy of the static relation RLMS_LUSDY.

3. AUDIT CHECKS

The MSALWSG audit invokes the DRSTATIC driver to step through the rIMS_ALWSG tuples. It also uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code, see the comment blocks given below.

EXTRATUP = Check to see if an extra tuple of the relation exists.

GRBG = Check if the "ids" [LU (RISLU/ISLU) identifier] field is consistent with the corresponding field in the static relation RLMS_LUSDY.

KEY = Check the key of the relation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit did not find a static tuple of type rIMS_LUSDY when it tried to access that tuple from dynamic rIMS_ALWSG tuple. This means that this dynamic tuple is an extra tuple.
Possible Error Effect:	The result will be erroneous which will cause erroneous identification of non-existent LUs/ISLUs.
Corrective Action Taken:	The audit will report the error and will delete the extra dynamic tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the dynamic tuple buffer.
<i>Bad Data =</i>	0. There is no bad data.
<i>Good Data =</i>	0. There is no good data.
<i>Logical Index =</i>	The value of the key field of the extra dynamic tuple.

<i>Dump</i> =	The extra dynamic tuple of type rIMS_ALWSG.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	The key field - line unit type and the study group number of the current dynamic tuple rIMS_ALWSG does not match the same field in the static tuple of type rIMS_LUSDY.
Possible Error Effect:	The dynamic tuple has wrong information. This can lead to identification of the wrong line unit type and study group number allowed for study.
Corrective Action Taken:	The audit will report the error and will correct the line unit type and the study group number field of the dynamic tuple from the static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the dynamic tuple buffer.
<i>Bad Data</i> =	The line unit type and the study group number of the dynamic tuple being audited.
<i>Good Data</i> =	The key field of the dynamic tuple buffer.
<i>Logical Index</i> =	The line unit type and the study group number of the dynamic tuple being audited.
<i>Dump</i> =	The dynamic tuple of type rIMS_ALWSG being audited.

Table 3 Error Code: GRBG

Error Code:	GRBG
Error Description:	The LU (ISLU) identifier of the current dynamic tuple rIMS_ALWSG does not match the same field in the static tuple of type rIMS_LUSDY.
Possible Error Effect:	The dynamic tuple has wrong information. This can lead to false identification of SMs and LUs (ISLUs).
Corrective Action Taken:	The audit will report the error and will correct the LU (ISLU) identifier field of the dynamic tuple from the static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the dynamic tuple buffer.
<i>Bad Data</i> =	The wrong line unit of the dynamic tuple.
<i>Good Data</i> =	0. There is no good data.
<i>Logical Index</i> =	The value of the key field of the dynamic tuple.
<i>Dump</i> =	The dynamic tuple of type rIMS_ALWSG being audited. The static tuple rIMS_LUSDY.

MSGBUF

Software Release: 5E10 and later

1. MESSAGE BUFFER (MSGBUF) AUDIT

The message buffer (MSGBUF) audit verifies the allocation of the units carrying messages among the active operating system processes. MSGBUF has two audits (called members); one is called the message buffer queue audit and the other is called the message buffer extender block audit. In most cases, the audit corrects the errors it finds.

MSGBUF is routinely initiated by SIM at intervals that depend on the number of other audits scheduled. Either member can also be initiated by the user.

Only one member of MSGBUF can be requested at a time. If you want to run both members, input two requests.

The MSGBUF audit is a frequency-group audit that runs in the correcting mode.

1.1 Requesting MSGBUF Audits

To request an audit of the message buffers, use the following input message.

```
AUD:MSGBUF=a[:b|:c|:b,c];
```

where:

a	= The member number of the audit. Valid values are:
1	= Specifies a message buffer queue audit;
2	= Specifies a message buffer extender block audit.
b	= Indicates whether audit should correct errors. Valid values are:
CORR	= Audit should correct errors (default).
NCG	= Audit should not correct errors.
c	= Indicates what type of report. Valid values are:
DETL	= Generates a raw data report for each error found and an audit completion message (default). Only a limited number of raw data reports are generated.
SUM	= Generates only an audit completion message when the audit terminates. Raw data reports are not created for any errors found.

To interpret system responses to the AUD:MSGBUF input message, refer to the APP:AUD appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD MSGBUF Output Messages and Error Reports

The MSGBUF audit prints an output message after each audit. It is produced by the system integrity output formatter and includes the information requested by the input message. For more detailed information about the

AUD:MSGBUF output message, refer to the Output Messages manual (235-600-750).

If MSGBUF found an error, a code number is output. Table 1 gives the error numbers and their meanings.

For the message buffer queue audit (error number series 100):

- Category 0 - Errors corrected by updating a field in either a message buffer, a message extender, or a process dispatcher control table entry (DCTE).
- Category 1 - Errors corrected by: removing a message from a queue and returning it to the original sender or requeueing it, or terminating a process.
- Category 2 - Not used.
- Category 3 - Not used.
- Category 4 - Not used.
- Category 5 - Errors corrected by rebuilding a process message queue.

For the message buffer extender block audit (error number series 200):

- Category 0 - Errors corrected by updating a field in a message buffer, freeing a message or returning a message to the original sender, or invalid data returned to an audit fault message.
- Category 1 - Error corrected by returning message to original sender.
- Category 2 - Error corrected by freeing a nonqueued message
 - owned by supervisor process,
 - owned by kernel-audit process,
 - owned by the kernel,
 - in response to audit-fault message.
- Category 3 - Errors corrected by faulting a process or terminating a killable process.
- Category 4 - Not used.
- Category 5 - Not used.

Table 1 MSGBUF AUDIT ERROR MESSAGES

Category	Error Number	Meaning
5	110	Bad message queue head pointer.
5	111	Bad message queue link pointer.
5	112	Allocation bit not set in message extender.
0	113	Bad on-queue bit in message extender.
0	114	Allocation bit not set in message header.
1	115	Bad message-to field in message and bad message owner in message extender.
0	116	Bad message owner in message extender.
0	117	Bad message-to field in message.
5	118	Message queue out of order.
1	119	Old audit fault message queued to a nonkillable process.
1	120	Old audit fault message queued to a killable process.
1	121	Old queued message (except an old queued audit fault message or an old acknowledgement

		message queued to a supervisor process).
5	122	Process has a circular message queue.
5	123	Bad message queue tail pointer.
0	124	Bad message queue block count in process DCTE.
2	125	Old acknowledgement message queued to a nonkillable supervisor process.
2	126	Old acknowledgement message queued to a killable supervisor process.
0	200	Allocated message with an invalid owner field.
2	201	Old allocated message.
0	214	Allocation bit not set in message header.
1	216	Queued message with an invalid owner.
3	219	Old audit fault message queued to a nonkillable process.
3	220	Old audit fault message queued to a killable process.
1	221	Old queued message (except an old queued audit fault message).
0	270	Dequeued message with an invalid owner field.
3	271	Old dequeued audit fault message owned by a nonkillable process.
3	272	Old dequeued audit fault message owned by a killable process.
2	273	Old dequeued message (except an audit fault message).
3	274	Over nine old nonqueued messages (excluding audit fault messages) owned by a killable process.
0	290	Audit fault message response has an invalid message status.
0	291	Invalid status for a message pointer in an audit fault message.

1.3 Responding to AUD MSGBUF Error Reports

The MSGBUF audit corrects most errors it finds. Errors that are not corrected and the action that should be taken are listed in Table 2 .

Table 2 SPECIAL ERROR MESSAGES

Error Number	Action To Be Taken
119	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because the error indicates that a process has probably mishandled an audit fault message. It is likely that errors 201 and 273 will not be corrected if any of these errors occur.
120	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because the error indicates that a process has probably mishandled an audit fault message. It is likely that errors 201 and 273 will not be corrected if any of these errors occur.
121	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because the error indicates that a process may be having problems in processing its messages.
125	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because the error indicates that a process has probably mishandled an acknowledgement message.
126	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because the error indicates that a process has probably mishandled an acknowledgement message.
200	The error indicates a problem in the ownership of a message. This may occur if a breakdown happens in a process termination. Run the process administration audit (PROAD 1) to complete the process termination. Refer to the AUD:PROAD input message in the Input Messages manual (235-600-700).
201	Refer to the TECHNICAL ASSISTANCE of the INTRODUCTION section of the Audits manual, because the error indicates that a process may be having problems in processing its messages.
216	The error indicates a problem in the ownership of a message. This may occur if a breakdown happens in a process termination. Run the process administration audit (PROAD 1) to complete the process termination. Refer to the AUD:PROAD input message in the Input Messages manual (235-600-700).
219	This error indicates an error in a queued audit fault message found by the message extender audit. Run the message queue audit (MSGBUF 1). Refer to the AUD:MSGBUF input message in the Input Messages manual (235-600-700). If the error is not found and corrected, then run the message extender audit (MSGBUF 2) again. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual. It is likely that errors 201 and 273 will not be corrected if this error occurs.
220	This error indicates an error in a queued audit fault message found by the message extender audit. Run the message queue audit (MSGBUF 1). Refer to the AUD:MSGBUF input message in the Input Messages manual (235-600-700). If the error is not found and corrected, then run the message extender audit (MSGBUF 2) again. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual. It is likely that errors 201 and 273 will not be corrected if this error occurs.
221	This error indicates an error in a queued message found by the message extender audit. Run the message queue audit (MSGBUF 1) and if the error is not found and corrected, then run the message

	extender audit (MSGBUF 2) again. Refer to the AUD:MSGBUF input message in the Input Messages manual (235-600-700).
270	This error indicates a problem in the ownership of a message. This may occur if a breakdown happens in a process termination. Run the process administration audit (PROAD 1) to complete the process termination. Refer to the AUD:PROAD input message in the Input Messages manual (235-600-700).
271	Refer to the TECHNICAL ASSISTANCE of the INTRODUCTION section of the Audits manual, because this error indicates that a process has probably mishandled an audit fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
272	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because this error indicates that a process has probably mishandled an audit fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
273	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because this error indicates that a process may be having problems in processing its messages.
274	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because the error indicates that a process may be having problems in processing its messages.
290	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because this error indicates that a process has probably mishandled an audit fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
291	Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, because this error indicates that a process has probably mishandled an audit fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.

MSLUSDY

Software Release: 5E10 and later

1. MEASUREMENTS LINE UNIT STUDY (MSLUSDY)

The MSLUSDY audit performs static to dynamic existence checks between the static rIMS_LUSDY and the dynamic measurements allow study group (rIMS_ALWSG) tuples. This static relation is used for the integrated switch line unit/line unit (ISLU/LU) load administration measurements feature. This feature will assist the telephone companies in effective load administration and load balancing. This capability will provide traffic usage counts on all operational switch groups (SG) and line group controllers (LGC) on telephone company-specified operational LUs and ISLUs. The counts will be generated in the switching modules (SM) and reported on two new sections of the 30-minute traffic report (TRFC30); section 144 will report usage on per-LGC basis and section 145 will report usage on per-SG basis. A new recent change/verify view will be provided to enable the telephone company to specify the LUs and ISLUs to be studied. This feature will be part of the BASE package.

2. DATA STRUCTURES

The primary data structure for this audit is the RLMS_LUSDY relation. This relation identifies SM and line units (LU) [integrated switch line units (ISLUs)] allowed for study.

3. AUDIT CHECKS

The MSLUSDY audit invokes the DRSTATIC driver to step through the rIMS_LUSDY tuples.

LOSTTUP = Check to see if the tuple of a dynamic relation is lost.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit did not find a dynamic tuple of type rIMS_ALWSG when it tried to access that tuple from static rIMS_LUSDY tuple. There is a missing tuple from the dynamic relation.
Possible Error Effect:	This will cause lost resources as there will be reduced number of rIMS_ALWSG tuples.
Corrective Action Taken:	The audit will report the error and will insert the lost dynamic tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the static tuple buffer.
<i>Bad Data =</i>	0. There is no bad data.
<i>Good Data =</i>	0. There is no good data.
<i>Logical Index =</i>	The value of the key field of the static tuple.
<i>Dump =</i>	The static tuple of type rIMS_LUSDY.

MSMWIDA

Software Release: 5E10 and later

1. MESSAGE SERVICES MESSAGE WAITING INDICATOR DATA (MSMWIDA) AUDIT

The message services message waiting indicator data (MSMWIDA) audit verifies the RLMSMWIDA relation, used for the inter switch voice message (ISVM) feature. The ISVM allows voice message (VM) providers to expand their current intra-switch customer base to potential VM customers by other switches within the same local access transport area (LATA). As such, the crux of the feature is to provide switch support for the necessary inter-switch signaling. The feature does not seek to change or modify existing VM systems, or their interface to the 5ESS[®] switch.

2. DATA STRUCTURES

The primary data structure for this audit is the RLMSMWIDA relation. It is a dynamic, indirect access sequential keyed relation which contains call history information associated with a message waiting indicator (MWI) act/deact request. All idle rLMSMWIDA tuples must be linked with owner-member linkages to the idle list owned by rIDB_HEADC tuple, as shown in Figure 1. All busy rLMSMWIDA tuples must be linked one-to-one to rIDB_HEADC tuples. The global parameter Glapisvm indicates the number of rLMSMWIDA tuples to be populated during initialization.

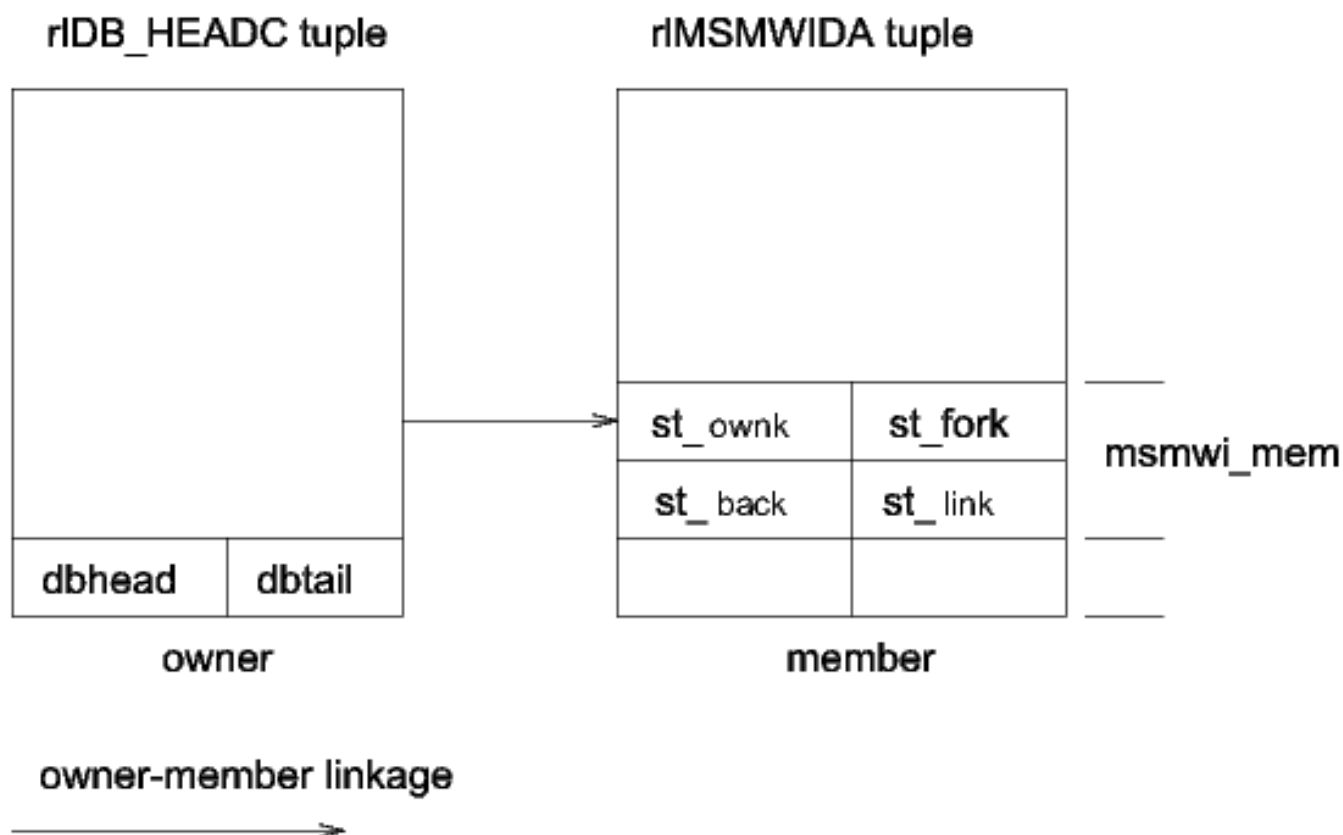


Figure 1 ((MSMWIDA-1) Owner-member Linkage with rIDB_HEADC Tuple

3. AUDIT CHECKS

The MSMWIDA audit invokes the DRINASEK driver to step through the RLMSMWIDA relation, verifying linkages with the relation RLDB_HEADC.

4. ERROR HANDLING

If linkage errors are detected, the rIMSMWIDA tuple is recovered to the idle list of rIDB_HEADC after all fields (except the key) are zeroed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes.

MSQ

Software Release: 5E10 and later

1. MESSAGE QUEUE (MSQ) AUDIT

The switch software provides a mechanism for reporting output messages based on their priority. This mechanism, the message queue, consists of several head cell lists. Each message scheduled by the system has a priority. The system assigns the message to the appropriate head cell list based on its priority. All messages pass through the message queues to the human/machine interface subsystem, which prints or logs them appropriately. The system uses the message queues to print the most important messages first. The only time the queues are circumvented is when they are being audited. When the queues are being audited, the system passes messages directly to the human/machine interface subsystem for assignment to their destinations, without regarding their priorities.

The audit runs in the OSDS kernel process (OKP) environment in the administrative module (AM), the switching module (SM) and the communications module processor (CMP).

2. DATA STRUCTURES

Each processor contains a message queue relation (RLMSQ). The MSQ audit verifies linkages within the RLMSQ relation, and also makes checks between this relation and the data base head cell relation (RLDB_HEADC). The message queue relations contain several lists of messages. There is a separate list for each priority of message. Refer to Figure 1. The priority of a message is essentially its alarm status, which prints in the priority of action field on the output message. Refer to the Output Messages manual, 235-600-750. The message queue relation therefore contains separate lists for messages with critical, major, minor, manual, and automatic alarms. It also contains queues for various priorities of unalarmed messages and an idle list.

The tuples of the message queue relation may not be large enough to contain the text of an entire output message. Some messages require two tuples (see Figure 1 for an example). In this case, the tuples are linked by a one-to-one linkage ("mate" linkage). But no message will ever require more than two tuples.

In addition to the RLMSQ and RLDB_HEADC relations, the MSQ audit uses the global indicator SImsqlock. The audits uses SImsqlock to stabilize the queues during auditing.

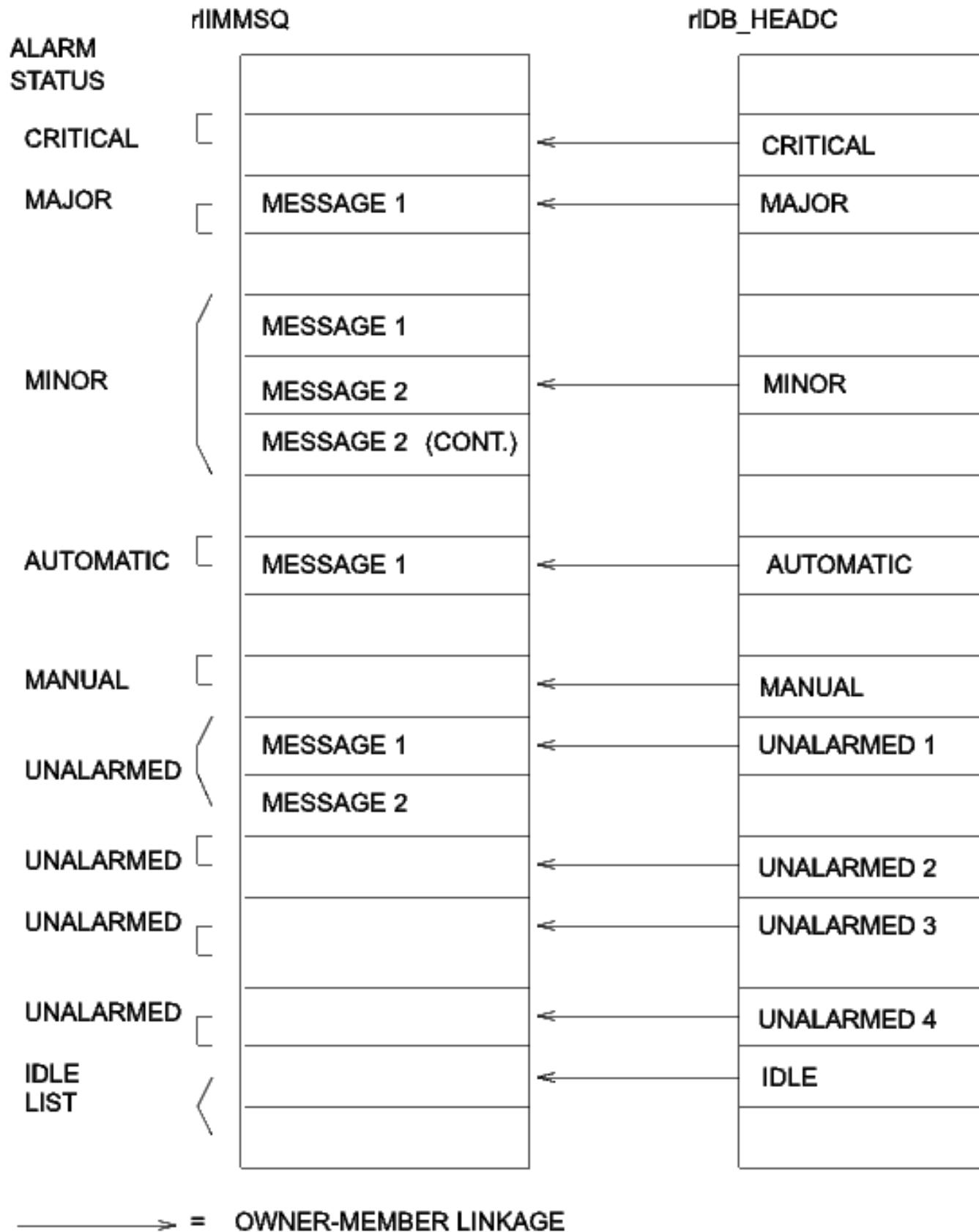


Figure 1 (MSQ-1) Message Queue Lists

3. AUDIT CHECKS

The MSQ audit invokes the DRLINK driver to check the relation's linkages.

For each tuple, the MSQ audit makes the following checks after the driver has verified the linkages:

IDLE_LINK = Check if the tuple should be on the idle list.

MATELINK = Check whether or not a mate tuple should be linked.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The length field in the OSMSGHEAD field of the output message is zero and the rIMSQ.mate.link field is empty (0), indicating an idle tuple, but the tuple was not on the idle list.
Possible Error Effect:	RLMSQ tuples may be lost to the system. This could result in delayed printing, logging of output messages, or even lost messages.
Corrective Action Taken:	The audit reinitializes the rIMSQ tuple and links it on to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The rIMSQ tuple's address.
<i>Bad Data</i> =	Owner linkage from the rIMSQ tuple to the owning rIDB HEADC tuple.
<i>Good Data</i> =	Idle list head cell key.
<i>Logical Index</i> =	Logical key of the rIMSQ tuple.
<i>Dump</i> =	The rIMSQ tuple.

Table 2 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	Either the length field in the OSMSGHEAD field of the output message was greater than the maximum allowable for a single tuple and the rIMSQ.mate.link field was 0, or the length was less than the maximum allowable for one tuple but the rIMSQ.mate.link field was not 0. That is, either the length indicated that a mate tuple should be linked but it wasn't, or the length indicated that no mate was necessary but one was there.
Possible Error Effect:	An incorrect message length field would result in the following: (a) The output message being truncated when sent to HMsip for printing. (b) The message length being too long for OSsendmsg to handle.
Corrective Action Taken:	The audit reinitializes the message queue block and puts it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The rIMSQ tuple's address.
<i>Bad Data</i> =	Message length from the rIMSQ tuple.
<i>Good Data</i> =	Message length from the mate tuple (if any).
<i>Logical Index</i> =	Logical key of the rIMSQ tuple.

<i>Dump 1</i> =	The rMSQ tuple.
<i>Dump 2</i> =	The mate rMSQ tuple (if any).

MSSGCNT

Software Release: 5E10 and later

1. MEASUREMENTS SWITCH GROUP COUNT (MSSGCNT) AUDIT

The MSSGCNT audit verifies the extra tuples in the dynamic relation RLMS_SGCNT and also does data comparisons between the dynamic rLMS_SGCNT tuple and the static measurements switch group equipped (rLMS_SGEQ) tuple. This dynamic relation is used for the integrated switch line unit/line unit (ISLU/LU) load administration measurements feature. This feature will assist the telephone companies in effective load administration and load balancing. It will provide traffic usage counts on all operational switch groups (SG) and line group controllers (LGC) on telephone company-specified operational LUs and ISLUs. The counts will be generated in the switching modules (SMs) and reported on two new sections of the 30-minute traffic report (TRFC30); section 144 will report usage on per-LGC basis and section 145 will report usage on per-SG basis. A new recent change/verify view will be provided to enable the telephone company to specify the LUs and ISLUs to be studied. This will be part of the BASE package.

2. DATA STRUCTURES

The primary data structure for this audit is the RLMS_SGCNT relation. This relation counts the number of operational switch groups (SGs) or line group controllers (LGCs) on a line unit (LU) or remote integrated switch line unit (RISLU) or integrated switch line unit (ISLU). This is dynamic copy of the static relation RLMS_SGEQ.

3. AUDIT CHECKS

The MSSGCNT audit invokes the DRSTATIC driver to step through the rLMS_SGCNT tuples. This listing gives a basic indication of the order of the checks.

- EXTRATUP = Check to see if an extra tuple of the relation exists.
- GRBG = Check if the "lu_sgcnt" [number of SGs (LGCs) equipped on LU (RISLU/ISLU)] field is consistent with the corresponding field in the static relation RLMS_SGEQ).
- KEY = Check the key of the relation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit did not find a static rLMS_SGEQ tuple when it tried to access that tuple from dynamic rLMS_SGCNT tuple. This means that this dynamic tuple is an extra tuple.
Possible Error Effect:	The number of operational SGs (LGCs) on a LU (RISLU/ISLU) will be erroneous.
Corrective Action Taken:	The audit will report the error and will delete the extra dynamic tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the dynamic tuple buffer.
<i>Bad Data =</i>	0. There is no bad data.
<i>Good Data =</i>	0. There is no good data.
<i>Logical Index =</i>	The value of the key field of the extra dynamic tuple.

<i>Dump</i> =	The extra dynamic tuple of type rIMS_SGCNT.

Table 2 Error Code: GRBG

Error Code:	GRBG
Error Description:	The number of SGs (LGCs) equipped on LU ((R)ISLU) of the current dynamic tuple rIMS_SGCNT does not match the same field in the static rIMS_SGEQ tuple.
Possible Error Effect:	The dynamic tuple has wrong information. This can lead to false identification of SGs (LGCs) equipped on LUs (RISLU/ISLU).
Corrective Action Taken:	The audit will report the error and will correct the number of SGs (LGCs) equipped on LUs (RISLU/ISLU) of the dynamic tuple from the static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the dynamic tuple buffer.
<i>Bad Data</i> =	The wrong number of SGs (LGCs) equipped on LUs (RISLU/ISLUs) of the dynamic tuple.
<i>Good Data</i> =	0. There is no good data.
<i>Logical Index</i> =	The value of the key field of the dynamic tuple.
<i>Dump</i> =	The dynamic tuple of type rIMS_SGCNT being audited. The static tuple rIMS_SGEQ.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	The key field - line unit type and the SM number of the current dynamic tuple rIMS_SGCNT does not match the same field in the static tuple of type rIMS_SGEQ.
Possible Error Effect:	The dynamic tuple has wrong information. This can lead to identification of the wrong line unit type and SM number allowed for study.
Corrective Action Taken:	The audit will report the error and will correct the line unit type and the SM number field of the dynamic tuple from the static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the dynamic tuple buffer.
<i>Bad Data</i> =	The line unit type and the SM number of the dynamic tuple being audited.
<i>Good Data</i> =	The key field of the dynamic tuple buffer.
<i>Logical Index</i> =	The line unit type and the SM number of the dynamic tuple being audited.
<i>Dump</i> =	The dynamic rIMS_SGCNT tuple being audited.

MSSGEQ

Software Release: 5E10 and later

1. MEASUREMENTS SWITCH GROUP EQUIPPED (MSSGEQ)

The MSSGEQ audit performs static to dynamic existence checks between the static rIMS_SGEQ and the dynamic measurements switch group count (rIMS_SGCNT) tuples. This static relation is used for the integrated switch line unit/line unit (ISLU/LU) load administration measurements feature. This feature will assist the telephone companies in effective load administration and load balancing. This capability will provide traffic usage counts on all operational switch groups (SG) and line group controllers (LGC) on telephone company-specified operational LUs and ISLUs. The counts will be generated in the switching modules (SM) and reported on two new sections of the 30-minute traffic report (TRFC30); section 144 will report usage on per-LGC basis and section 145 will report usage on per-SG basis. A new recent change/verify view will be provided to enable the telephone company to specify the LUs and ISLUs to be studied. This feature will be part of the BASE package.

2. DATA STRUCTURES

The primary data structure for this audit is the RLMS_SGEQ relation. This relation counts the number of operational switch groups (SGs) and line group controllers (LGC) on a line unit (LU) or remote integrated switch line unit (RISLU) or integrated switch line unit (ISLU).

3. AUDIT CHECKS

The MSSGEQ audit invokes the DRSTATIC driver to step through the rIMS_SGEQ tuples. It also uses the following error code.

LOSTTUP = Check to see if the tuple of a dynamic relation is lost.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit did not find a dynamic rIMS_SGCNT tuple when it tried to access that tuple from the static rIMS_SGEQ tuple. There is a missing tuple from the dynamic relation.
Possible Error Effect:	This will cause lost resources as there will be reduced number of rIMS_SGCNT tuples.
Corrective Action Taken:	The audit will report the error and will insert the lost dynamic tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the static tuple buffer.
<i>Bad Data =</i>	0. There is no bad data.
<i>Good Data =</i>	0. There is no good data.
<i>Logical Index =</i>	The value of the key field of the static tuple.
<i>Dump =</i>	The static tuple of type rIMS_SGEQ.

MSSGLIST

Software Release: 5E10 and later

1. MEASUREMENTS SWITCH GROUP LIST (MSSGLIST) AUDIT

The MSSGLIST audit performs dynamic to static existence checks between the array MSsglist[] and the static relations measurements switch group study (RLMS_SGSDY), circuit data (RLCKTDATA) and peripheral interface control bus (RLPICB). This array is used for the integrated switch line unit/line unit (ISLU/LU) load administration measurements feature. This feature will assist the telephone companies in effective load administration and load balancing. This capability will provide traffic usage counts on all operational switch groups (SG) and line group controllers (LGC) on telephone company specified operational LUs and ISLUs. The counts will be generated in the switching modules (SM) and reported on two new sections of the 30-minute traffic report (TRFC30); section 144 will report usage on per-LGC basis and section 145 will report usage on per-SG basis. A new recent change/verify view will be provided to enable the telephone company to specify the line units (LUs) and integrated switch line unit (ISLUs) to be studied. This feature will be part of the BASE package. The relations and arrays used for the MSSGLIST audit are shown in the Figure 1 .

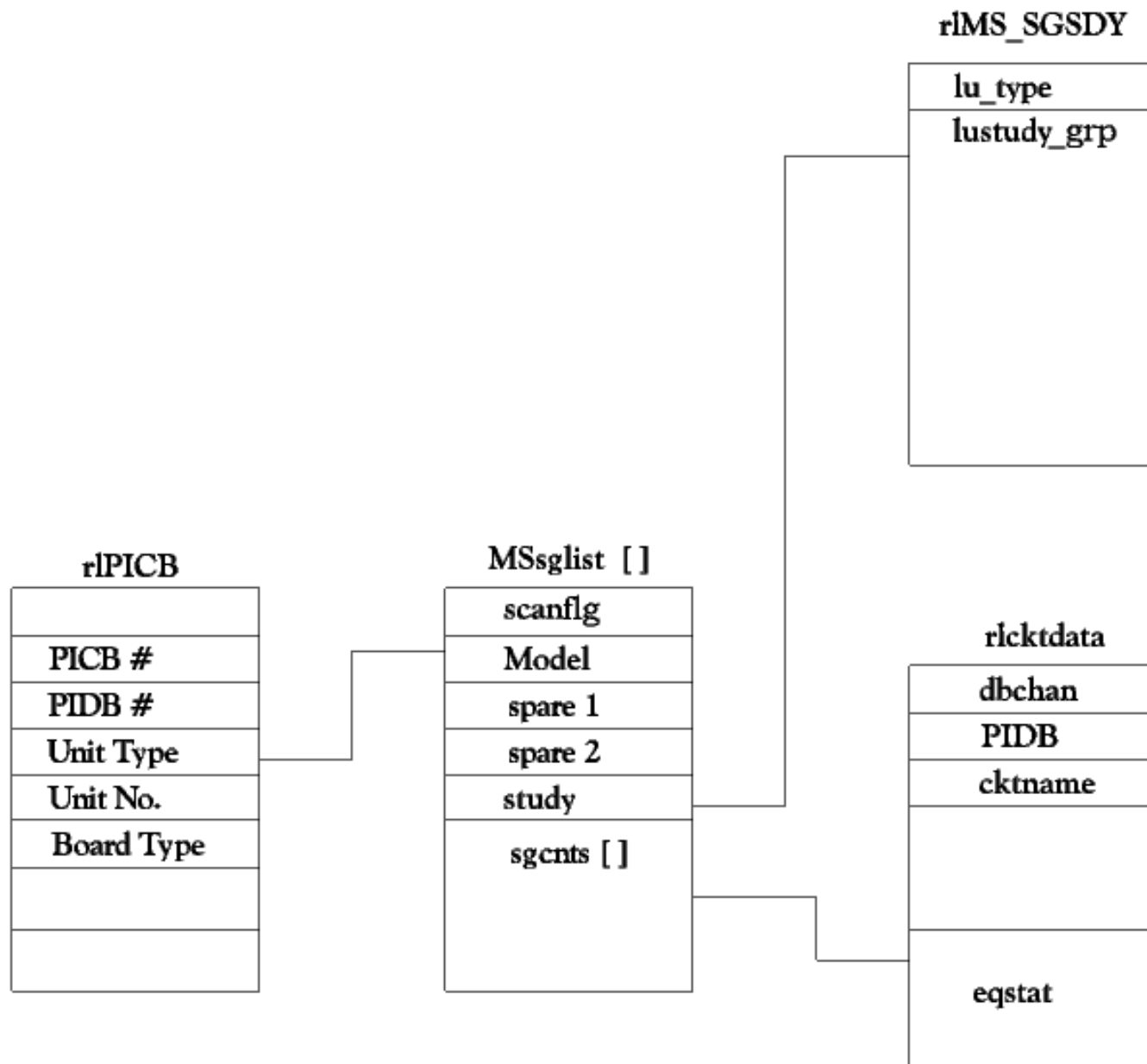


Figure 1 (MSSGLIST-1) Data Structures of the MSSGLIST Audit

2. DATA STRUCTURES

The primary data structure for this audit is the MSSGLIST array. This array is used by the measurement data allocation for the structure type of per SG line group controllers (LGC) for sections 144 and 145, respectively, of the 30-minute traffic report.

3. AUDIT CHECKS

This listing gives a basic indication of the order of the checks. For details regarding each error code, see the comment blocks given below.

BADFLAG = Check to see if "scanflg" field (section 144/145 enumeration values to indicate which SGs

should be scanned and/or collected) field is off.

BADUSE	= Check if the "model" field [structure of LU (ISLU) IDs - it could either be a SMLU, SMC1LU, SMLU3, SMISLU or SMRISLU] field is consistent with the corresponding field in the relation RLPICB.
GRBG	= Check if the "sg" [SG (LGC) identifier] field is consistent with the corresponding field in the static relation RLCKTDATA.
INV_STATE	= Check the validity of the "scanflg" field.
PICB	= Inconsistency between RLPICB and PClusname.
SCNFLG	= Check to see if "scanflg" field is off.
WRONGUSE	= Check to see if the "study" (study groups) field is consistent with the corresponding field in the static relation RLMS_SGSDY.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The scanflag is turned off. This check is done after updating the RLPICB relation and if the line unit is not equipped.
Possible Error Effect:	If the scanflag is turned off, we will be unable to recognize which of the switch groups from the array MSsglist[] should be scanned and/or allowed for collection.
Corrective Action Taken:	The audit will report the error and will turn the scanflag on.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The line unit of the array.
<i>Bad Data</i> =	The scanflag of the line unit under audit.
<i>Good Data</i> =	The value of MSLUOFF (0).
<i>Logical Index</i> =	The value of the key field of the array.
<i>Dump</i> =	NA.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The structure of the line unit IDs (SMLU, SMC1LU, SMLU3, SMISLU or SMRISLU) is not consistent with the corresponding field in the relation RLPICB.
Possible Error Effect:	The array has wrong information. This can lead to identification of the wrong line unit type.
Corrective Action Taken:	The audit will report the error and will correct the line unit type of the array from the static rPICB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the line unit.
<i>Bad Data</i> =	The line unit type field of the array being audited.
<i>Good Data</i> =	The line unit type field of the relation RLPICB.
<i>Logical Index</i> =	The value of the line unit from the array.
<i>Dump</i> =	The line unit of the array.

Table 3 Error Code: GRBG

Error Code:	GRBG
Error Description:	The SG or the LGC identifier is not consistent with the corresponding field in the static relation RLCKTDATA.
Possible Error Effect:	The array has wrong information. We will be obtaining a switch group for which the traffic usage count is not needed.
Corrective Action Taken:	The audit will report the error and will correct the sg field of the array from the static tuple rLCKTDATA.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the line unit of the array.
<i>Bad Data =</i>	The wrong switch group identifier of the array.
<i>Good Data =</i>	The switch group identifier from the static tuple rLCKTDATA.
<i>Logical Index =</i>	The value of the key field of the array.
<i>Dump =</i>	The line unit of the array.

Table 4 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The MSsglist[] .scanflag field is not set properly. It is either set to on when it should be off or set to off when it should be on.
Possible Error Effect:	If the scanflag is turned off, we will be unable to recognize which of the switch groups from the array MSsglist[] should be scanned and/or allowed for collection. If the scanflag is turned on, the switch group will unnecessarily be allowed for collection when the traffic usage count is not needed.
Corrective Action Taken:	The audit will set the scanflag to the appropriate state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0 - if the flag is ON when it should be OFF. .br 1 - if the flag is OFF when it should be ON.
<i>Bad Data =</i>	The invalid MSsglist[] .scanflag value.
<i>Good Data =</i>	The correct MSsglist[] .scanflag value.
<i>Logical Index =</i>	Index into MSsglist[] .
<i>Dump =</i>	The MSsglist[] entry.

Table 5 Error Code: PICB

Error Code:	PICB
Error Description:	There is an inconsistency between the array PClusgname[] associated with the line unit under audit and the relation RLPICB. The array may not be containing the correct mask.
Possible Error Effect:	The array has wrong information. The circuit names for this line unit could be miscalculated. This could cause serious problems for growth, maintenance and operational software.
Corrective Action Taken:	The audit will report the error and will schedule the PICB audit. The PICB audit will correct the array entry using the PICB number in static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the line unit of the array.
<i>Bad Data =</i>	The line unit under audit.
<i>Good Data =</i>	0. There is no good data.
<i>Logical Index =</i>	The value of the key field of the array.
<i>Dump =</i>	NA.

Table 6 Error Code: SCNFLG

Error Code:	SCNFLG
Error Description:	The scanflag is turned off. After switching on the line unit type, the key of the rIMS_SGSDY tuple is set to the type of line unit. In case of default, a check is made to see if the scanflag is turned off.
Possible Error Effect:	If the scanflag is turned off, we will be unable to recognize which of the switch groups from the array MSsglist[] should be scanned and/or allowed for collection.
Corrective Action Taken:	The audit will report the error and will turn the scanflag on.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the line unit of the array.
<i>Bad Data</i> =	The scanflag of the line unit under audit.
<i>Good Data</i> =	0. There is no good data.
<i>Logical Index</i> =	The value of the key field of the array.
<i>Dump</i> =	NA.

Table 7 Error Code: WRONGUSE

Error Code:	WRONGUSE
Error Description:	The array MSsglist[] does not have the correct study group.
Possible Error Effect:	The array has wrong information. We will be obtaining a study group for which the traffic usage count is not needed.
Corrective Action Taken:	The audit will report the error and will correct the study group field of the array from the static tuple rIMS_SGSDY.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the line unit of the array.
<i>Bad Data</i> =	The wrong study group of the array.
<i>Good Data</i> =	The study group from the static tuple rIMS_SGSDY
<i>Logical Index</i> =	The value of the key field of the array.
<i>Dump</i> =	NA.

MTIB

Software Release: 5E10 and later

1. METALLIC TEST INTERCONNECT BUS (MTIB) AUDIT

The metallic service units (MSUs) in the switching modules consist of two service groups that performs GDX compensation, metallic access functions, automatic line insulation testing, scan distribution, and power conversion. Each switching module may have two MSUs. The MSUs evaluate the quality of subscriber lines and trunks, and form an access network that interconnects analog facilities and frequency test equipment. By means of the metallic test interconnect buses, customers can connect test ports through the MSU to any line in an office. The test buses interconnect MSU service groups or, for modular MSUs, shelves within a service group. For multiple MSUs, there is one shelf for each service group and one access pack per shelf. For modular MSUs, there are up to four shelves per service group; as many as four shelves may share an access pack. Each access pack has 14 access points. Each access point connects to a metallic interconnect test bus. The metallic test interconnect buses are a global resource allocated in the communications module processor (CMP) and there are 14 buses per office.

The allocation and deallocation of metallic test interconnect buses is done by means of messages passed between the CMP and SMs. If the message is lost, the test buses become lost resources. The metallic test interconnect bus audit (MTIB) minimizes the consequences of a lost message and ensures that the test buses are not lost. The MTIB audit is an interprocessor audit that ensures the integrity of the metallic path data block relation (RLMPHDATA) by verifying the connections of the metallic test interconnect buses in the CMP and SMs. Note that the allocation strategy for metallic test interconnect buses differs for RSMs. Each RSM has its own set of 14 metallic test interconnect buses and they are allocated in the RSM; the circuit data audit (CKTDATA) audit ensures their consistency.

The MTIB audit is part of the peripheral control audits subgroup. It runs in the CMP and SM in the segmented modes.

2. DATA STRUCTURES

The primary data structure this audit checks is the RLMPHDATA relation in the CMP. This relation contains connection fields for the switching modules and process identification fields, all of which the audit checks. Before accessing RLMPHDATA tuple, the audit reads the static metallic unit metallic test interconnect bus relation (RLMUMTIB) to determine whether an switching module has an MSU. The audit also reads the central processor equipment status table relation (RLCPEST) to obtain the maintenance status of the test buses.

In the switching module, the audit uses the peripheral control MSU service group relation (RLPC_MSUSG) to step through all the service groups in the SMs. As it steps through RLPC_MSUSG, the audit calls the semantic check function for each service group. The audit checks the maintenance status of the metallic test interconnect buses recorded in the switch maintenance equipment status relation (RLSMEST). To find the owners of the buses, the MTIB audit checks the circuit data block common area relation (RLCDBCOM), which contains the software representation of the access points, and the process control block linkage area relation (RLPCBLA). Figure 1 shows these data structures.

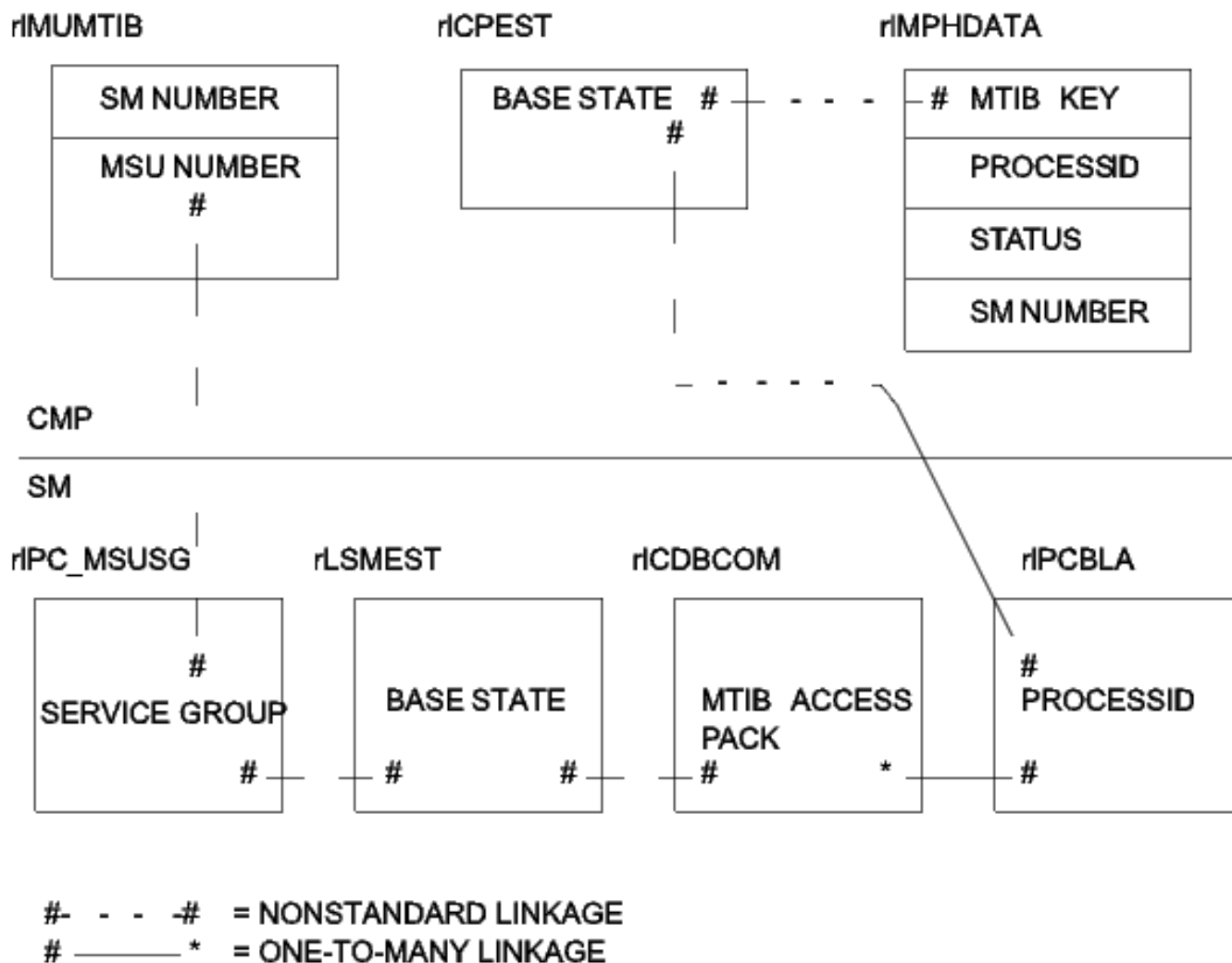


Figure 1 (MTIB-1) Data Structures of the MTIB Audit

3. AUDIT CHECKS

The CMP provides the master control of the audit. The master data resides in the switching modules. In the CMP the following audit checks are performed. If the audit has just been entered:

ZAP = Check if the audit has been entered due to a system interrupt.

Otherwise after the audit sends a message to all SMs from the CMP to collect MTIB status and receiving the reply:

BUSYIDLE = Check if the module connect one (mod_1conn) and the module connect two (mod_2conn) are idle.

MOD_1CONN = Check if the module connect one (mod_1conn) field is set correctly compared to data collected from the SMs.

MOD_2CONN = Check if the module connect two (mod_2conn) field is set correctly compared to data collected from the SMs.

NOTIDLE = Check if an SM indicates there is a connection to this MTIB but the module connect one

(mod_1conn) and the module connect two (mod_2conn) are idle.

- UNEQUAL = Check if the module connect two (mod_1conn) and the module connect two (mod_2conn) fields are set correctly compared to data collected from the SMs.
- BDPID = Check if the process identification one (ospid1) or the process identification two (ospid2) fields are set correctly compared to data collected from the SMs.
- COUNTS = Check if an SM indicates it has more than two connections to this MTIB.

In the SM the following audit checks are performed.

If the audit has just been entered:

- Call the DRSTATIC audit driver to step through the RLPC_MSUSG relation.

Otherwise collect MTIB status:

- ODDLOSTTUP = Check if the rICKTDATA tuple is missing for this MTIB.
- CKTDSMEST = Check if the rISMEST tuple is missing for this MTIB.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	The process identification one field (ospid1) and the process identification two field (ospid2) in the rIMPHDATA tuple has found to be different from the process identification data received from the SMs. For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rISMEST tuple and the operational status using the rICDBCOM tuple. If the rISMEST tuple is in-service (bas_stat = SMACT) and the rICDBCOM tuple is busy (cdccm.st_ownk != DBIDL MISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rICDBCOM or rISMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMPHDATA tuple found in error.
<i>Bad Data</i> =	The SM number believed to have the MTIB connection.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (mtibid) of the rIMPHDATA tuple found in error.
<i>Dump 1</i> =	rIMPHDATA tuple found in error.
<i>Dump 2</i> =	The number of MTIB connections and the two process identification fields

	received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 2 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The data collected from the SM indicates that there are no connections to this MTIB, but the module connect one (mod_1conn) or the module connect two (mod_2conn) fields in the rIMPHDATA tuple in the CMP indicates that this SM is connected to the MTIB. For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple and the operational status using the rCDBCOM tuple. If the rSMEST tuple is in-service (bas_stat = SMACT) and the rCDBCOM tuple is busy (cdccm.st_ownk != DBIDLISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rCDBCOM or rSMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The SM number believed to have the MTIB connection.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple found in error.
<i>Dump 1 =</i>	rIMPHDATA tuple found in error.
<i>Dump 2 =</i>	The number of MTIB connections and the two process identification fields received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 3 Error Code: CKTDSMEST

Error Code:	CKTDSMEST
Error Description:	Information in the static rMUMTIB tuple in the CMP and the static rIPC_MSUSG and rCKTDATA tuples in the SM indicates that a dynamic rSMEST tuple should exist for a particular conference circuit but it does not.
Possible Error Effect:	A rSMEST tuple is lost to the system and could cause asserts from maintenance application code, the inability to remove and restore the MTIB, and the inability to allocate the MTIB to perform metallic testing.
Corrective Action Taken:	The CKTDATA audit will be scheduled and should insert the rSMEST tuple back into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple buffer.
<i>Bad Data =</i>	Logical key (unit_id) of the rSMEST tuple found missing.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 4 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The data collected from the SM indicates that the SM has more than the two allowable connections to this rIMPHDATA tuple (or MTIB). For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple and the operational status using the rCDBCOM tuple. If the rSMEST tuple is in-service (bas_stat = SMACT) and

	the rICDBCOM tuple is busy (cdccm.st_ownk != DBIDLMISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rICDBCOM or rSMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The number of MTIB connections the SM believes it has.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple in error.
<i>Dump 1 =</i>	rIMPHDATA tuple found in error.
<i>Dump 2 =</i>	The number of MTIB connections and the two process identification fields received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 5 Error Code: MOD_1CONN

Error Code:	MOD_1CONN
Error Description:	The data collected from the SM indicates that this SM has one connection to this MTIB. The module connect one (mod_1conn) in the rIMPHDATA tuple indicates that this SM is using this MTIB, however, the process identification one (ospid1) field in the rIMPHDATA has found to be different from the process identification of the MTIB in the SM. For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple and the operational status using the rICDBCOM tuple. If the rSMEST tuple is in-service (bas_stat = SMACT) and the rICDBCOM tuple is busy (cdccm.st_ownk != DBIDLMISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rICDBCOM or rSMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The SM number believed to have the MTIB connection.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple found in error.
<i>Dump 1 =</i>	rIMPHDATA tuple found in error.
<i>Dump 2 =</i>	The number of MTIB connections and the two process identification fields received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 6 Error Code: MOD_2CONN

Error Code:	MOD_2CONN
Error Description:	The data collected from the SM indicates that this SM has one connection to this MTIB. The module connect two (mod_2conn) in the rIMPHDATA tuple indicates that this SM is using this MTIB, however, the process identification two (ospid2) field in the rIMPHDATA has found to be different from the process identification of the MTIB in the SM. For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved

	using data in the rSMEST tuple and the operational status using the rCDBCOM tuple. If the rSMEST tuple is in-service (bas_stat = SMACT) and the rCDBCOM tuple is busy (cdccm.st_ownk != DBIDLMISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rCDBCOM or rSMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The SM number believed to have the MTIB connection.
<i>Good Data =</i>	None.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple found in error.
<i>Dump 1 =</i>	rIMPHDATA tuple found in error.
<i>Dump 2 =</i>	The number of MTIB connections and the two process identification fields received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 7 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The data collected from the SM indicates that there is one or two connections to this MTIB, but the module connect one (mod_1conn) and the module connect two (mod_2conn) fields in the rIMPHDATA tuple in the CMP indicates that this SM is not connected to the MTIB. For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple and the operational status using the rCDBCOM tuple. If the rSMEST tuple is in-service (bas_stat = SMACT) and the rCDBCOM tuple is busy (cdccm.st_ownk != DBIDLMISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rCDBCOM or rSMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIMPHDATA tuple found in error.
<i>Bad Data =</i>	The SM number believed to have the MTIB connection.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key (mtibid) of the rIMPHDATA tuple found in error.
<i>Dump 1 =</i>	rIMPHDATA tuple found in error.
<i>Dump 2 =</i>	The number of MTIB connections and the two process identification fields received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 8 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Information in the static rIMUMTIB tuple in the CMP and the static rIPC_MSUSG tuple in the SM indicates that there is an MTIB circuit on this SM, but the associated static rCKTDATA tuple is missing in the SM.
Possible Error Effect:	A rCKTDATA tuple and possible the rSMEST tuple are lost to the system and could cause asserts from maintenance application code, the inability to remove and restore the MTIB, and the inability to allocate the MTIB to perform metallic testing.
Corrective Action Taken:	None.

Manual Action Required:	If there is a MSU with MTIBs in this SM then the rICKTDATA tuple should be inserted into the SM data base using ODBE. Otherwise, the RLPC_MSUSG relation in the SM should be updated using ODBE and the RLMUMTIB relation in the CMP should be updated using ODBE.
Dump Description:	
<i>Error Address</i> =	Address of the rICKTDATA tuple buffer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key (name) of the rICKTDATA tuple found missing.
<i>Dump</i> =	The rIPC_MSUSG tuple associated with this MTIB circuit.

Table 9 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The data collected from the SM indicates that this SM has two connections to this MTIB. The module connect one (mod_1conn) and/or the module connect two (mod_2conn) fields in the rIMPHDATA tuple indicate that this SM is not using this MTIB. For each SM containing an MSU, a message is sent from the CMP to the SM to collect data for all 14 MTIBs. In the SM the maintenance status is retrieved using data in the rSMEST tuple and the operational status using the rCDBCOM tuple. If the rSMEST tuple is in-service (bas_stat = SMACT) and the rCDBCOM tuple is busy (cdccm.st_ownk != DBIDL MISC), the owning process of this MTIB is send to the CMP and compared to data stored in the rIMPHDATA tuple.
Possible Error Effect:	The MTIB associated with this rIMPHDATA tuple may not be able to be allocated by terminal maintenance to perform metallic testing.
Corrective Action Taken:	The audit will reinitialize (idle) the rIMPHDATA tuple and send a message to the SMs to recover the rCDBCOM or rSMEST tuples associated with the rIMPHDATA tuple (or MTIB).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIMPHDATA tuple in error.
<i>Bad Data</i> =	The SM number believed to have the MTIB connection.
<i>Good Data</i> =	None.
<i>Logical Index</i> =	Logical key (mtibid) of the rIMPHDATA tuple found in error.
<i>Dump 1</i> =	rIMPHDATA tuple found in error.
<i>Dump 2</i> =	The number of MTIB connections and the two process identification fields received from the SM for the MTIB associated with this rIMPHDATA tuple.

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to perform recovery after taking a system interrupt (such as, write protect, out-of-range read, and so forth).
Possible Error Effect:	May cause further resets, audits, and/or asserts.
Corrective Action Taken:	None, all data that would have caused the reset is reinitialized on the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The SM number believed to have be connected to the MTIB (rIMPHDATA tuple) under audit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

MUPHST

Software Release: 5E10 and later

1. MULTIPLE POSITION HUNT STATUS (MUPHST) AUDIT

Information about ports, including their multi-line hunt groups (MLHGs), is stored in the office-dependent data (ODD) of each office. The information is kept on several relations, including the line status relation (RLLNSTAT), the large line status relation (RLLGLNSTAT), and the multiple position hunt status relation (RLRTMUPHST). The RLRTMUPHST relation contains information about multiple hunt groups connecting to the local switch through attendant consoles. Each console can restrict the amount of traffic it receives. The multiple position hunt status audit (MUPHST) ensures the integrity of the information in RLRTMUPHST.

The rRLTMUPHST relation may reside in any type of switching module (SM), and the relation contains tuples for all MLHG ports on this SM and other SMs. These port statuses, in the rRLTMUPHST tuple on the designated MLHG hunting SM, are validated with a static RLRT_LGMEM relation. No RLRTMUPHST relation exists on a member SM or a SM where global MLHG ports reside, if the SM has communication with the administrative module (AM). However, during the standalone operation, the RLRTMUPHST and RLLGMEM relations are created on the member SM with global MLHG ports.

The audit is part of the routing and terminal allocation. It runs in the operating system for distributed switching-switching module (OSDS-M) environment in the segmented mode.

2. DATA STRUCTURES

The MUPHST audit compares information in the dynamic RLRTMUPHST relation to information in the static multi-line hunt group relation (RLRT_MHG). The audit checks the key of each relation (group number field) to perform this check. Each rRLTMUPHST tuples is checked against either the static large MLHG member status relation (RLRT_LGMEM) in the hunting SM, or the dynamic large MLHG member status relation (RLLGMEM) in the member SM. The data structures of the MUPHST audit are shown in Figure 1 .

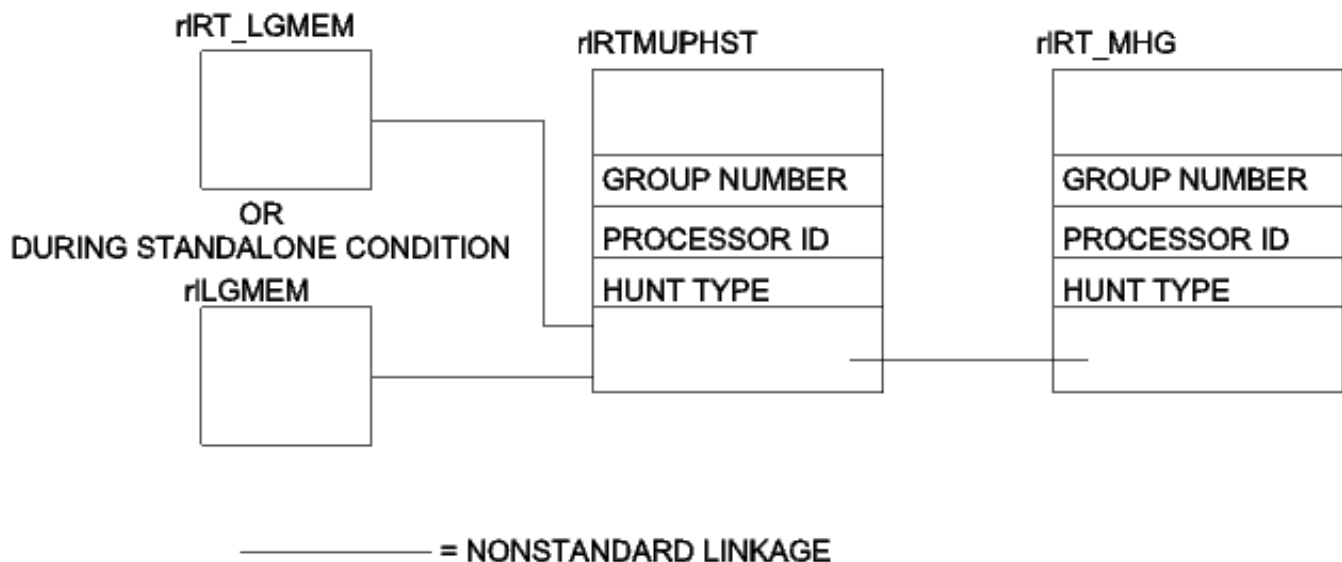


Figure 1 (MUPHST-1) Data Structures of the MUPHST Audit

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) to perform a dynamic-to-static existence check. The audit checks the key of each tuple in RLRTMUPHST, and tries to find an associated tuple in RLRT_MHG. If no corresponding tuple is

found in RLRT_MHG, the audit reports an error.

The audit also ensures that tuples exist in the correct processor by checking the processor ID field. In the SM, and in the RSM, the RLRTMUPHST relation contains tuples for all local hunt group ports that exist in that SM or RSM. In the SM, designated as hunting SM, the RLRTMUPHST relation contains tuples for all global and local hunt group ports belonging to that hunting SM. During stand-alone operation, global hunt groups are allowed in an RSM.

4. ERROR HANDLING

The audit deletes tuples from RLRTMUPHST if it cannot find a corresponding tuple in RLRT_MHG.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	While auditing an rLGLNSTAT tuple, a DAGETMEM did not find the associate rLGMEM tuple. Since during stand alone RLMEM relation is audited against static RLGROUP_PORT relation, audit should believe dynamic rLGMEM tuple and delete dynamic rLGLNSTAT tuple.
Possible Error Effect:	Incorrect use of data resources for MLHGs.
Corrective Action Taken:	The extra rRTMUPHST tuple deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLGMEM tuple.
<i>Bad Data =</i>	The key of the extra rMUPHST tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The busy/idle status in a tuple of the relation RLMUPHST is inconsistent with the corresponding member equipage tuple bit map. The nonequipped member should be marked busy. If the KEY field of the dump is 1 (SM stand alone) then an rRTMUPHST tuple is inconsistent with an rLGMEM tuple, or if it is 0 then the rLGLNSTAT tuple is inconsistent with an rRT_LGMEM tuple.
Possible Error Effect:	Failure in terminating to an invalidly marked idle members.
Corrective Action Taken:	The audit set the nonequipped members to correct status, busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rRTMUPHST tuple having the wrong busy/idle status.
<i>Bad Data =</i>	The array containing the members busy/idle status.
<i>Good Data =</i>	The array containing the members equipage information.
<i>Logical Index =</i>	This is not a key. Verification of busy/idle status is with rLGMEM tuple if this dump is 1, or else with rRT_LGMEM tuple if this dump is 0.
<i>Dump =</i>	NA.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	A future recent change attempting to add a MLHG of MUPH hunt type would fail due to this tuple already being present.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	rRTMUPHST tuple.

Table 4 Error Code: MUPHKEY

Error Code:	MUPHKEY
Error Description:	The tuple of the relation RLRTMUPHST has wrong key.
Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The correct value of the key will be inserted in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rRTMUPHST tuple having the wrong key.
<i>Bad Data</i> =	The value of the wrong key.
<i>Good Data</i> =	The correct value of the key.
<i>Logical Index</i> =	This is not a key. Verification of busy/idle status is with rLGMEM tuple if this field is 1, or else with rIRT_LGMEM tuple if this field is 0.
<i>Dump</i> =	The contents of the rIRT_MHG tuple.

Table 5 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	The audit has found an inconsistency between the static relations RLRT_LGMEM and RLRT_MHG. Current SM is the hunting SM of the MLHG as indicated by the field in rIRT_MHG tuple and therefore the key of the tuple is used to access the MLHG member equipage tuple in RLRT_LGMEM relation but no matching tuple is found and the audit cannot continue to do checks.
Possible Error Effect:	Audit continue to the next tuple of RLMUPHST relation.
Corrective Action Taken:	Audit report an error.
Manual Action Required:	Verify tuples rIRT_MHG and rIRT_LGMEM in the ODD, using the rIRT_MHG tuple key found in the "KEY" field of the audit report. Corresponding tuple rIRT_LGMEM with the same key should exist on the same SM. Correct any inconsistencies found.
Dump Description:	
<i>Error Address</i> =	The address of the rIRT_MHG tuple.
<i>Bad Data</i> =	The key of rIRT_LGMEM tuple returned from data base.
<i>Good Data</i> =	The hunt module of MLHG.
<i>Logical Index</i> =	The key of rIRT_MHG tuple used to access rIRT_LGMEM tuple.
<i>Dump</i> =	The contents of the rIRT_MHG tuple.

Table 6 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the SM and in the RSM, the RLRTMUPHST relation contains tuples for all local hunt group ports that exist in that SM or RSM. In the SM designated as hunting SM, the RLRTMUPHST relation contains tuples for all global and local hunt group ports belonging to that hunting SM. During stand-alone operation, global hunt groups are allowed in an RSM.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Processor which should contain the tuple.
<i>Logical Index</i> =	grpnum.
<i>Dump</i> =	rRTMUPHST tuple.

Table 7 Error Code: RELQKEY

Error Code:	RELQKEY
Error Description:	The tuple of the relation RLLGMEM has wrong key.

Possible Error Effect:	Use of this tuple might propagate errors.
Corrective Action Taken:	The related audit LGMEM will run and correct the key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rLGMEM tuple having the wrong key.
<i>Bad Data =</i>	The value of the wrong key.
<i>Good Data =</i>	The correct value of the key.
<i>Logical Index =</i>	This is not a key but the hunt module of the MLHG.
<i>Dump =</i>	NA.

Table 8 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in RLRTMUPHST
Possible Error Effect:	A future recent change attempt to change this group to MUPH hunt type would fail due to this tuple already being present.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Hunt type of the group.
<i>Logical Index =</i>	grpnum.
<i>Dump =</i>	rRTMUPHST tuple.

NAILUP-A

Software Release: 5E10 only

1. NAILUP (NAILUP) AUDIT

Nail-up connections are software connections that give the effect of direct, hard-wired connections from one trunk port to another. They allow the switch to interconnect with nonswitched special service (NSWSS) trunks and they eliminate the need to segregate externally switched and nonswitched trunks for integrated digital (T-carrier) interfaces. The nail-up connections are made using recent change transactions by operating company personnel. The connections may be made to digital facility interfaces (DFIs), digital equipment numbers (DENs), and trunk equipment numbers (TENs). The connections are continually monitored and data about them may be obtained using nail-up input messages. The NAILUP audit ensures that the database accurately reflects the nail-up connections.

The NAILUP audit is part of the routing and terminal allocation audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The data structures checked by the NAILUP audit are the static nailup relation (RLNAILUP), the dynamic port linkage area relation (RLPORTLA), and the static port type relation (RLFC_PORTTYP). Refer to Figure 1.

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) to step through the RLNAILUP relations. For each tuple, the audit calls the semantic check function. The semantic check verifies the existence of corresponding tuples in RLPORTLA.

- (a) If a corresponding tuple exists in RLPORTLA, the audit tests it for a nailup marking. If the tuple has a nailup marking, the audit proceeds to the next tuple. If the tuple does not have a nailup marking, the audit marks it and reports an error.
- (b) If a tuple in RLNAILUP does not have a corresponding tuple in RLPORTLA, the audit checks RLFC_PORTTYP for a corresponding tuple. If a corresponding tuple exists in RLFC_PORTTYP, a tuple is added to RLPORTLA and an error is reported. If a corresponding tuple does not exist in RLFC_PORTTYP, the error is reported with a request for manual action to insert the needed tuple in RLFC_PORTTYP.

4. ERROR HANDLING

Missing tuples or incorrect port markings will generate an audit error report.

No related audits are scheduled.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP(1)

Error Code:	LOSTTUP(1)
Error Description:	The corresponding rIORTLA tuple is missing.
Possible Error Effect:	The nail-up monitor can detect this error, but at a later time.
Corrective Action Taken:	If rIORTLA tuple is missing, it will be inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Port key.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	None.

Table 2 Error Code: NOTNU(1)

Error Code:	NOTNU(1)
Error Description:	This matching rNAILUP tuple is not marked as nail-up.
Possible Error Effect:	The nail-up monitor will detect this error, but at a later time.
Corrective Action Taken:	The nail-up bit is set in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	Contents of bad nailup field.
<i>Good Data</i> =	NAILUP bit should be set.
<i>Logical Index</i> =	Contents of port(key) field.
<i>Dump</i> =	None.

Table 3 Error Code: ODDLOSTTUP(1)

Error Code:	ODDLOSTTUP(1)
Error Description:	The corresponding rIPORTTYP tuple is missing.
Possible Error Effect:	This fault requires a manual action to fix. RC must be used to repair the static rNAILUP and static rIPORTTYP mismatch.
Corrective Action Taken:	None.
Manual Action Required:	Use RC to fix the mismatch of static relations.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTTYP tuple.
<i>Bad Data</i> =	Port key.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	None.

NAILUP-B

Software Release: 5E11 and later

1. NAILUP (NAILUP) AUDIT

Nail-up connections are software connections that give the effect of direct, hard-wired connections from one trunk port to another. They allow the switch to interconnect with nonswitched special service (NSWSS) trunks and they eliminate the need to segregate externally switched and nonswitched trunks for integrated digital (T-carrier) interfaces. The nail-up connections are made using recent change transactions by operating company personnel. The connections may be made to digital facility interfaces (DFIs), digital equipment numbers (DENs), and trunk equipment numbers (TENs). The connections are continually monitored and data about them may be obtained using nail-up input messages. The NAILUP audit ensures that the data base accurately reflects the nail-up connections.

The NAILUP audit is part of the Routing and Terminal Allocation audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The data structures checked by the NAILUP audit are the static nailup relation (rNAILUP), the dynamic port linkage area relation (RLPORTLA), and the static port type relation (RLFC_PORTTYP). Refer to Figure 1.

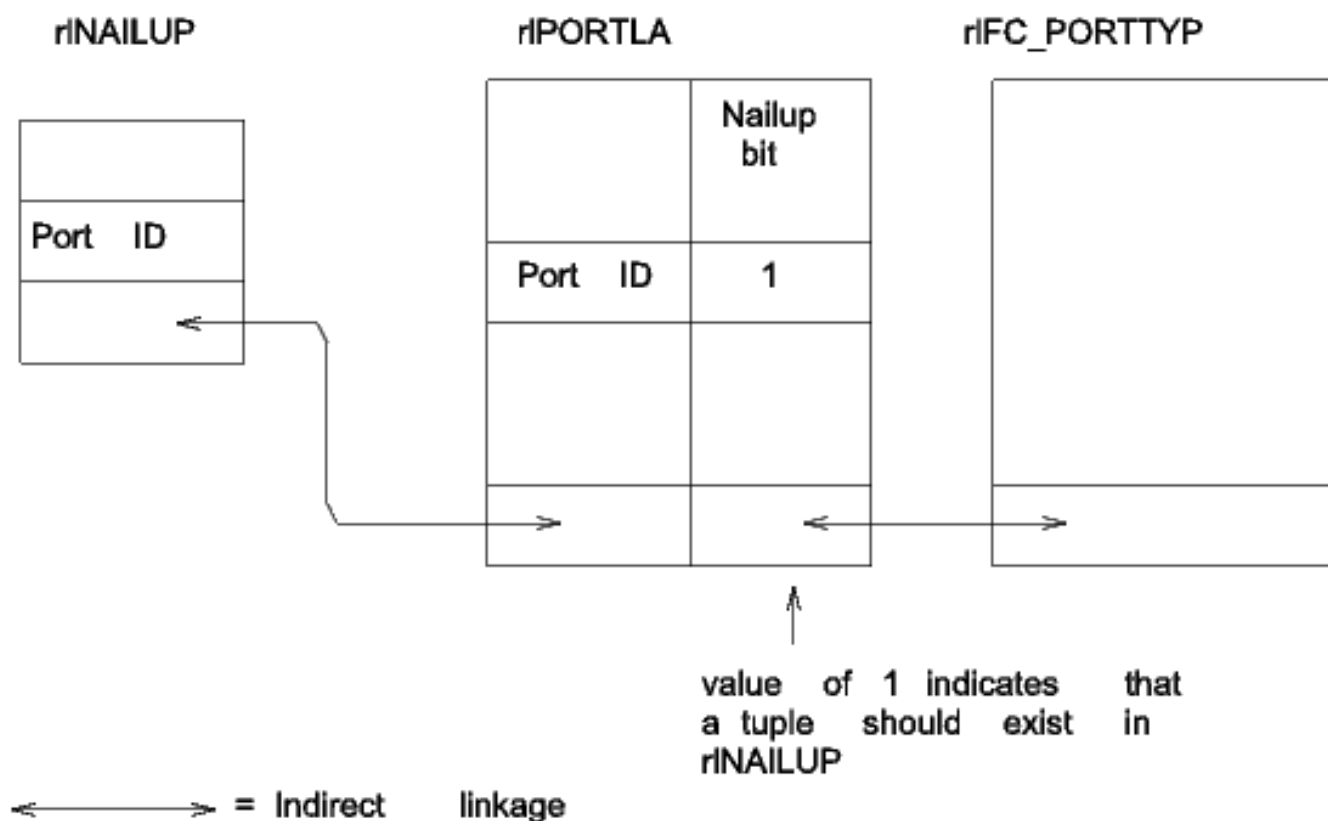


Figure 1 (NAILUP-1) Data Structures of the NAILUP Audit

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) to step through the rNAILUP relations. For each tuple, the audit calls the semantic check function. The semantic check verifies the existence of corresponding tuples in RLPORTLA.

- (a) If a corresponding tuple exists in rIORTLA, the audit tests it for a nailup marking. If the tuple has a nailup marking, the audit proceeds to the next tuple. If the tuple does not have a nailup marking, the audit marks it and reports an error.
- (b) If a tuple in RLNAILUP does not have a corresponding tuple in RLPORTLA, the audit checks RLFC_PORTTYP for a corresponding tuple. If a corresponding tuples exists in RLFC_PORTTYP, a tuple is added to RLPORTLA and an error is reported. If a corresponding tuple does not exist in RLFC_PORTTYP, the error is reported with a request for manual action to insert the needed tuple in RLFC_PORTTYP.

4. ERROR HANDLING

Missing tuples or incorrect port markings will generate an audit error report.

No related audits are scheduled.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP(1)

Error Code:	LOSTTUP(1)
Error Description:	The corresponding rIORTLA tuple is missing.
Possible Error Effect:	The nail-up monitor can detect this error, but at a later time.
Corrective Action Taken:	If rIORTLA tuple is missing, it will be inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	rIFC_PORTTYP.servlass.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port key.
<i>Dump 1 =</i>	rINAILUP tuple.
<i>Dump 2 =</i>	rIFC_PORTTYP tuple.

Table 2 Error Code: NOTNU(1)

Error Code:	NOTNU(1)
Error Description:	This matching rINAILUP tuple is not marked as nail-up.
Possible Error Effect:	The nail-up monitor will detect this error, but at a later time.
Corrective Action Taken:	The nail-up bit is set in the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Contents of bad nailup field.
<i>Good Data =</i>	NAILUP bit should be set.
<i>Logical Index =</i>	Contents of port (key) field.
<i>Dump =</i>	None.

Table 3 Error Code: ODDLOSTTUP(1)

Error Code:	ODDLOSTTUP(1)
Error Description:	The corresponding rIFC_PORTTYP tuple is missing.
Possible Error Effect:	This fault requires a manual action to fix. RC must be used to repair the static rINAILUP and static rIFC_PORTTYP mismatch.
Corrective Action Taken:	None.
Manual Action Required:	Use RC to fix the mismatch of static relations.
Dump Description:	
<i>Error Address =</i>	Address of rIFC_PORTTYP tuple.
<i>Bad Data =</i>	Port key in rIFC_PORTTYP.
<i>Good Data =</i>	Port key in rINAILUP.
<i>Logical Index =</i>	NA.

<i>Dump</i> = riNAILUP tuple.

NIDATA

Software Release: 5E10 and later

1. INTERNAL DATA (NIDATA) AUDIT

The internal data audits compare the tables contained in main memory with tables maintained on disk and report any mismatches. The tables are also checked for invalid data within a table and inconsistencies across tables. The audit reports all errors that are found using the appropriate AUD:NIDATA output message (OM). The error code contained in the OM can be used to find the action to be taken to correct the problem in the tables contained in this section with each audit description. If any errors are uncovered which cannot be corrected using recent change, then the audit will correct the error as necessary and report the correction.

The internal data audits are manual audits only. To run these audits, the user must manually invoke the audit by either entering the following input message:

AUD:NIDATA x

where x represents the specific audit to be run (1-5, 8, 10).

It is suggested that these audits be run weekly to ensure the reliability of the CNI database. It is also suggested that these audits be run after completion of recent change activity to audit the success of the session. It is not necessary to run these audits after each recent change order, but once the user has completed the recent change activity for a session, it is then suggested that these audits be activated.

2. ERROR HANDLING

Any errors found by these audits should be referred to the administrator of the database.

ERROR CODE	EXPLANATION OF ERROR	ACTION TO BE TAKEN
100	Unable to open the office data disk file.	This error is corrected by the audit by creating a new disk file and populating it with the data in the AM. It is suggested that the audit be rerun to verify that this condition no longer exists.
101	Read error encountered with the office data disk file.	This error is corrected by the audit by creating a new disk file and populating it with the data in the AM. It is suggested that the audit be rerun to verify that this condition no longer exists.
102	The local CLLI code on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using an office data recent change.
103	Invalid local CLLI code.	This error indicates that one of the data items is invalid. This can be corrected by doing a recent change of the office data and modifying the data item in question.
104	Links cannot use the local CLLI code.	This is a discrepancy between the office data and the link data. If the link data is in error, then doing link data recent changes on the appropriate links should clear up the problems. If however, the office data is wrong, then a recent change of the office data can be done to modify the item(s) in error.
105	The region number on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using an office data recent change.
106	Invalid region number.	This error indicates that one of the data items is invalid. This can be corrected by doing a recent change of the office data and

		modifying the data item in question.
107	The function number on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using an office data recent change.
108	Invalid function number.	This error indicates that one of the data items is invalid. This can be corrected by doing a recent change of the office data and modifying the data item in question.
109	Links cannot use the local function number as their far end function number.	This is a discrepancy between the office data and the link data. If the link data is in error, then doing link data recent changes on the appropriate links should clear up the problems. If however, the office data is wrong, then a recent change of the office data can be done to modify the item(s) in error.
110	The point code on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using an office data recent change.
111	Invalid point code.	This error indicates that one of the data items is invalid. This can be corrected by doing a recent change of the office data and modifying the data item in question.
112	The region number contained in the point code does not match the local region number.	This error indicates that one of the data items is invalid. This can be corrected by doing a recent change of the office data and modifying the data item in question.
113	Links cannot use the local point code as their far end point code.	This is a discrepancy between the office data and the link data. If the link data is in error, then doing link data recent changes on the appropriate links should clear up the problems. If however, the office data is wrong, then a recent change of the office data can be done to modify the item(s) in error.
114	The link integrity status (LIS) destination on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using an office data recent change.
115	Invalid LIS destination.	This error indicates that one of the data items is invalid. This can be corrected by doing a recent change of the office data and modifying the data item in question.
116	The international point code on disk does not match the AM copy.	These errors indicate that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using an office data recent change.
117	Invalid international point code.	These errors all indicate that one of the data items is invalid. These can be corrected by doing a recent change of the office data and modifying the data item in question.
118	Links can not use the international local point code as their far-end point code.	These are discrepancies between the office data and the link data. If the link data is in error, then doing link data recent changes on the appropriate links should clear up the problems. If however, the office data is wrong, then a recent change of the office data can be done to modify the item(s) in error.
119	Cannot write or create the office data disk file.	According to this error message, the audit is having difficulty in creating or writing the new office data disk file. It is suggested that the audit be rerun and if the error message appears again, the user should consult with his system administrator to find out the cause of the error.
120	Invalid version number on disk file.	The version number on the disk does not match the version

		number the audit expects. Check whether the disk file was retrofitted or whether it is in the proper format.
121	Unable to open the link set to multiple point code disk file.	The audit is having difficulty in opening and/or reading the office or link set to multiple point code data disk file. This will be corrected during the audit by creating a new disk file and populating it with the data in the AM. Re-run the audit to verify that this condition no longer exists.
122	Read error encountered with the link set to multiple point code data disk.	The audit is having difficulty in opening and/or reading the office or link set to multiple point code data disk file. This will be corrected during the audit by creating a new disk file and populating it with the data in the AM. Re-run the audit to verify that this condition no longer exists.
123	The point code on disk does not match AM copy.	The data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If, however, the disk version is correct, then the data in the AM must be changed using an office data or link set multiple point code data recent change.
124	Invalid point code in the link set to multiple point code table.	One of the data items is invalid. These can be corrected by doing a recent change of the office data or link set to multiple point code data and modifying the data item in question.
125	Up to 8 different point codes are allowed in the link set to multiple point code table.	One of the data items is invalid. These can be corrected by doing a recent change of the office data or link set to multiple point code data and modifying the data item in question.
126	At least one of the point codes in the table should be equal to a local point code from the office data disk file.	One of the data items is invalid. These can be corrected by doing a recent change of the office data or link set to multiple point code data and modifying the data item in question.
127	The signal transfer point (STP) indicator flag is not 0.	The value 0 indicates that the office is not an STP. This can be corrected by performing a common network interface (CNI) level 3 initialization.
128	The ANS [®] standard point code flag on disk does not match the AM copy.	The ANS [®] standard point code flag is invalid. If the flag in the AM is correct, then doing a recent change that copies AM office data to disk will remedy the problem. If the flag in the AM is not correct, then use cnidboc's "U" option for office data to change the flag, cDstdpc, to the correct value. A value of 0 means that the office has the AT&T point code format. A value of 1 means that the office has the ANS [®] point code format. After correcting the AM flag, perform a recent change that copies AM office data to disk.
129	The ANS [®] standard point code flag is invalid.	If the flag in the AM is correct, then doing a recent change that copies AM office data to disk will remedy the problem. If the flag in the AM is not correct, then use cnidboc's "U" option for office data to change the flag, cDstdpc, to the correct value. A value of 0 means that the office has the AT&T point code format. A value of 1 means that the office has the ANS [®] point code format. After correcting the AM flag, perform a recent change that copies AM office data to disk.
130	The critical event throttling parameter "ce_num" on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If the disk version is correct, then the data in the AM must be changed using an office data recent change.
131	The critical event throttling parameter "ce_num" is out of range.	This error indicates that one of the data items is out of range. These can be corrected by doing a recent change of the office data and modifying the data item in question.
132	The critical event throttling parameter "ce_time" on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If the disk version

		is correct, then the data in the AM must be changed using an office data recent change.
133	The critical event throttling parameter "ce_time" is out of range.	This error indicates that one of the data items is out of range. These can be corrected by doing a recent change of the office data and modifying the data item in question.
134	The signaling route set test low priority timer "Srs_low_timer" on disk does not match the AM copy.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change that copies AM to disk will remedy the problem. If the disk version is correct, then the data in the AM must be changed using an office data recent change.
135	The signaling route set test low priority timer "Srs_low_timer" is out of range.	This error indicates that one of the data items is out of range. These can be corrected by doing a recent change of the office data and modifying the data item in question.
136	The signaling route set test low priority timer "Srs_low_timer" is not a multiple of 30.	This error indicates that one of the data items is out of range. These can be corrected by doing a recent change of the office data and modifying the data item in question.
200	Unable to open the link data disk file.	This error indicates problems in opening/reading the link data disk file. These will be corrected in the audit by creating a new file and populating it based on the data in the AM. It is suggested that the audit be re-run in order to insure that this condition no longer exists.
201	Read error encountered in the link data disk file.	This error indicates problems in opening/reading the link data disk file. These will be corrected in the audit by creating a new file and populating it based on the data in the AM. It is suggested that the audit be re-run in order to insure that this condition no longer exists.
202	Link data exists for an RPC.	This error indicates that a link data disk file exists for an RPC. This file should be removed.
203	Disk file exists but no data in the AM.	If the disk file is not needed, remove it. If however, this link is supposed to have data, then do a link data recent change and add the appropriate information (the old disk file will be over-written with the new data).
204	Invalid protocol.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
205	Invalid encryption field.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
206	Invalid link type.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
207	Invalid far-end signal transfer point (STP).	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
208	Inappropriate syu code.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
209	Inappropriate break-in.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
210	Invalid combined link set.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
211	Invalid mate link.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
212	Mate link is an RPC.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link

		data recent change and modifying the appropriate field.
213	Invalid mate relationship.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
214	Mate links are going to same far-end STP.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
215	Invalid CLLI code (first 6 chars or last 5 chars).	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
216	A/E/B/D-link has local/mate CLLI code.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
217	A/E/B/D-link has an invalid far-end function.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
218	A/E/B/D-link has local/mate function.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
219	Invalid major state.	
220	Available links can not be self-looped.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
221	Available CCS6 links must have signaling links (SLKs) assigned.	Available CCS6 links must have an SLK assigned if the logical-physical tables are to be used. The SLK can be assigned using a logical-physical add recent change.
222	Invalid speed (that is, does not match protocol).	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
223	Invalid vfl field.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
224	Invalid voice access test quadrant.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
225	B/D-link has an invalid far-end region.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
226	Invalid pool number.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
227	No pool but there is a combined pool.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
228	The pool-link table is not consistent with the link data.	
229	Invalid layer number.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
230	Invalid signaling link code.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
231	Invalid link set.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
232	No link set but there is a combined link set.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
233	A/E/B/D-link has an invalid far-end point	These error messages all indicate that there is some invalid data

	code.	items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
234	A/E/B/D-link has local/mate point code.	These error messages all indicate that there is some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
235	The link data in the AM does not match the copy on disk.	Error messages of this type reflect the fact that the data in the AM does not match the data on disk. If the data in the AM is deemed correct, then do a recent change that copies the AM to disk. If however, the data on disk is correct, then do a link data recent change and modify the appropriate fields.
236	Links going to the same destination need to have consistent data.	These errors are due to the fact that links going to the same far-end function have inconsistent data. The link(s) that are incorrect can be fixed using a link data recent change.
237	The derived data items are incorrect.	Certain items stored in the link data do not appear on the recent change (RC) form because these items can be derived based on the speed of the link. If these items should become inconsistent with the current speed of the link, then doing a link data recent change without really changing anything should cause the items to be derived correctly.
238	The linkset-link table is not consistent with the link data.	
239	The linkset relation table is not consistent with the link data.	While scanning through the link data, a linkset relation table will be generated. If the current linkset relation table does NOT match the table just generated, then the existing table will be corrected accordingly. Running the audit again will verify that the table was indeed updated correctly.
240	There are files in the link data directory which do not conform to the naming convention.	This error indicates that there are files in the link data directory which do not fit the 'lnXX-X' format. These files should be inspected and corrected accordingly.
241	The VL assignments as shown in the link data do not match the logical-physical tables.	These errors can be corrected by doing a recent change that copies the logical-physical tables to disk. When the logical-physical tables are updated on disk, the vl assignments in the link data will be updated at the same time.
242	Mate link should be a CCS link.	These error messages indicate that there are some invalid data items in the link data. These can be corrected by doing a link data recent change and modifying the appropriate field.
243	For SADC node, facility access type (FACIL) must be MODEM.	These error messages indicate that there are some invalid data items in the link data. These can be corrected by doing a link data recent change and modifying the appropriate field.
244	Invalid LI4 equipage.	These error messages indicate that there are some invalid data items in the link data. These can be corrected by doing a link data recent change and modifying the appropriate field.
245	Invalid Mate link field.	These error messages indicate that there are some invalid data items in the link data. These can be corrected by doing a link data recent change and modifying the appropriate field.
246	Invalid Link Speed.	These error messages indicate that there are some invalid data items in the link data. These can be corrected by doing a link data recent change and modifying the appropriate field.
247	Invalid Link Facility ID.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
248	Invalid circuit ID number (CIN).	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
249	Invalid User.	This error message indicates that there is an invalid data item in the link data. By doing link data recent change without really changing anything should cause the item to be derived correctly.
250	Invalid ring node address (RNA) or RNA	These error messages indicate that there are some invalid data

	out of range.	items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
251	For SADC node with MODEM facility, T1FA control node (TCN) field must be unassigned.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
253	Invalid mate link major field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
254	Mated mate DCHN link should not be in the same node.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
255	Inconsistent Prim/Sec field and mate link field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
256	Invalid logical link ID (LACID).	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
257	LACID provisioned for link does not match the LACID provisioned on the mate link.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
258	Mate link must be SECONDARY, since link is provisioned as PRIMARY.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
259	Mate link must be PRIMARY, since link is provisioned as SECONDARY.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
260	Invalid Primary Secondary field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
261	Invalid mate link RNA.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
262	Invalid mate link node type.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
263	Invalid facility field (must be T1FA or MODEM).	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
264	Invalid medium field (must be SATELLITE or TERRESTRIAL).	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
265	Invalid ec_mode field (error correction mode).	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
266	Invalid receive queue (rqlen) and/or transmit queue (xqlen) fields.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
267	Invalid congestion onset threshold field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
268	Invalid congestion abatement threshold field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
269	Invalid normal alignment error rate monitor (AERM) signal unit (SU) error rate threshold (tin_val) field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
270	Invalid emergency AERM SU error rate threshold (tie_val) field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.

271	Invalid SUERM SU error rate threshold (ts_val) field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
272	Invalid re-transmit queue (rxqlen) field.	These error messages indicate that there are some invalid data items in the link data. These can all be corrected by doing a link data recent change and modifying the appropriate field.
300	The user has not provided these tables to the CNI.	
301	Open/read error encountered with the lxp_tab disk file.	These errors indicate problems in opening/reading the disk files associated with the logical-physical translation tables. These will be corrected in the audit by creating a new disk file and populating it based on the data in the AM. It is suggested that the audit be rerun in order to insure that this condition no longer exists.
302	Open/read error encountered with the pxl_tab disk file.	These errors indicate problems in opening/reading the disk files associated with the logical-physical translation tables. These will be corrected in the audit by creating a new disk file and populating it based on the data in the AM. It is suggested that the audit be rerun in order to insure that this condition no longer exists.
303	Open/read error encountered with the vl_tab disk file.	These errors indicate problems in opening/reading the disk files associated with the logical-physical translation tables. These will be corrected in the audit by creating a new disk file and populating it based on the data in the AM. It is suggested that the audit be rerun in order to insure that this condition no longer exists.
304	Open/read error encountered with the ln7slots disk file.	These errors indicate problems in opening/reading the disk files associated with the logical-physical translation tables. These will be corrected in the audit by creating a new disk file and populating it based on the data in the AM. It is suggested that the audit be rerun in order to insure that this condition no longer exists.
305	The logical-physical data on disk does not match the AM copy.	
306	The physical-logical data on disk does not match the AM copy.	
307	All CCS6 links in the AVL major state must have an SLK assigned.	
308	A link has an SLK assigned but there is no link data for the link.	
309	Physical-logical table indicates that the link is CCS7 but the link data is CCS6.	
310	The index into the virtual link table is invalid.	
311	A link is using an entry in the virtual link table which is marked as unused.	
312	The virtual link data on disk does not match the AM copy.	
313	A link is using an invalid SLK.	
314	A ring node address/virtual link (RNA/VL) translates to an SLK but the reverse translation does not hold true.	
315	A CCS7 link has an entry in the virtual link table but it doesn't have any VL's assigned.	
316	Physical-logical table indicates that the link is CCS6 but the link data is CCS7.	
317	An SLK translates to an invalid RNA/VL.	
318	An SLK translates to a link which is unequipped (that is, no link data).	
319	An SLK translates to a CCS6 link but the virtual link is not 0.	
320	An SLK translates to a RNA/VL but the reverse translation does not hold true.	
321	An SLK translates to a CCS7 link but no	

	index into virtual link table.	
322	An SLK translates to a CCS7 link but the virtual link is invalid.	
323	The ln7slots data on disk does not match the AM copy.	
324	Invalid value in the ln7slots table.	
325	A virtual link entry is marked used but no link is using this entry.	
326	A virtual link entry is marked unused but there is a virtual link assigned.	
327	The virtual link assignments in the logical-physical tables are inconsistent with the link data.	
328	An SLK translates to a link which is equipped but is not a CCS link.	
400	Unable to open the ccs7_pc routing data disk file.	These errors are corrected during the audit. A new disk file is created and populated from the table in the AM. Rerunning the audit verifies that the condition no longer exists.
401	Unable to open the rtg_patterns data disk file.	These errors are corrected during the audit. A new disk file is created and populated from the table in the AM. Rerunning the audit verifies that the condition no longer exists.
402	Read error was encountered with the ccs7_pc routing data disk file.	These errors are corrected during the audit. A new disk file is created and populated from the table in the AM. Rerunning the audit verifies that the condition no longer exists.
403	Read error was encountered with the rtg_patterns data disk file.	These errors are corrected during the audit. A new disk file is created and populated from the table in the AM. Rerunning the audit verifies that the condition no longer exists.
404	Invalid version number was encountered on disk file.	The version number on the disk does not match the version number the audit expects. Check whether or not the file was retrofitted. If the file has been retrofitted, then verify the routing data in the AM is correct and do a disk update.
405	Data on disk is not in sequential order, NIDATA is attempting to sort the disk file.	This error is not corrected during the audit. The memory buffer that contains the disk data is sorted by point code as the first key and the routing flag as the second key, so that the audit can continue testing. After the audit is complete, do the recent change and update the disk.
406	Inconsistent prim, alt1, and alt2 routes were encountered between the rtg_patterns table on disk and the routing priority table in the administrative module (AM) memory.	Audit found a mismatch between the prim/alt1/alt2 routes in the rtg_patterns table on the disk and the routing priority table in the AM. Manually check all the ccs7 point code data in the AM. If any ccs7 point code has an incorrect prim/alt1/alt2, do a recent change to correct the routes for the point codes. Once all the ccs7 point codes in the AM are correct, do a disk update.
407	Missing prim, alt1, and alt2 routes were encountered from the rtg_patterns table on disk.	Audit found missing information during comparison of the disk and the AM. Do a recent change or disk update accordingly. If error 408 is found, use the AUD:NMDATA input message in order to have a complete correction.
408	Missing prim, alt1, and alt2 routes were encountered from the routing priority table in the AM.	Audit found missing information during comparison of the disk and the AM. Do a recent change or disk update accordingly. If error 408 is found, use the AUD:NMDATA input message in order to have a complete correction.
409	Invalid items on ccs7_pc disk file. Found duplicate point code and routing flag, but different priority routes and/or director index.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message.
410	Invalid items on ccs7_pc disk file. Duplicate point code and conflicting routing flag were found.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message.

411	Fatal error 409, 410, or 431 was found, NIDATA 4 is terminated.	Fatal error, the audit terminates. According to the error messages, do the recent change fix and rerun the audit.
412	Invalid prim/alt1/alt2 routes were found on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
413	Invalid director index was found on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
414	Point code has invalid cluster number on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
415	Point code has invalid member number on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
416	Prim/alt1/alt2 routes are not active on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
417	Local network has invalid director index on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
418	Point code with cluster or member flag has no network data on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different

		situations.
419	Point code with member flag has no cluster data on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message. NOTE: Errors 412 through 419 produce more error messages since the same error might be caught in different situations.
420	Unable to create a new file.	The audit is having difficulty in creating or writing a new file. It is suggested that the audit be rerun and if the error message appears again, consult with the system administrator to find out the cause of the error.
421	Mismatch between the routing flags on disk and in the AM.	During the comparison of the disk and the AM, the audit found the data in the AM memory does not match the data on the disk. Analyze which data is correct, do the recent change, and update the disk.
422	Mismatch between the prim/alt1/alt2 routes on disk and in the AM.	During the comparison of the disk and the AM, the audit found the data in the AM memory does not match the data on the disk. Analyze which data is correct, do the recent change, and update the disk.
423	Mismatch between the director index on disk and in the AM.	During the comparison of the disk and the AM, the audit found the data in the AM memory does not match the data on the disk. Analyze which data is correct, do the recent change, and update the disk.
424	Missing routing data on ccs7_pc routing data disk file.	Audit found missing information during comparison of the disk and the AM. Do a recent change or disk update accordingly. If error 408 is found, run the AUD:NMDATA input message in order to have a complete correction.
425	Missing routing data in the AM.	Verify the routing data on disk is correct, then recent change the missing point code and rerun the AUD:NIDATA input message.
426	Invalid routing flag on ccs7_pc routing data disk file.	These errors are found when AUD:NIDATA validates the routing information on disk. The data is invalid. Do the correction by doing recent change on the point code and update the disk. Then rerun the AUD:NIDATA input message.
428	Duplicate routing information on ccs7_pc routing data disk file.	This error is automatically corrected by deleting the duplicated point code in the routing data table.
429	Too many error messages, NIDATA 4 terminated.	Too many errors occurred during the audit. It is suggested that all reported errors be fixed and then rerun the AUD:NIDATA input message.
430	Unable to write a file.	The audit is having difficulty in creating or writing a new file. It is suggested that the audit be rerun and if the error message appears again, consult with the system administrator to find out the cause of the error.
431	Not enough room on disk for all the AM routing data while updating the disk from the AM.	These errors were found during the disk update from AM memory. Check the data in the AM, make the correction, and rerun the AUD:NIDATA input message.
432	Invalid routing flag found in the AM while in the process of updating the disk from the AM.	These errors were found during the disk update from AM memory. Check the data in the AM, make the correction, and rerun the AUD:NIDATA input message.
500	Unable to open the subsystem distribution disk file.	This error is corrected during the audit. A new disk file will be created and populated from the table in the AM. Rerunning the audit will verify that the condition no longer exists.
501	Read error encountered in the subsystem distribution disk file.	This error is corrected during the audit. A new disk file will be created and populated from the table in the AM. Rerunning the audit will verify that the condition no longer exists.
502	Unable to open the local subsystem information disk file.	This error is corrected during the audit. A new disk file will be created and populated from the table in the AM. Rerunning the

		audit will verify that the condition no longer exists.
503	Read error encountered in the local subsystem information disk file.	This error is corrected during the audit. A new disk file will be created and populated from the table in the AM. Rerunning the audit will verify that the condition no longer exists.
504	The data in the AM does not match the copy on disk.	Error messages of this type reflect the fact that the data in the AM does not match the data on disk. If the data in the AM is deemed correct, then do a recent change that copies the AM to disk. If however, the data on disk is correct, then do a subsystem information recent change and modify the appropriate fields.
505	The subsystem should be equipped.	Certain subsystems must be equipped by default. The "equipped indicator" does not appear on the RC form, but is handled automatically. This error is corrected by doing a "define subsystem information" recent change, followed by a recent change that copies AM to disk. The "equipped indicator" will be set automatically.
506	Invalid equipped indicator.	The "equipped indicator" does not appear on the RC form, but is handled automatically. This error is corrected by doing a "define subsystem information" or "delete subsystem information" recent change, followed by a recent change that copies AM to disk. The "equipped indicator" will automatically be set or cleared appropriately.
507	For an equipped subsystem: Invalid channel to receive signaling messages.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
508	For a subsystem which is equipped by default: Invalid local function index.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
509	For a subsystem which is not equipped by default: Invalid local function index.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
510	For an equipped subsystem: Invalid want pause indicator.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
511	For an equipped subsystem: Invalid channel to receive management indications.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
512	Invalid simplex/duplex indicator.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
513	For an unequipped subsystem: Invalid address to receive signaling messages.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
514	For an unequipped subsystem: Invalid	These error messages all indicate that there is some invalid data

	channel to receive signaling messages.	items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
515	For an unequipped subsystem: Invalid local function index.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
516	For an unequipped subsystem: Invalid want pause indicator.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
517	For an unequipped subsystem: Invalid simplex/duplex indicator.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
518	For an unequipped subsystem: Invalid channel to receive management indications.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
519	For an unequipped subsystem: Invalid mate subsystem number.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
520	For an unequipped subsystem: Invalid mate point code.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
521	For an unequipped subsystem: Invalid address to receive management indications.	These error messages all indicate that there is some invalid data items for an unequipped subsystem. These errors are corrected during the audit. The invalid data items are replaced by the proper default data items for an unequipped subsystem. Rerunning the audit will verify that the condition no longer exists.
522	For an equipped subsystem: Invalid address to receive signaling messages, or invalid address to receive management indications.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
523	For an equipped subsystem: The address to receive signaling messages or the address to receive management indications is an RPC.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
524	For an equipped subsystem: The address to receive signaling messages or the address to receive management indications should be an equipped CCS7 link.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
525	For a simplex subsystem: Should have a null mate subsystem number.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which

		modifies the appropriate field(s), followed by a recent change that copies AM to disk.
526	For a simplex subsystem: Should have a null mate point code.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
527	For a duplex subsystem: Invalid mate subsystem number.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
528	For a duplex subsystem: Invalid mate point code.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
529	For a subsystem which is not equipped by default: Invalid outgoing local function index.	These error messages all indicate that there is some invalid data items for an equipped subsystem. These can all be corrected by doing a "define subsystem information" recent change which modifies the appropriate field(s), followed by a recent change that copies AM to disk.
530	For an unequipped subsystem: Invalid outgoing local function index.	
531	Subsystem Manager has an invalid field.	This error message indicates that there is an invalid field for the subsystem manager. These errors are corrected during the audit. The invalid fields are replaced by the proper default values for subsystem manager and the table entry is copied from the AM to disk. Rerunning the audit will verify that the condition no longer exists.
800	Unable to open the global title translator data disk file.	According to these error messages, the audit is having difficulty in opening or reading the global title translator data disk file. This will be corrected during the audit by creating a new disk file and populating it with the data in the AM. It is suggested that the audit be rerun to verify that this condition no longer exists.
801	Read error encountered with the global title translator data disk file.	According to these error messages, the audit is having difficulty in opening or reading the global title translator data disk file. This will be corrected during the audit by creating a new disk file and populating it with the data in the AM. It is suggested that the audit be rerun to verify that this condition no longer exists.
802	Unable to create or write the global title translator data disk file.	According to this error message, the audit is having difficulty in creating or writing the new global title translator data disk file. It is suggested that the audit be rerun and if the error message appears again, the user should consult with his system administrator to find out the cause of the error.
803	The AM copy and disk do not match.	This error indicates that the data on disk does not match the data in the AM. If the data in the AM is correct, then doing a recent change which copies AM to disk will remedy the problem. If however, the disk version is correct, then the data in the AM must be changed using a global title translator data recent change.
804	The global title translator pair is zero.	These errors all indicate that one of the data items in the table is invalid. These can be corrected by doing a recent change of the global title translator data and modifying the data item in question.
805	Invalid network identifier.	These errors all indicate that one of the data items in the table is invalid. These can be corrected by doing a recent change of the global title translator data and modifying the data item in question.
806	Member should be zero for an stp point	These errors all indicate that one of the data items in the table is

	code.	invalid. These can be corrected by doing a recent change of the global title translator data and modifying the data item in question.
807	No routing data exists for the point code.	This error indicates that no routing data exists for the point code in the table. The user will have to enter routing data by doing a recent change.
1000	Unable to open the protocol timer/parameter disk file.	These errors are corrected during the audit. A new disk file will be created and populated from the data in the AM. Re-running the audit will verify that the condition no longer exists.
1001	Read error encountered in the protocol timer/parameter disk file.	These errors are corrected during the audit. A new disk file will be created and populated from the data in the AM. Re-running the audit will verify that the condition no longer exists.
1002	Unable to create/write protocol timer/parameter disk file.	This error indicates that the operating system prevented the audit from creating/writing a new disk file. The reason should be determined and corrected before re-running the audit.
1003	Office timer data in administrative module (AM) memory differs from that in the disk file.	Error messages of this type reflect the fact that the office timer data in the AM does not match the data on the disk. If the data in the AM is deemed correct, then do an office timer recent change that copies the AM memory office timers to disk. If the data on disk is correct, then do an office timer recent change for AM memory and modify the appropriate timer(s).
1004	Office timer data in AM memory is invalid.	These error messages indicate that there are invalid office timer data items in AM memory. These can be corrected by doing an office timer recent change for AM memory. Consider following this with an office timer recent change that copies the AM memory office timers to disk.
1005	Link set timer data in AM memory differs from that in the disk file.	Error messages of this type reflect the fact that the link set timer data in the AM does not match the data on the disk. If the data in the AM is deemed correct, then do a link set timer recent change that copies the AM memory link set timers to disk. If the data on disk is correct, then do a link set timer recent change for AM memory and modify the appropriate timer(s).
1006	Link set timer data in AM memory is invalid.	These error messages indicate that there are invalid link set timer data items in AM memory. These can be corrected by doing a link set timer recent change for AM memory. Consider following this with a link set timer recent change that copies the AM memory link set timers to disk.
1007	Link set threshold data in AM memory differs from that in the disk file.	Error messages of this type reflect the fact that the link set threshold data in the AM does not match the data on the disk. If the data in the AM is deemed correct, then do a link set threshold recent change that copies the AM memory link set thresholds to disk. If the data on disk is correct, then do a link set threshold recent change for AM memory and modify the appropriate threshold(s).
1008	Link set threshold data in AM memory is invalid.	These error messages indicate that there are invalid link set threshold data items in AM memory. These can be corrected by doing a link set threshold recent change for AM memory. Consider following this with a link set threshold recent change that copies the AM memory link set thresholds to disk.
1009	Link timer data in AM memory differs from that in the disk file.	Error messages of this type reflect the fact that the link timer data in the AM does not match the data on the disk. If the data in the AM is deemed correct, then do a link parameter (not link node data) recent change that copies the AM memory link parameters to disk. If the data on disk is correct, then do a link parameter (not link node data) recent change for AM memory and modify the appropriate timer(s).
1010	Link timer data in AM memory is invalid.	These error messages indicate that there are invalid link timer data items in AM memory. These can be corrected by doing a link

		parameter (not link node data) recent change for AM memory. Consider following this with a link parameter recent change that copies the AM memory link parameters to disk.
1011	Signal unit error rate monitor (SUERM) threshold data in AM memory differs from that in the disk file.	Error messages of this type reflect the fact that the SUERM threshold data in the AM does not match the data on the disk. If the data in the AM is deemed correct, then do a link parameter (not link node data) recent change that copies the AM memory link parameters to disk. If the data on disk is correct, then do a link parameter (not link node data) recent change for AM memory and modify the SUERM threshold(s) accordingly.
1012	SUERM threshold data in AM memory is invalid.	These error messages indicate that there are invalid SUERM threshold data items in AM memory. These can be corrected by doing a link parameter (not link node data) recent change for AM memory. Consider following this with a link parameter recent change that copies the AM memory link parameters to disk.
1013	Invalid version number in disk file.	The version number on the disk does not match the version number the audit expects. The user should check whether the disk file was created or retrofitted, and whether it is in the proper format.
1014	Too many bytes in the protocol timer/parameter disk file.	This error is corrected during the audit. A new disk file will be created and populated from the data in the AM. Re-running the audit will verify that the condition no longer exists.

NLI

Software Release: 5E10 and later

1. NETWORK LINK INTERFACE (NLI) AUDIT

The network link interface (NLI) circuits reside in the SM-2000 switching modules but are considered part of the office network and timing complex (ONTC).

Status information about these circuits is kept in the peripheral configuration database (PCD). Several audits check the PCD, including the communication link (CLNK) audit, the dual link interface (DLI) audit, the peripheral configuration database (PCD) audit, and the time-multiplexed switch link (TMSLNK) audit.

During initialization the NLI audit will only correct software states. No hardware versus software checks are made; hardware recovery actions are not performed.

During segmented mode, if the SMKP environment is fully operational and there is a major alarm in the office network and timing complex (ONTC), the full audit is run, which can cause hardware recovery actions.

The audit runs through the units one switching module (SM) at a time and within the SM, one peripheral unit per real time segment.

2. DATA STRUCTURES

The NLI audit uses the following data structures:

- Switch maintenance arrays
- RLMODATT relation (stores switching module attributes)
- RLSAUTISS relation (stores automatic timeslot switching information)
- RLNLI_LNK relation (stores SM/NLI to TMSLNK mapping information).

3. AUDIT CHECKS

The audit makes software checks, vertical checks, external checks, and deferred maintenance administrator monitor (DMAM) checks.

- (a) The audit checks for the internal consistency within the dynamic peripheral configuration database for the NLI and checks the consistency between the office-dependent data (ODD) and the PCD. The NLI must be in an allowable state. Allowable states include active (ACT) if the SM is operational, out-of-service unequipped (OOS UNEQ) if the SM is unequipped or not a local SM, and out-of-service manual remove (OOS MAN RMV) if the SM is in a special growth state (SGRO).
 - INDEX (Check for out-of-range unit index).
 - MODATT (Check for the existence of a valid rIMODATT tuple).
 - NLI_LNK (Check for the existence of a valid rNLI_LNK tuple).
 - PCD (Check for invalid Peripheral Configuration Data).
 - ZAP (A reset was taken during the audit).

- (b) The vertical check ensures the consistency of the NLI state in the PCD with the state of its controlling hardware unit, the office network and timing complex common (ONTCCOM).
 - NLI_FE (Check for NLI out of service, but parent ONTC in service).
 - NLI_PCD (Check for invalid network link interface state).
- (c) The audit makes external checks on special data such as data delivery bits.
 - NLI_DD_UNIQUE (Check for invalid unique part of NLI DD).
 - NLI_DD_UNIVERSAL (Check for invalid universal part of NLI DD).
 - RANGE (Check for equipage between RLNLI_LNK and RLMODATT).
 - UNEQUIP (Verify equipage status in PCD against RLMODATT).
- (d) For out-of-service units in transient states, the audit ensures that some activity is scheduled as pending on DMAM's queue.
 - CSANLI (Check for active NLI, but out of service TMSLNKs).
 - NLI_DMAM (Check for deferred maintenance work on out of service NLI).

Before taking corrective action, the audit ensures that the NLI is not in a transient state by checking the switch maintenance and recovery software (SMARS) queue. The SMARS queue contains a record of all maintenance activities. If fault recovery is in progress, the audit is rescheduled. If fault recovery is not in progress, an error is reported.

4. ERROR HANDLING

The type of error determines the actions the audit takes.

If a software internal check or a vertical check fails, the audit places the NLI in a valid state.

If the external check fails, the audit assumes that the NLI state is correct and updates any corresponding data to agree with the state of the NLI.

If the DMAM check fails, the NLI is in a stuck state. The audit places it in a valid state and requests a conditional restoration of the NLI.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CSANLI

Error Code:	CSANLI
Error Description:	The NLI is in an active state but both TMSLNKs are out of service trouble.
Possible Error Effect:	The NLI is active when it should not be.
Corrective Action Taken:	The audit puts the NLI in an out of service state and sends a conditional restore for the NLI.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the NLI which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 2 Error Code: INDEX

Error Code:	INDEX
Error Description:	The audit's cursor is out-of-range.
Possible Error Effect:	Audit asserts or errors from trying to read information for a unit which does not exist.
Corrective Action Taken:	The cursor is reset to a valid value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Audit's cursor value found to be bad.
<i>Dump</i> =	NA.

Table 3 Error Code: MODATT

Error Code:	MODATT
Error Description:	The audit has found a problem with the RLMODATT relation in the ODD. This error will continue to print out until fixed by the service provider.
Possible Error Effect:	An SM may be having communication problems.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats check, the RLMODATT relation tuple whose key is the SM number given in GOOD DATA. If the tuple does not exist, insert it using ODBE.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The error code received while attempting to access the Data Base.
<i>Good Data</i> =	SM number used to read the RLMODATT relation.
<i>Logical Index</i> =	Number of the NLI which was found in error.
<i>Dump</i> =	Unit state (AUBIG_ST).

Table 4 Error Code: NLI_DD_UNIQUE

Error Code:	NLI_DD_UNIQUE
Error Description:	The unique part of the data delivery data for the NLI was found to be in error.
Possible Error Effect:	The NLI may be configured wrong.
Corrective Action Taken:	The audit updates the data delivery data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the NLI which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 5 Error Code: NLI_DD_UNIVERSAL

Error Code:	NLI_DD_UNIVERSAL
Error Description:	The universal part of the data delivery data for the NLI was found to be in error.
Possible Error Effect:	The NLI may be configured wrong.
Corrective Action Taken:	The audit updates the data delivery data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the NLI which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 6 Error Code: NLI_DMAM

Error Code:	NLI_DMAM
Error Description:	The NLI is in an out of service state where there should be some deferred work being done on it but deferred maintenance administrator and monitor (DMAM) does not have an entry in its queue for this unit.
Possible Error Effect:	The NLI is out-of-service with no automatic action pending to restore it to service.
Corrective Action Taken:	Audit requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the NLI which was found in error.
<i>Logical Index =</i>	ONTC side the error was found on (0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 7 Error Code: NLI_FE

Error Code:	NLI_FE
Error Description:	The NLI was in an out of service family state but the parent ONTC is in service.
Possible Error Effect:	The NLI is out of service when it should not be.
Corrective Action Taken:	The audit restores the NLI.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the NLI which was found in error.
<i>Logical Index =</i>	ONTC side the error was found on (0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 8 Error Code: NLI_LNK

Error Code:	NLI_LNK
Error Description:	The audit has found a problem with the RLNLI_LNK relation in the ODD. The entry for the SM whose number is the logical key, or the NLI whose number is included in the dump, has been found to be incorrect. This error will continue to print out until fixed by the service provider.
Possible Error Effect:	The NLI may be configured incorrectly.
Corrective Action Taken:	None.
Manual Action Required:	Check the RLNLI_LNK relation entry whose key is the same as the logical key in the audit output. This entry has been found to be incorrect and should be corrected.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DBfrdtup return code.
<i>Good Data =</i>	The module number used as the key.
<i>Logical Index =</i>	Number for the SM which has been found to have an incorrect ODD.
<i>Dump =</i>	Unit's state (AUBIG_ST structure).

Table 9 Error Code: NLI_PCD

Error Code:	NLI_PCD
Error Description:	A NLI has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Number of the NLI which was found in error.
<i>Good Data =</i>	SM number.
<i>Logical Index =</i>	ONTC side the error was found on (0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 10 Error Code: PCD

Error Code:	PCD
Error Description:	The audit has found an error which it cannot correct. This may be due to a bad ODD, bad static data or the audit recovery table needs to be changed. This error will continue to print out until fixed by the service provider.
Possible Error Effect:	The audit will not recover this error.
Corrective Action Taken:	The audit will not recover this error.
Manual Action Required:	Check if the ODD is correct for the PCD. The relations this includes are: RLSAUTISS - For automatic time-slot switching. RLMODATT - For the SMs.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Number of the NLI which was found in error.
<i>Logical Index =</i>	ONTC side the error was found on (0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 11 Error Code: RANGE

Error Code:	RANGE
Error Description:	An NLI is unequipped according to RLNLI_LNK (rNLI_LNK.eqstat != SMOPER and != SMGROW) but should be equipped according to RLMODATT (rMODATT.eq_lis > NLI number).
Possible Error Effect:	The NLI status is inconsistent in the ODD. The effect of this depends on which relation is used by which functionality. For example, if the NLI is marked unequipped when it should be equipped, SM connectivity may be lost. If the NLI is marked equipped when it should not be, operational code may try to use the unequipped NLI, possibly resulting in lost traffic and/or asserts. This error will continue to be reported until manual action is taken to correct the ODD.
Corrective Action Taken:	None.
Manual Action Required:	Either rMODATT.eq_lis or rNLI_LNK.eqstat need to be updated. The rMODATT tuple in question is keyed by the SM number associated with this NLI (given in good data). The RLNLI_LNK tuple is keyed by the SM number (good data) and NLI number (bad data).
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Number of the NLI being audited.
<i>Good Data =</i>	SM number associated with the NLI being audited.
<i>Logical Index =</i>	ONTC side the error was found on (0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 12 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	An NLI's equipage status in the PCD is inconsistent with RLMODATT. Specifically, either: - The NLI is marked as unequipped in the PCD (PCD.state.qual1 = SMQ1UNEQ), but the NLI number is less than or equal to the number of equipped NLIs on the switching module (NLI number $\leq$ rMODATT.eq_lis). - The NLI is marked as equipped or grow in the PCD (PCD.state.qual1 != SMQ1UNEQ), but the NLI number is greater than the number of equipped NLIs on the switching module (NLI number > rMODATT.eq_lis).
Possible Error Effect:	If the NLI is marked unequipped when it should be equipped, SM connectivity

	may be lost. If the NLI is marked equipped when it should not be, operational code may try to use the unequipped NLI possibly resulting in lost traffic and/or asserts.
Corrective Action Taken:	The audit writes the NLI's PCD state to agree with that in rNLI_LNK.eqstat (where rNLI_LNK is keyed by the SM number and NLI number given in good data and bad data). The new equippage state appears in the data block dumped below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Number of the NLI being audited.
<i>Good Data</i> =	SM number associated with the NLI being audited.
<i>Logical Index</i> =	ONTC side the error was found on (0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 13 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Reinitialize all data that could have made the audit cause a system reset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into PCbthead[].
<i>Dump</i> =	NA.

NMCC

Software Release: 5E10 and later

1. NETWORK MANAGEMENT CODE CONTROLS (NMCC) AUDIT

Network management code controls (NMCC) limit traffic to particular destination codes that are determined to be difficult to reach, freeing up network resources for calls that are more likely to complete. These controls are implemented by means of several global data structures that are audited by the NMCC audit.

The NMCC audit is part of the routing and terminal allocation (RTA) audits subgroup.

The NMCC audit runs in the operational kernel process (OKP) and the switching module (SM) environments in the initialization and segmented modes.

The NMCC audit runs in the OKP environment in the initialization and segmented modes.

2. DATA STRUCTURES

The code controls data structures may be broken into two parts, the list and the tree. The list consists of the NMcclist array which is the master copy of the controls. Each element of this array represents a single code control and contains all data associated with that control. The tree consists of digit nodes (NMdigpool array) and control nodes (NMcntrlpool array) and is used to provide fast access to the list. The digit nodes are primarily used to locate a given digit string and each one contains a forward pointer for digits '0' - '9', '*' and '#'; a backward pointer to the previous digit node; and a pointer to a list of control nodes. The control nodes link the tree back to the master list and are kept in doubly linked lists off of a digit node. Each control node contains a forward, backward, and owner pointer. The control nodes also contain a pointer to an element in NMcclist. The busy/idle status of the tree structures is controlled using the NMfreelist array, which contains a busy/idle bit for each structure in NMdigpool and NMcntrlpool.

3. AUDIT CHECKS

The NMCC audit performs range checks on the data in the NMcclist. Once the list is verified, it checks the tree structures, verifying all linkages.

4. ERROR HANDLING

If the audit finds a list error on the administrative module (AM), the invalid control is removed and the tree is rebuilt. If a list error is found on the SM, the SM list is repopulated from the AM list. If the audit finds a tree error, the tree is rebuilt from the list.

If the audit finds a list error, the invalid control entry is removed and the tree is rebuilt. If the audit finds a tree error, the tree is rebuilt from the list. See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: OLEGAL_CL

Error Code:	OLEGAL_CL
Error Description:	This error code indicates that a code control was found when the code control list (cclist) was traversed by the audit.
Possible Error Effect:	Idle code controls that are not shown may reduce the space needed to store busy controls. If this condition is allowed to continue, a number of code controls may become unavailable.
Corrective Action Taken:	The audit stops and rebuilds the cclist. Network management administration (NMA) is notified of the changes.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Current cntrl_id.
<i>Bad Data</i> =	Entry in the cntrlpool array in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entry in the cntrlpool array in error.

Table 2 Error Code: 1LEGAL_CL

Error Code:	1LEGAL_CL
Error Description:	The code controls' list is not in the proper order.
Possible Error Effect:	The code controls will not be consistent.
Corrective Action Taken:	The code controls' list will be rebuilt and NMA will be notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Entry in the cntrlpool array in error.
<i>Bad Data</i> =	Forward pointer of the bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	cntrlpool entry that was passed in.

Table 3 Error Code: 2LEGAL_CL

Error Code:	2LEGAL_CL
Error Description:	The next control is checked to see if they are in the correct order.
Possible Error Effect:	The code control lists may be inconsistent.
Corrective Action Taken:	The list is rebuilt and NMA is notified.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Bad cntrlpool entry.
<i>Bad Data</i> =	Address of the bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	cntrlpool entry that was passed in.

Table 4 Error Code: ANNCMNT

Error Code:	ANNCMNT
Error Description:	The code controls' disposition is out-of-range.
Possible Error Effect:	The disposition is invalid.
Corrective Action Taken:	The erroneous entry is nulled out and the structures are rebuilt. NMA is then notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address index of the erroneous cclist entry.
<i>Bad Data</i> =	Index of the erroneous cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Index of the erroneous cclist entry.

Table 5 Error Code: BKDTOWH

Error Code:	BKDTOWH
Error Description:	The code control node does not point backward to the correct cclist entry.
Possible Error Effect:	The link between the cntrlpool and cclist are incorrect.
Corrective Action Taken:	The audit is stopped and the cclist is rebuilt. Also, NM personnel are notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Index of the bad cntrlpool entry and the bad cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad cntrlpool entry, the cclist entry and the digits gotten by traversing the tree backward.

Table 6 Error Code: CCINDEX

Error Code:	CCINDEX
Error Description:	The cclist entry is not valid.
Possible Error Effect:	A code control may be lost or erroneous.
Corrective Action Taken:	The cclist is rebuilt and NMA is notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool array entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	cntrlpool entry that was passed in.

Table 7 Error Code: CCONON

Error Code:	CCNON
Error Description:	An active code control's indptr is null and the code control is not topnode.
Possible Error Effect:	Since this is a busy node, information on a code control may be missing.
Corrective Action Taken:	The code controls' list is rebuilt and NMA is notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad digit pool entry.
<i>Bad Data</i> =	Bad digit pool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad digit pool entry.

Table 8 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The digit pool's back-entry's forward pointer is incorrect.
Possible Error Effect:	The links between digpool nodes are incorrect.
Corrective Action Taken:	The code control is removed and the list is rebuilt. NMA will then be notified of the changes.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad digpool entry.
<i>Bad Data</i> =	Bad digpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Digpool entry being audited and the digpool entry it points back to.

Table 9 Error Code: CTL_NMCC

Error Code:	CTL_NMCC
Error Description:	The pointers to an idle control are not null.
Possible Error Effect:	Information is being stored on a non-existent code control.
Corrective Action Taken:	The audit stops and rebuilds the code control's list. NMA is notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	cntrlpool entry that was passed in.

Table 10 Error Code: DIG_NMCC

Error Code:	DIG_NMCC
Error Description:	The top node's back pointer is out-of-range.
Possible Error Effect:	The digpool's entry is being linked to an invalid pointer.
Corrective Action Taken:	The audit stops and rebuilds the list. NMA is notified of the changes.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the erroneous digpool entry index.
<i>Bad Data</i> =	Erroneous digpool entry index.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Digpool entry that is in error.

Table 11 Error Code: DRPTRBAD

Error Code:	DRPTRBAD
Error Description:	The linkage between this control node and its predecessor is wrong.
Possible Error Effect:	A control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the code control list and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the bad cntrlpool entry.
<i>Bad Data</i> =	The bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The two control pool entries involved in the error.

Table 12 Error Code: FBKDPTR

Error Code:	FBKDPTR
Error Description:	This cntrl node belongs to a carrier-only control and the owner is not top node.
Possible Error Effect:	The links between the cclist node and the cntrlpool node are incorrect.
Corrective Action Taken:	The audit stops and the cclist is rebuilt. NMA is notified.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad cntrlpool entry and the bad cclist entry.

Table 13 Error Code: FFWDPTR

Error Code:	FFWDPTR
Error Description:	No special prefix (sprefix) (carrier) is associated with this cntrl_id and one should be there.
Possible Error Effect:	This cntrl_id may be linked to the wrong cclist entry.
Corrective Action Taken:	The audit stops and rebuilds the cclist. NMA is notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad cntrlpool entry and the bad cclist entry.

Table 14 Error Code: FWDTOWH

Error Code:	FWDTOWH
Error Description:	There are erroneous digits in the buffer.
Possible Error Effect:	A code control may be placed on a number that is not on the control list.
Corrective Action Taken:	The audit stops and rebuilds the cclist. NMA is notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Erroneous cntrlpool entry.

Table 15 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER

Error Description:	This cntrlpool entry is not linked to its correct digpool entry.
Possible Error Effect:	A code control may be placed on the wrong number.
Corrective Action Taken:	The audit will stop and rebuild the cclist. Then, NMA will be notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Entry in the cntrlpool array in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	cntrlpool entry that was passed in.

Table 16 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	This control node and its predecessor do not have the same owner.
Possible Error Effect:	A control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the code control list and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the bad cntrlpool entry.
<i>Bad Data =</i>	The bad cntrlpool entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The two control pool entries involved in the error.

Table 17 Error Code: INVOWNER2

Error Code:	INVOWNER2
Error Description:	This control node and its successor do not have the same owner.
Possible Error Effect:	A control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the code control list and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the bad cntrlpool entry.
<i>Bad Data =</i>	The bad cntrlpool entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The two control pool entries involved in the error.

Table 18 Error Code: LEGAL_CL

Error Code:	LEGAL_CL
Error Description:	This cntrl id's owner does not point to the correct digpool entry.
Possible Error Effect:	A bad linkage between the cntrlpool and the digpool entries.
Corrective Action Taken:	The audit stops and the cclist is rebuilt. NMA is then notified.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the bad cntrlpool entry.
<i>Bad Data =</i>	Bad cntrlpool entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	cntrlpool entry that was passed in.

Table 19 Error Code: LIST

Error Code:	LIST
Error Description:	This code control is not on the list.
Possible Error Effect:	The code control for this number should be active.
Corrective Action Taken:	The cclist is rebuilt and stops the audit. NMA is notified that a change has occurred in the cclist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the erroneous cclist entry.
<i>Bad Data =</i>	Erroneous cclist entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.

<i>Dump</i> =	cclist entry passed in.

Table 20 Error Code: LNKILST

Error Code:	LNKILST
Error Description:	An invalid indptr was found in a digpool entry.
Possible Error Effect:	Code controls may be missing.
Corrective Action Taken:	The code controls list will be rebuilt and NMA will be notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous digpool entry.
<i>Bad Data</i> =	The erroneous digpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The bad digpool entry.

Table 21 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	An invalid forward pointer was found in a digpool entry.
Possible Error Effect:	Code controls may be missing.
Corrective Action Taken:	The code controls list will be rebuilt and NMA will be notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous digpool entry.
<i>Bad Data</i> =	The erroneous digpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The bad digpool entry.

Table 22 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The code control's special prefix (sprefix) (carrier) is null.
Possible Error Effect:	The carrier for this office is invalid.
Corrective Action Taken:	The audit stops, rebuilds the code control tree, and notifies NMA that a change has taken place in the code controls.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous cclist entry.
<i>Bad Data</i> =	Erroneous cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Erroneous cclist entry.

Table 23 Error Code: LST_NMCC

Error Code:	LST_NMCC
Error Description:	The code control was turned off, but data was found in the structure.
Possible Error Effect:	There may be data in an idle code control entry.
Corrective Action Taken:	The audit stops, rebuilds the code control tree, and notifies the NMA personnel of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the cclist entry that was in error.
<i>Bad Data</i> =	Bad cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad cclist entry.

Table 24 Error Code: MASTER_OWN

Error Code:	MASTER_OWN
Error Description:	This code control entry does not point to the correct owner.
Possible Error Effect:	The linkage between NMcntrlpool and digpool are incorrect.
Corrective Action Taken:	The audit stops, rebuilds the code control list, and notifies NMA of the change.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Entry in the cntrlpool array involved in the error.

Table 25 Error Code: MEMBER

Error Code:	MEMBER
Error Description:	This entry's digit count exceeds the maximum allowed or the digits contained are erroneous.
Possible Error Effect:	A code control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the cc tree, and notifies the NMA personnel that a change has occurred in the code control list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous cclist entry.
<i>Bad Data</i> =	Bad cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad cclist entry.

Table 26 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	This digpool entry's back pointer is null.
Possible Error Effect:	The linkage between the digpool entry and the cclist entry is incorrect.
Corrective Action Taken:	The cclist is rebuilt after the audit stops. NMA is notified of the change in the cclist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad digpool entry.
<i>Bad Data</i> =	Bad digpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Digit pool entry that is in error.

Table 27 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	Data is contained in an idled control.
Possible Error Effect:	A code control contains garbaged data.
Corrective Action Taken:	The audit stops after the error is found. It rebuilds the cclist and notifies the NMA personnel of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad digpool entry.
<i>Bad Data</i> =	Bad digpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad digit pool entry.

Table 28 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	This idle entry has pointers.
Possible Error Effect:	Garbaged data is contained in this null entry.
Corrective Action Taken:	The audit stops, rebuilds the cclist and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the digit pool entry in error.
<i>Bad Data</i> =	Bad digit pool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.

<i>Dump</i> =	Bad digit pool entry.

Table 29 Error Code: NULLCNTL

Error Code:	NULLCNTL
Error Description:	This code control has no data.
Possible Error Effect:	There might be missing data.
Corrective Action Taken:	NMA is notified that a change has occurred after the audit stops and the cclist is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad list entry.
<i>Bad Data</i> =	Bad list entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad list entry.

Table 30 Error Code: ORCOUNTS

Error Code:	ORCOUNTS
Error Description:	The digits are not in the right order.
Possible Error Effect:	A control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the cclist and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntlpool entry.
<i>Bad Data</i> =	Bad cntlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	cntlpool entry and the digpool entry involved in the error.

Table 31 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An invalid gap interval was found in the code controls list.
Possible Error Effect:	Controls might have been applied incorrectly.
Corrective Action Taken:	The code controls list will be rebuilt and NMA will be notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous cclist entry.
<i>Bad Data</i> =	The erroneous cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The bad cclist entry.

Table 32 Error Code: OWNER

Error Code:	OWNER
Error Description:	This entry's forward pointer is out-of-range.
Possible Error Effect:	A control may be on the wrong number.
Corrective Action Taken:	The audit stops and rebuilds the code control list. NMA is notified of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous cclist entry.
<i>Bad Data</i> =	Erroneous cclist entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Erroneous cclist entry.

Table 33 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The prefixes are not in the right order.
Possible Error Effect:	A control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the code control list and notifies NMA of the change.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the bad cntrlpool entry.
<i>Bad Data</i> =	The bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The cntrlpool entry and the digpool entry involved in the error.

Table 34 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	The linkage between this control node and its successor is wrong.
Possible Error Effect:	A control may be placed on the wrong number.
Corrective Action Taken:	The audit stops, rebuilds the code control list and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the bad cntrlpool entry.
<i>Bad Data</i> =	The bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The two control pool entries involved in the error.

Table 35 Error Code: TRAFFIC

Error Code:	TRAFFIC
Error Description:	An rINMDSNARC tuple contains invalid ARC control data.
Possible Error Effect:	Controls might have been applied incorrectly.
Corrective Action Taken:	Tuple is initialized and controls are set to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the current rINMDSNARC tuple.
<i>Bad Data</i> =	See dump information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	dsnarc_id of the rINMDSNARC tuple.
<i>Dump</i> =	rINMDSNARC tuple.

Table 36 Error Code: TRAVERSE

Error Code:	TRAVERSE
Error Description:	This cntrlpool entry has more than one pointer to the digpool.
Possible Error Effect:	A bad linkage between the cntrlpool and the digpool.
Corrective Action Taken:	The code control list is rebuilt and NMA is notified.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the bad cntrlpool entry.
<i>Bad Data</i> =	Bad cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Bad cntrlpool entry.

Table 37 Error Code: WBKDPTR

Error Code:	WBKDPTR
Error Description:	This cntrlpool entry exceeds the maximum size of the control pool.
Possible Error Effect:	Controls may be placed on the wrong numbers.
Corrective Action Taken:	The audit rebuilds the code control list and notifies NMA of the change.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous cntrlpool entry.
<i>Bad Data</i> =	Erroneous cntrlpool entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Erroneous cntrlpool entry.

NMDCOFC

Software Release: 5E10 and later

1. NETWORK MANAGEMENT DIRECTLY CONNECTED OFFICE (NMDCOFC) AUDIT

Automatic route cancellation is a feature of the automatic voice network (AUTOVON). Audits are needed to ensure the accuracy of information related to automatic route cancellation because the routing and terminal allocation (RTA) subsystem will not be able to retrieve the control information associated with a trunk group if a key or control ID is corrupted.

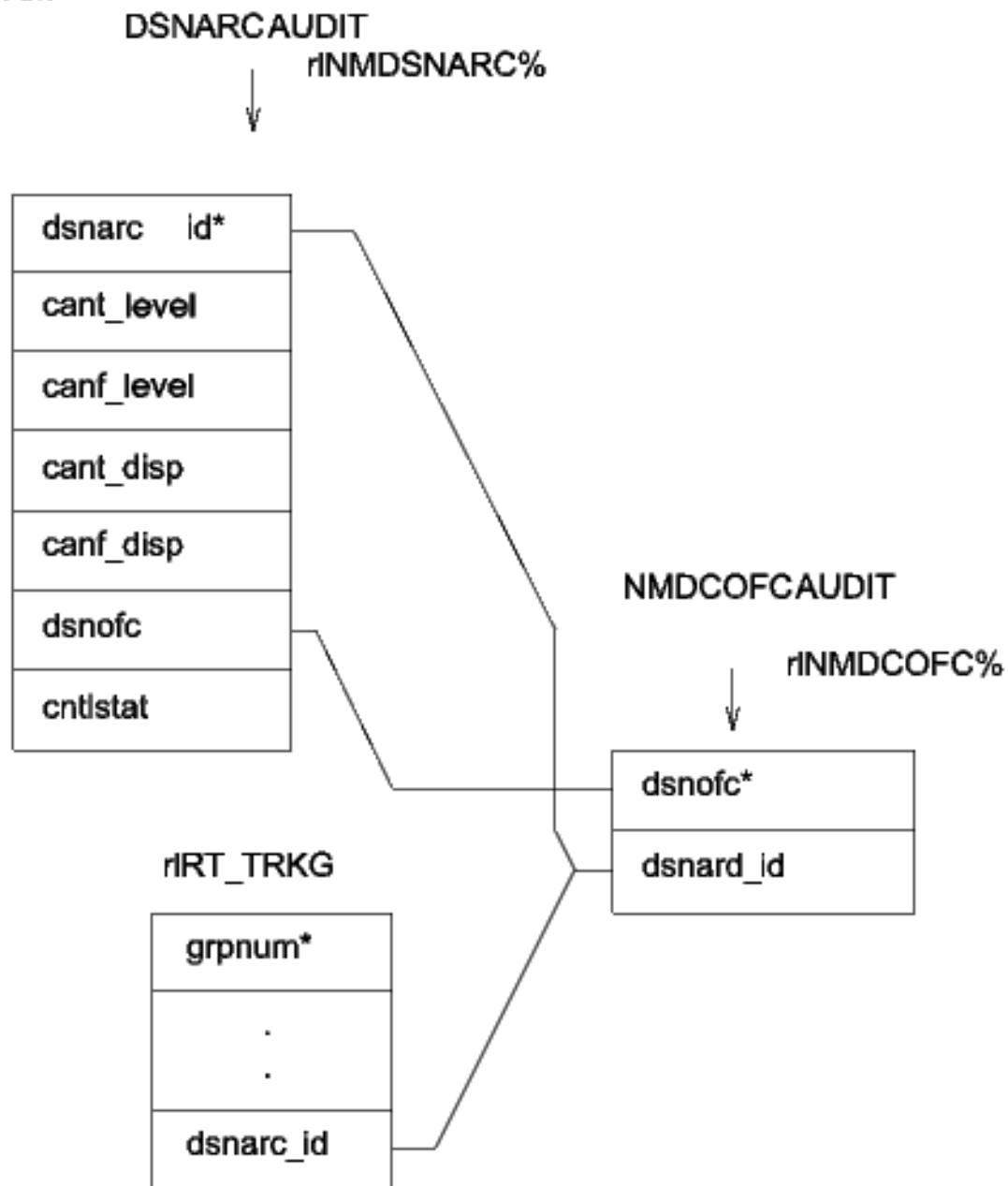
The NMDCOFC audit is part of the administrative services audits subgroup. This audit verifies the data structures used by the network management subsystem of a DSN switch. It runs in the intraprocessor-AM (OKP) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The NMDCOFC audit checks the network management directly connected office relation (RLNMDCOFC) and the network management defense switched network alternate route cancellation relation (RLNMDSNARC). These data structures are shown in Figure 1 .

SM#RSM*

AM



* - Key of the relation

% - Audited relation

_ - NON-STANDARD linkages

Figure 1 (NMDCOFC-1) Data Structures of the NMDCOFC Audit

3. AUDIT CHECKS

The NMDCOFC audit invokes the static driver to step through the RLNMDCOFC relation and retrieve tuples. For each tuple retrieved, the audit performs the following semantic checks:

- (a) The range of the `dsnarc_id` attribute is verified. Valid values are from 1 through 63.
- (b) If the `dsnarc_id` is in range, it is used to retrieve a corresponding tuple in RLNMDSNARC.
- (c) If a corresponding tuple is found, the key attribute `dsnofc` of RLNMDCOFC is compared to its mate in RLNMDSNARC.

4. ERROR HANDLING

If the audit detects a range error, manual action is required to fix it. If no matching tuple is found, or if the `dsnofc` attribute of the retrieved tuple is in error, the tuple is initialized.

The audit schedules the NMDSNARC audit as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DSNARCID

Error Code:	DSNARCID
Error Description:	A good rINMDSNARC tuple is garbaged and there is no tuple to match an rINMDCOFC tuple.
Possible Error Effect:	ARC controls might have been applied to the wrong office.
Corrective Action Taken:	Tuple is initialized and ARC control is set to off for that office. ARC control has to be turned on if necessary.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINMDSNARC tuple whose physical key is same as "dsnarc_id" of corresponding rINMDCOFC tuple.
<i>Bad Data =</i>	Contents of the "dsnarc_id" in rINMDSNARC tuple.
<i>Good Data =</i>	Contents of the "dsnarc_id" in rINMDCOFC tuple.
<i>Logical Index =</i>	"dsnarc_id" of the rINMDCOFC tuple.
<i>Dump =</i>	NA.

Table 2 Error Code: OFCNAME

Error Code:	OFCNAME
Error Description:	An rINMDSNARC tuple has bad data in the office name (dsnofc).
Possible Error Effect:	ARC controls might have been applied to the wrong office.
Corrective Action Taken:	Tuple is initialized and ARC control is set to off for that office. NM will turn ARC control back on if necessary.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the "dsnofc" field in the rINMDSNARC tuple.
<i>Bad Data =</i>	Contents of the "dsnofc" field in the rINMDSNARC tuple.
<i>Good Data =</i>	Contents of the "dsnofc" field in the rINMDCOFC tuple.
<i>Logical Index =</i>	"dsnarc_id" of the rINMDSNARC tuple.
<i>Dump =</i>	rINMDSNARC tuple.

Table 3 Error Code: RANGE

Error Code:	RANGE
Error Description:	An rINMDCOFC tuple has out of range value in the "dsnarc_id" field.
Possible Error Effect:	May result in assert failures.
Corrective Action Taken:	None.

Manual Action Required:	Use ODBE or recent change to update the rINMDCOFC tuple. This error will persist until it is fixed manually by office personnel.
Dump Description:	
<i>Error Address</i> =	Address of the "dsnarc_id" field in the rINMDCOFC tuple.
<i>Bad Data</i> =	Contents of the "dsnarc_id" field in the rINMDCOFC tuple.
<i>Good Data</i> =	None.
<i>Logical Index</i> =	"dsnofc" of the rINMDCOFC tuple.
<i>Dump</i> =	NA.

NMDSNARC

Software Release: 5E10 and later

1. NETWORK MANAGEMENT DEFENSE SWITCHED NETWORK ALTERNATE ROUTE CANCELLATION (NMDSNARC) AUDIT

Automatic route cancellation is a feature of the automatic voice network (AUTOVON). Audits are needed to ensure the accuracy of information related to automatic route cancellation because the routing and terminal allocation (RTA) subsystem will not be able to retrieve the control information associated with a trunk group if a key or control ID is corrupted.

The NMDSNARC audit is part of the administrative services audits subgroup. This audit verifies the data structures used by the network management subsystem of a DSN switch. It runs in the intraprocessor-AM (OKP) environment in the segmented modes.

2. DATA STRUCTURES

The audit checks the network management defense switched network alternate route cancellation relation (RLNMDSNARC), a dynamic relation in the administrative module (AM) by referencing the network management directly connected office relation (RLNMDCOFC), a static relation in the AM. These data structures are shown in Figure 1.

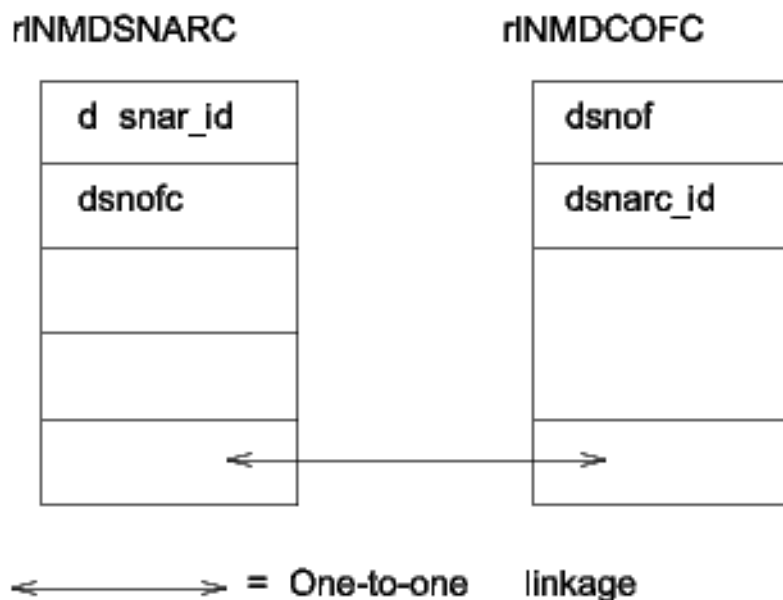


Figure 1 (NMDSNARC-1) Data Structures of the NMDSNARC Audit

3. AUDIT CHECKS

The audit invokes the linkage driver (DRLINK) to retrieve rNMDSNARC tuples. The audit makes the following checks:

- The non-key field dsnofc of rNMDSNARC, is used to retrieve a corresponding tuple in rNMDCOFC, since rNMDCOFC is keyed on dsnof.
- If a corresponding tuple is found, the key dsnar_id of rNMDSNARC is compared to its counterpart in rNMDCOFC.

- (c) If the key check passes, the audit checks the control and level fields. The algorithm for checking the controls is as follows:

cntlstat	cant_level	canf_level	ralid
off	off	off	yes
on	on	off	yes
on	off	on	yes
on	on	on	yes

If any other combination is found, a "TRAFFIC" error is reported. This audit also checks the ranges of cant_level, canf_level, cant_disp and canf_disp.

4. ERROR HANDLING

If the audit detects an error in the control fields, it initializes the rNMDSNARC tuple. If the audit detects an error in the key fields it initializes the rNMDSNARC relation.

The audit schedules the NMDCOFC audit as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DSNARCID

Error Code:	DSNARCID
Error Description:	There is an rNMDSNARC tuple with a garbaged key.
Possible Error Effect:	Controls might have been applied to the wrong office.
Corrective Action Taken:	Relation is initialized and controls are set to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rNMDSNARC.
<i>Bad Data</i> =	Contents of the "dsnarc_id" in rNMDSNARC tuple.
<i>Good Data</i> =	Contents of the "dsnarc_id" in rNMDCOFC tuple.
<i>Logical Index</i> =	"dsnarc_id" of the rNMDCOFC tuple.
<i>Dump</i> =	rNMDSNARC tuple.

Table 2 Error Code: FDISP

Error Code:	FDISP
Error Description:	An rNMDSNARC tuple contains invalid values in the canf disposition field.
Possible Error Effect:	An assert will fire in case of blocking.
Corrective Action Taken:	Tuple is initialized and controls are set to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the current rNMDSNARC tuple.
<i>Bad Data</i> =	Contents of the "canf_disp" field in rNMDSNARC tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	"dsnarc_id" of the rNMDSNARC tuple.
<i>Dump</i> =	rNMDSNARC tuple.

Table 3 Error Code: OFCNAME

Error Code:	OFCNAME
Error Description:	There is an rNMDSNARC tuple with a office name that doesn't exist in rNMDCOFC.
Possible Error Effect:	Controls might have been applied to the wrong office.
Corrective Action Taken:	Relation is initialized and controls are set to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rNMDSNARC tuple.

<i>Bad Data</i> =	Contents of the "dsnarc_id" in rNMDSNARC tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	"dsnarc_id" of the rNMDSNARC tuple.
<i>Dump</i> =	rNMDSNARC tuple.

Table 4 Error Code: TDISP

Error Code:	TDISP
Error Description:	An rNMDSNARC tuple contains invalid values in the cant disposition field.
Possible Error Effect:	An assert will fire in case of blocking.
Corrective Action Taken:	Tuple is initialized and controls are set to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the current rNMDSNARC tuple.
<i>Bad Data</i> =	Contents of the "cant_disp" field in rNMDSNARC tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	"dsnarc_id" of the rNMDSNARC tuple.
<i>Dump</i> =	rNMDSNARC tuple.

Table 5 Error Code: TRAFFIC

Error Code:	TRAFFIC
Error Description:	An rNMDSNARC tuple contains invalid ARC control data.
Possible Error Effect:	Controls might have been applied incorrectly.
Corrective Action Taken:	Tuple is initialized and controls are set to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the current rNMDSNARC tuple.
<i>Bad Data</i> =	See dump information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	"dsnarc_id" of the rNMDSNARC tuple.
<i>Dump</i> =	rNMDSNARC tuple.

NMTR

Software Release: 5E10 and later

1. NETWORK MANAGEMENT TRUNK RESERVATION (NMTR) AUDIT

The NMTR system audit verifies the data used by the network management (NM) trunk reservation (TR) control. This control is used by the network manager to reserve a number of individual trunks within a trunk group for preferred traffic during conditions of network congestion or overload. If these data structures become corrupt, incorrect calls could be blocked or the NM may not be able to install the necessary controls.

2. DATA STRUCTURES

The following data structures are used in this audit:

- RLTR** = This relation stores the information associated with the individual TR controls currently assigned. There is one tuple on the busy list for each active TR control. All other tuples are on the idle list.
- RLDB_HEADC** = This relation contains the head cells for the busy and idle lists for the RLTR relations. The relation is the owner of RLTR.
- RLGRP_TGC** = This relation is used to link the rLTR tuple to the appropriate dynamic trunk group relation. The relation is the linkage used on a per-call basis to check for TR controls, and has a one-to-one linkage to RLTR.

Figure 1 illustrates the linkages of RLTR that are checked by the NMTR audit.

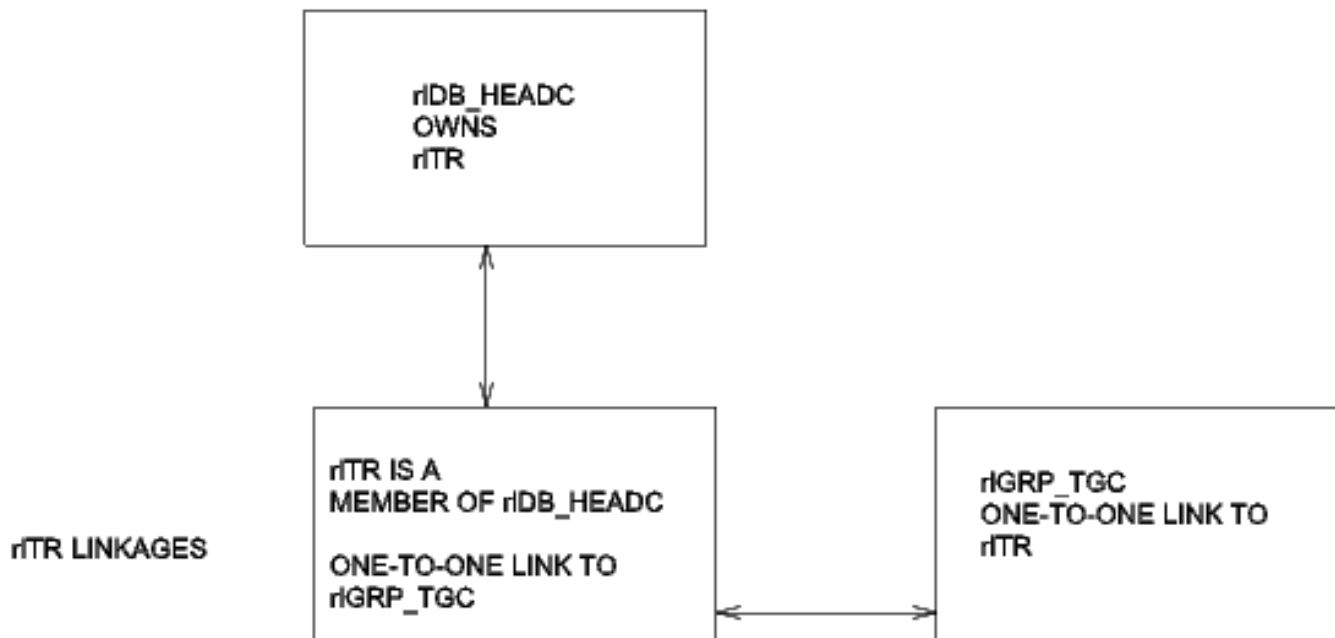


Figure 1 (NMTR-1) rLTR Linkages

3. AUDIT CHECKS

The NMTR audit uses the DRLINK driver to step through all busy and idle rLTR tuples. The DRLINK driver verifies the owner-member linkages with rIDB_HEADC and the one-to-one linkages with rGRP_TGC. After the linkages are

checked, the NMTR audit performs semantic checks of the contents of the rLTR tuple itself.

4. ERROR HANDLING

When linkage or semantic errors are detected, the rLTR tuple is recovered to the idle list of rIDB_HEADC with all fields, except the key initialized to zeros. The network managers are informed of the TR control change using the automatic trunk group control change discrete. Related audits are run to recover other data structures.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACTION

Error Code:	ACTION
Error Description:	The action control type for a busy rLTR tuple is invalid.
Possible Error Effect:	No Trunk Reservation controls are placed on this call.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLTR tuple.
<i>Bad Data</i> =	Action control type.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 2 Error Code: CATEGORY

Error Code:	CATEGORY
Error Description:	The response category in rLTR for an idle tuple is not set to no category.
Possible Error Effect:	Lost trunk reservation resources.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Category of the rLTR tuple.
<i>Good Data</i> =	NM_NOCAT.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple.

Table 3 Error Code: CAT_WRONG

Error Code:	CAT_WRONG
Error Description:	The tuple is busy; the category is out of range.
Possible Error Effect:	Incorrect percentage of calls are controlled.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Category of the rLTR tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 4 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	Inconsistent hunt type of rLTR and static rIRT_TRKG.
Possible Error Effect:	Unable to apply trunk reservations.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLTR tuple.
<i>Bad Data</i> =	Htype field of rLTR tuple.
<i>Good Data</i> =	Htype field of static rIRT_TRKG tuple.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited and rIRT_TRKG tuple.

Table 5 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	The rLTR tuple being audited is on the busy list linked to rIDB_HEADC, however, no linkage exists to rGRP_TGC. This error will fire when the 1 to 1 linkage to rGRP_TGC contains a zero, indicating no linkage present.
Possible Error Effect:	Trunk reservation control resources are lost.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Busyside.st link 1 to 1 linkage field of rLTR.
<i>Good Data</i> =	Owner portion of the linkage field of the rLTR tuple being audited.
<i>Logical Index</i> =	Key of the rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 6 Error Code: MAXRLS_ZERO

Error Code:	MAXRLS_ZERO
Error Description:	The rLTR busy tuple has no maximum reservation levels set. Both levels are set to zero.
Possible Error Effect:	No reservation levels are applied.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Res. level 1 - compare with level 2.
<i>Good Data</i> =	Res. level 2 - compare with level 1.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 7 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	A tuple could not be found in static rIRT_TRKG using the key of group number from the 1 to 1 linked rGRP_TGC.
Possible Error Effect:	Trunk reservation controls are placed on a non-existent trunk group.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Groupnum of rGRP_TGC used as key for rIRT_TRKG.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rLTR tuple.

<i>Dump</i> =	rITR tuple and rIGRP_TGC tuple.

Table 8 Error Code: MRL_MISMATCH

Error Code:	MRL_MISMATCH
Error Description:	The rITR busy tuple has both maximum reservation levels set to a value other than zero, but maximum reservation level 1 is not greater than maximum reservation level 2.
Possible Error Effect:	The wrong reservation level is applied.
Corrective Action Taken:	The rITR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITR tuple.
<i>Bad Data</i> =	Max r11 of rITR - compare with r12.
<i>Good Data</i> =	Max r12 of rITR - compare with r11.
<i>Logical Index</i> =	Key of rITR tuple.
<i>Dump</i> =	rITR tuple being audited.

Table 9 Error Code: MRL_ZERO

Error Code:	MRL_ZERO
Error Description:	The rITR automatic reservation adjustment is active, and the maximum for reservation level 1 or level 2 or both levels are zero.
Possible Error Effect:	The automatic reservation adjustment is ineffective.
Corrective Action Taken:	The rITR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITR tuple.
<i>Bad Data</i> =	Max. rel. level 1 - compare with level 2.
<i>Good Data</i> =	Max. rel. level 2 - compare with level 1.
<i>Logical Index</i> =	Key of rITR tuple.
<i>Dump</i> =	rITR tuple being audited.

Table 10 Error Code: OVERSHOOT

Error Code:	OVERSHOOT
Error Description:	The automatic reservation adjustment is active, and reservation level 2 is non-zero. Reservation level 1 minus reservation level 2 is not equal to the maximum for reservation level 1 minus maximum for reservation level 2.
Possible Error Effect:	Trunk reservation controls lose the distinction between reservation level 1 and reservation level 2.
Corrective Action Taken:	The rITR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITR tuple.
<i>Bad Data</i> =	r11 - r12 total.
<i>Good Data</i> =	max r11 - max r12 total.
<i>Logical Index</i> =	Key of rITR tuple.
<i>Dump</i> =	rITR tuple being audited.

Table 11 Error Code: OVERSHOOT2

Error Code:	OVERSHOOT2
Error Description:	The automatic reservation adjustment is active, and reservation level 2 is zero. Reservation level 1 minus reservation level 2 is greater than the maximum for reservation level 1 minus maximum for reservation level 2.
Possible Error Effect:	Trunk reservation controls lose the distinction between reservation level 1 and

	reservation level 2.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	r11 - r12 total.
<i>Good Data</i> =	max r11 - max r12 total.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 12 Error Code: RESL_ERROR

Error Code:	RESL_ERROR
Error Description:	The automatic reservation adjustment is active, but reservation level 1 is not less than or equal to the maximum for reservation level 1, and/or reservation level 2 is not less than or equal to the maximum for reservation level 2.
Possible Error Effect:	Trunk reservation reserves too many trunks.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rLTR tuple.
<i>Bad Data</i> =	Res. level 1 - compare with level 2.
<i>Good Data</i> =	Res. level 2 - compare with level 1.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 13 Error Code: RL_MISMATCH

Error Code:	RL_MISMATCH
Error Description:	The automatic reservation adjustment is inactive for this rLTR tuple, but reservation level 1 does not equal maximum for reservation level 1, or reservation level 2 does not equal maximum for reservation level 2.
Possible Error Effect:	The reservation level has moved. Trunk reservations controls are reserving incorrect numbers of trunks.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Res. level 1 - compare with level 2.
<i>Good Data</i> =	Res. level 2 - compare with level 1.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple being audited.

Table 14 Error Code: RL_WRONG

Error Code:	RL_WRONG
Error Description:	The automatic reservation adjustment is active; reservation level 1 is not greater than or equal to reservation level 2.
Possible Error Effect:	Wrong reservation level is applied, blocking incorrect percentages of calls.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	R11 of rLTR tuple.
<i>Good Data</i> =	R12 of rLTR tuple.
<i>Logical Index</i> =	Key of rLTR tuple.

<i>Dump</i> =	rLTR tuple being audited.

Table 15 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rLTR tuple being audited, should be on an idle list linked to rIDB_HEADC. This error will fire when the owner portion of the linkage field indicates this tuple should be linked to the idle list, while another portion of the linkage field indicates that it should be linked to a rIGRP TGC.
Possible Error Effect:	Incorrect trunk reservation controls.
Corrective Action Taken:	The rLTR tuple is recovered, reinitialized, and placed on the idle list of the RLTR relation. Network management administration is notified to set the 30 second automatic control change discrete.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLTR tuple.
<i>Bad Data</i> =	Return from linkage check function. A "1" indicates linkage fields are incorrect.
<i>Good Data</i> =	Owner portion of the linkage field of rLTR.
<i>Logical Index</i> =	Key of rLTR tuple.
<i>Dump</i> =	rLTR tuple.

NODEST

Software Release: 5E10 and later

1. NODE STATE (NODEST) AUDIT

This is a routine audit that is run automatically a number of times every hour.

The purpose of the audit is to detect and correct any inconsistencies in the node state data maintained by CNI in the central processor. The AUD:NODEST audit currently has only one member. Member 1 compares the node availability map with the IMS driver node state data. The node availability map indicates whether a node is an available LN or not. It is used by other software (such as the neighbor node audit) to determine LN states. Normally, it is updated simultaneously with the IMS driver data (that is, when some node's major state changes). This audit corrects any inconsistencies by copying the IMS driver data to the node availability map.

The input messages are described in the Input Messages manual (235-600-700), and the output messages are described in the Output Messages manual (235-600-750). If the audit is running automatically or the detailed option (DETL) was input, up to 10 detailed reports per segment are provided. These reports provide ring node identification (group and member), and the states maintained by the node availability map and within the IMS driver.

This audit usually takes about 2.5 minutes to run, depending upon how busy the CNI is at the time. A lot of nondeferrable work could slow it down. The audit corrects all of the errors it finds.

NSBILL-A

Software Release: 5E10 only

1. NETWORK SERVICES BILLING (NSBILL) AUDIT

The advanced system platform (ASP) family of services includes the ability to handle calls with multiple legs. Billing information for each leg is kept in a tuple of RLNS_BILL.

2. DATA STRUCTURES

The NSBILL audit references dynamic relations:

RLNS_IDX = one tuple of this relation acts as the idle list headcell for relation RLNS_BILL. Other tuples may own busy lists of rINS_BILL tuples.

RLNS_BILL = the relation under audit.

3. AUDIT CHECKS

The NSBILL audit invokes the DRINASEK audit driver.

In addition to those checks made by the driver, the NSBILL audit checks the range of the **leg_num** attribute.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An rINS_BILL tuple is not idle (rINS_BILL.nsidx_mem.st_ownk != DBNSBLIDLE + DBSFNS_IDX) and has an invalid leg_num attribute (rINS_BILL.leg_num > FCMXSTRIG).
Possible Error Effect:	Billing for the call leg associated with this tuple will not be correct.
Corrective Action Taken:	The rINS_BILL tuple is unlinked from any list it may be in, has all fields other than the key (rINS_BILL.ns_billkey) zeroed out and is then linked to the idle list (owned by the rINS_IDX tuple with an nsidxkey = DBNSBLIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINS_BILL tuple.
<i>Bad Data</i> =	The invalid call leg (rINS_BILL.leg_num).
<i>Good Data</i> =	The rINS_BILL's owner (rINS_BILL.nsidx_mem.st_ownk).
<i>Logical Index</i> =	Key of the rINS_BILL tuple (rINS_BILL.ns_billkey).
<i>Dump</i> =	The rINS_BILL tuple.

NSBILL-B

Software Release: 5E11 and later

1. NETWORK SERVICES BILLING (NSBILL) AUDIT

The advanced system platform (ASP) family of services includes the ability to handle calls with multiple legs. Billing information for each leg is kept in a tuple of RLNS_BILL.

2. DATA STRUCTURES

The NSBILL audit references dynamic relations:

RLNS_IDX = one tuple of this relation acts as the idle list headcell for relation RLNS_BILL. Other tuples may own busy lists of rINS_BILL tuples.

RLNS_BILL = the relation under audit.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CR_DATA

Error Code:	CR_DATA
Error Description:	A rINS_BILL tuple was found linked to an rINS_IDX tuple (rINS_BILL.nsidx_mem.st_ownk - DBSFNS_IDX == rINS_IDX.nsidxkey), however the one-to-one linkage with RLCRECORD was not null (DALNKNUL). This linkage (rINS_BILL.nsidx_mem.st_link) should not be in use for rINS_BILL tuples which are owned by rINS_IDX tuples.
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of ASP resources.
Corrective Action Taken:	The rINS_BILL tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rINS_BILL tuple.
<i>Bad Data =</i>	The contents of the one-to-one linkage with RLCRECORD.
<i>Good Data =</i>	DALNKNUL.
<i>Logical Index =</i>	The logical key, (rINS_BILL.ns_billkey) of the rINS_BILL tuple.
<i>Dump =</i>	The contents of the rINS_BILL tuple found in error.

Table 2 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	For an idle rINS_BILL tuple, (rINS_BILL.nsidx_mem.st_ownk - DBSFDB_HEADC == DBNSBLIDLE), the one-to-one linkage to RLCRECORD is not null (DALNKNUL). This linkage should not be in use for idle rINS_BILL tuples.
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of ASP resources.

Corrective Action Taken:	The rINS_BILL tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rINS_BILL tuple.
<i>Bad Data =</i>	The contents of the one-to-one linkage with RLCRECORD.
<i>Good Data =</i>	DALNKNUL.
<i>Logical Index =</i>	The logical key, (rINS_BILL.ns_billkey) of the rINS_BILL tuple.
<i>Dump =</i>	The contents of the rINS_BILL tuple found in error.

Table 3 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An rINS_BILL tuple is not idle (rINS_BILL.nsidx_mem.st_ownk != DBNSBLIDLE + DBSFNS_IDX) and has an invalid leg_num attribute (rINS_BILL.leg_num > FCMXSTRIG).
Possible Error Effect:	Billing for the call leg associated with this tuple will not be correct.
Corrective Action Taken:	The rINS_BILL tuple is unlinked from any list it may be in, has all fields other than the key (rINS_BILL.ns_billkey) zeroed out and is then linked to the idle list (owned by the rINS_IDX tuple with an nsidxkey = DBNSBLIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINS_BILL tuple.
<i>Bad Data =</i>	The invalid call leg (rINS_BILL.leg_num).
<i>Good Data =</i>	The rINS_BILL's owner (rINS_BILL.nsidx_mem.st_ownk).
<i>Logical Index =</i>	Key of the rINS_BILL tuple (rINS_BILL.ns_billkey).
<i>Dump =</i>	The rINS_BILL tuple.

Table 4 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	A rINS_BILL tuple was found with no owner linkage (rINS_BILL.nsidx_mem.st_ownk == DALNKNUL), and the associated linkage to RLCRECORD was not valid (rINS_BILL.nsidx_mem.st_link - DBSFCRECOR != rICRECORD.cnum) OR (rICRECORD.clink.st_link - DBSFNS_BILL != rINS_BILL.ns_billkey). All rINS_BILL tuples which are not linked to the RLDB_HEADC idle list head cell nor a rINS_IDX owner tuple, must be linked to a rICRECORD tuple.
Possible Error Effect:	Possible asserts or failures from AMA billing applications. Possible exhaustion of ASP resources.
Corrective Action Taken:	The rINS_BILL tuple is recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rINS_BILL tuple.
<i>Bad Data =</i>	The contents of the one-to-one linkage with RLNS_BILL which is located in RLCRECORD (rICRECORD.clink.st_link).
<i>Good Data =</i>	The logical key (cnum) of the associated rICRECORD tuple.
<i>Logical Index =</i>	The logical key, (rINS_BILL.ns_billkey) of the rINS_BILL tuple.
<i>Dump =</i>	The contents of the rINS_BILL tuple found in error.

NSDB

Software Release: 5E10 and later

1. NETWORK SERVICES DATA BLOCK (NSDB) AUDIT

Command channel signaling system 7 of transaction capability application protocol (CCS7TCAP).

The network services data is used by the private virtual network services feature to save necessary information throughout the life of a call. The NSDB audit verifies linkages of the RLNSDB relation, and runs in the segmented and initialization modes in the SM environment. The audit verifies linkages between RLNSDB and RLDB_HEADC (owner of RLNSDB) and one to one linkages with RLNS_IDX. All idle rINSDB tuples must be on the idle list of RLDB_HEADC. Busy tuples must be linked one to one to RLNS_IDX and not on any lists of RLDB_HEADC.

NSDB uses the DRINASEK driver and runs in segmented and initialization modes.

2. DATA STRUCTURES

- RLNSDB.
- RLDB_HEADC.
- RLNS_IDX.

Figure 1 shows the relationship of these data structures.

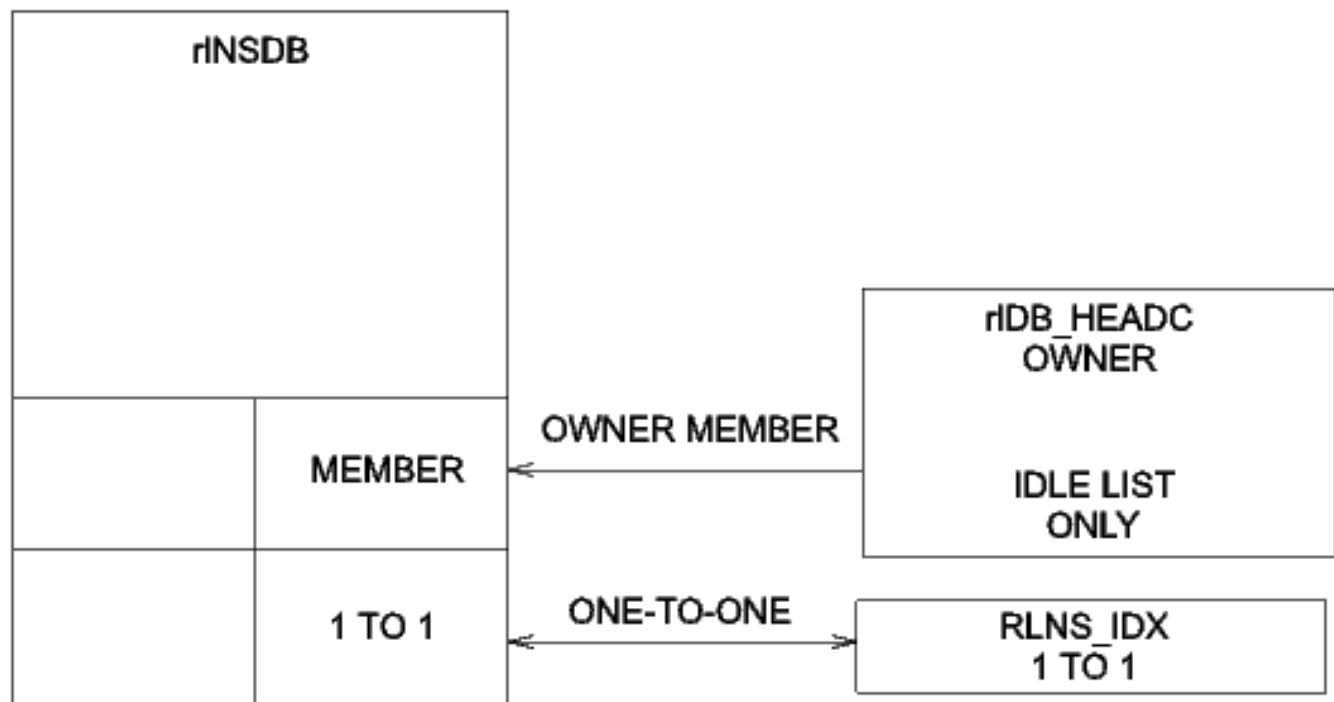


Figure 1 (NSDB-1) Data Structures of the NSDB Audit

3. AUDIT CHECKS

The NSDB audit steps through the RLNSDB relation, verifying linkages with RLDB_HEADC and RLNS_IDX.

4. ERROR HANDLING

If linkage or semantic errors are detected, the rINSDB tuple is recovered to the idle list of rIDB_HEADC after all fields are zeroed and the key is reinitialized. Related audits are run to recover other data structures.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rINSDB tuple being audited should be on a busy list (linked to rINS_IDX). This error will fire when the owner portion of the linkage field indicates this tuple should be linked to the busy list while another portion of the linkage field indicates that it should be linked to an rIDBHEADC tuple (on an idle list).
Possible Error Effect:	Advanced services platform (ASP) data may be incorrect for a call in the ringing state.
Corrective Action Taken:	The rINSDB tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINSDB tuple.
<i>Bad Data =</i>	st_link portion of linkage field; used to show linkage to an rINS_IDX tuple.
<i>Good Data =</i>	Owner portion of the linkage field of the rINSDB tuple being audited.
<i>Logical Index =</i>	Key of rINSDB tuple.
<i>Dump =</i>	rINSDB tuple being audited.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rINSDB tuple being audited is indicating that it is not owned by the idle list or a valid busy list.
Possible Error Effect:	ASP data may be incorrect for a call in the ringing state.
Corrective Action Taken:	The rINSDB tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINSDB tuple.
<i>Bad Data =</i>	st_ownk portion of linkage field; used to show the owner of the rINSDB.
<i>Good Data =</i>	Value of the idle list owner (DBNSDBIDLE).
<i>Logical Index =</i>	Key of rINSDB tuple.
<i>Dump =</i>	rINSDB tuple being audited.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rINSDB tuple being audited should be on an idle list (linked to rIDBHEADC). This error will fire when the owner portion of the linkage field indicates this tuple should be linked to the idle list while another portion of the linkage field indicates that it should be linked to a rINS_IDX (on a busy list).
Possible Error Effect:	ASP data may be incorrect for a call in the ringing state.
Corrective Action Taken:	The rINSDB tuple is recovered, reinitialized and put back on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINSDB tuple.
<i>Bad Data =</i>	Return from linkage check function 1 indicates the linkage fields are incorrect.
<i>Good Data =</i>	Owner portion of the linkage field of the rINSDB tuple being audited.
<i>Logical Index =</i>	Key of rINSDB tuple.
<i>Dump =</i>	rINSDB tuple being audited.

NSIDX

Software Release: 5E10 and later

1. NETWORK SERVICES INDEX (NSIDX) AUDIT

The advanced system platform (ASP) family of services includes the ability to handle calls with multiple legs. RLNS_IDX serves as a central place to link per-leg information.

2. DATA STRUCTURES

The NSIDX audit references dynamic relations:

RLDB_HEADC = One tuple of this relation acts as the idle list headcell for relation RLNS_IDX.

RLNS_IDX = The relation under audit.

When in use, a tuple of RLNS_IDX will be linked to a call record (that is, an entry in CRecord[]).

3. AUDIT CHECKS

The NSIDX audit invokes the DRINASEK audit driver.

In addition to those checks made by the driver, the NSIDX audit does the following:

If the tuple is idle:

ZERO_LINK = Check database linkage fields.

NOTZERO = Check linkage to CRECORD.

If the tuple is not idle:

IDLE_LINK = Check owner field.

CALLID_BAD = Check CRECORD key.

PT2PT = Check linkage to CRECORD.

OUTOFRANGE = Check query count.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CALLID_BAD

Error Code:	CALLID_BAD
Error Description:	A non-idle rINS_IDX tuple (rINS_IDX.nsidx_mem.st_ownk != DBNSIDXIDLE + DBSFDB_HEADC) has an out of range key to CRECORD in it (rINS_IDX.crd_idx ≥ CRMAX).
Possible Error Effect:	Either the tuple (and any other resources linked to it) are lost or the call it was

	supposed to be associated with will not be able to reliably access information about call legs.
Corrective Action Taken:	The rINS_IDX tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rINS_IDX.nsidxkey) and is then linked to the idle list (owned by the rIDB_HEADC tuple with an dbheadc attribute = DBNSIDXIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINS_IDX tuple.
<i>Bad Data =</i>	The invalid CRECORD index (rINS_IDX.crd_idx).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rINS_IDX tuple (rINS_IDX.nsidxkey).
<i>Dump =</i>	The rINS_IDX tuple.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An rINS_IDX tuple has an invalid owner (rINS_IDX.nsidx_mem.st_ownk != DBNSIDXIDLE + DBSFDB_HEADC or zero).
Possible Error Effect:	The rINS_IDX tuple may be lost.
Corrective Action Taken:	The rINS_IDX tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rINS_IDX.nsidxkey) and is then linked to the idle list (owned by the rIDB_HEADC tuple with an dbheadc attribute = DBNSIDXIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINS_IDX tuple.
<i>Bad Data =</i>	The invalid owner (rINS_IDX.busyidle.st_ownk).
<i>Good Data =</i>	The idle list owner value (DBNSIDXIDLE + DBSFDB_HEADC).
<i>Logical Index =</i>	Key of the rINS_IDX tuple (rINS_IDX.nsidxkey).
<i>Dump =</i>	The rINS_IDX tuple.

Table 3 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	An idle rINS_IDX tuple (rINS_IDX.nsidx_mem.st_ownk = DBNSIDXIDLE + DBSFDB_HEADC) has a non-null CRECORD key in it (rINS_IDX.crd_idx != 0).
Possible Error Effect:	If a CRECORD is actually linked to this rINS_IDX tuple, it will appear to have ASP related data associated with it which it does not have. This is likely to lead to asserts and call failure.
Corrective Action Taken:	The rINS_IDX tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rINS_IDX.nsidxkey) and is then linked to the idle list (owned by the rIDB_HEADC tuple with an dbheadc attribute = DBNSIDXIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINS_IDX tuple.
<i>Bad Data =</i>	The CRECORD key (rINS_IDX.crd_idx).
<i>Good Data =</i>	The rINS_IDX's owner (rINS_IDX.busyidle.st_ownk).
<i>Logical Index =</i>	Key of the rINS_IDX tuple (rINS_IDX.nsidxkey).
<i>Dump =</i>	The rINS_IDX tuple.

Table 4 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A non-idle rINS_IDX tuple (rINS_IDX.nsidx_mem.st_ownk != DBNSIDXIDLE + DBSFDB_HEADC) and has an out of range query count (rINS_IDX.f2qry_cnt > FCMXSTRIG).
Possible Error Effect:	Call legs may be handled incorrectly or the call may fail.
Corrective Action Taken:	The rINS_IDX tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rINS_IDX.nsidxkey) and is then linked to the idle list (owned by the rIDB_HEADC tuple with an dbheadc attribute = DBNSIDXIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rINS_IDX tuple.
<i>Bad Data =</i>	The bad query count (rINS_IDX.qry_cnt).

<i>Good Data</i> =	FCMXSTRIG.
<i>Logical Index</i> =	Key of the rINS_IDX tuple (rINS_IDX.nsidkey).
<i>Dump</i> =	The rINS_IDX tuple.

Table 5 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	A non-idle rINS_IDX tuple (rINS_IDX.nsidx_mem.st_ownk != DBNSIDXIDLE + DBSFDB_HEADC) contains a CRECORD index of a CRECORD which does not contain the rINS_IDX's key. That is, CRecord[rINS_IDX.crd_idx].ns_idx != rINS_IDX.nsidkey.
Possible Error Effect:	The rINS_IDX tuple (and any resources linked to it) may be lost.
Corrective Action Taken:	The rINS_IDX tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rINS_IDX.nsidkey) and is then linked to the idle list (owned by the rIDB_HEADC tuple with an dbheadc attribute = DBNSIDXIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the CRECORD (&CRecord[rINS_IDX.crd_idx]).
<i>Bad Data</i> =	The CR's key to RLNS_IDX (CRECORD.ns_idx).
<i>Good Data</i> =	The rINS_IDX's CRECORD key (rINS_IDX.crd_idx).
<i>Logical Index</i> =	Key of the rINS_IDX tuple (rINS_IDX.nsidkey).
<i>Dump</i> =	The rINS_IDX tuple.

Table 6 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rINS_IDX tuple (rINS_IDX.nsidx_mem.st_ownk = DBNSIDXIDLE + DBSFDB_HEADC) has a non-null linkage to some other relation.
Possible Error Effect:	Any resources linked to the idle rINS_IDX tuple cannot be used by the system for call processing.
Corrective Action Taken:	The rINS_IDX tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rINS_IDX.nsidkey) and is then linked to the idle list (owned by the rIDB_HEADC tuple with an dbheadc attribute = DBNSIDXIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINS_IDX tuple.
<i>Bad Data</i> =	The map_id of the non-null linkage.
<i>Good Data</i> =	The rINS_IDX's owner (rINS_IDX.busyidle.st_ownk).
<i>Logical Index</i> =	Key of the rINS_IDX tuple (rINS_IDX.nsidkey).
<i>Dump</i> =	The rINS_IDX tuple.

NSTN

Software Release: 5E10 and later

1. NUMBER SERVICES TERMINATION NOTIFICATION (NSTN) AUDIT

The NSTN relation saves necessary call processing information about a call throughout the life of a call. The NSTN audit verifies semantic data and linkages of the RLNS_TN relation. The audit runs segmented in the switching module (SM) environment, and also runs during a selective initialization and single process purge. The NSTN data is used for different features. Alternate structures, hold call processing information for each different feature.

Action control point for software defined networks Feature uses the alternate structure aINS_SDN. For these calls the audit checks linkages between RLDB_HEADC (owner of RLNS_TN) and one to one linkages to RLPCBLA and RLPSUUI. Busy tuples must be linked to the busy list of RLDB_HEADC and linked one to one to RLPCBLA. Additionally, busy tuples requiring user to user information must be linked to RLPSUUI.

Idle rINS_TN tuples must be on the idle list of rIDB_HEADC.

2. DATA STRUCTURES

- rINS_TN.
- rINS_SDN.
- rIPCBLA.
- rIDB_HEADC.
- rIPSUUI.

Figure 1 shows the data structures of the NSTN audit.

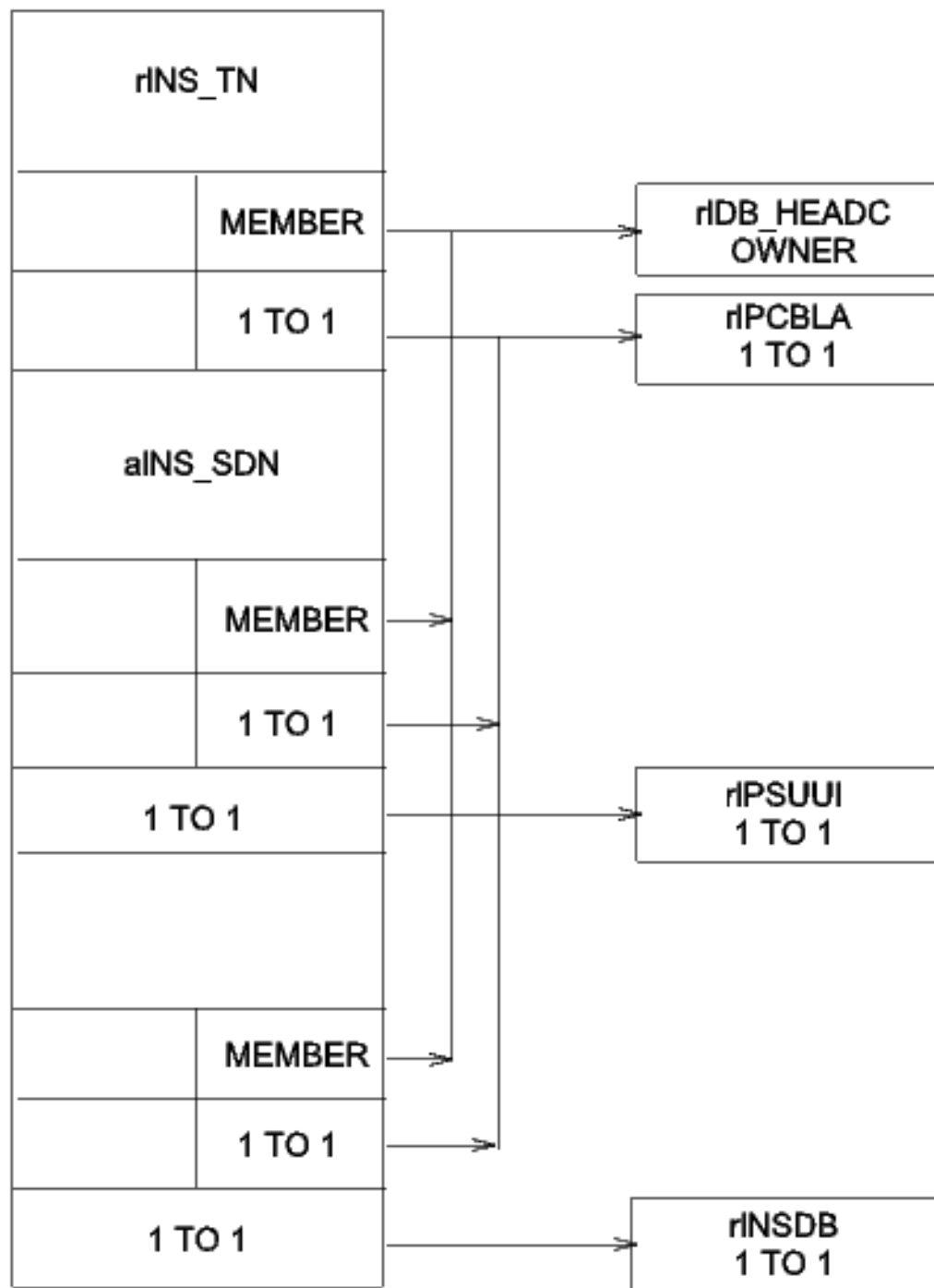


Figure 1 (NSTN-1) Data Structures of the NSTN Audit

3. AUDIT CHECKS

The NSTN audit steps through the rINS_TN relation using the DRINASEK audit driver, verifying linkages to rIDB_HEADC (the owner of rINS_TN), and one to one linkages with rPCBLA and rPSUUI.

4. ERROR HANDLING

When linkage or semantic errors are detected, the rINS_TN tuple is recovered to the idle list of rLDB_HEADC with all fields, except the key initialized to zeroes. Related audits are run to recover other data structures.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The linkage field of the rINS_TN indicates an active one to one linkage to rPCBLA. The PCBLA flag indicates this linkage is in error.
Possible Error Effect:	Call processing resources may be lost.
Corrective Action Taken:	The tuple is placed on the idle list of RLDBHEADC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The value of the linkage to the rPCBLA tuple (nstrn_ptr->busyidle.st_link - shift value of the RLPCBLA).
<i>Bad Data =</i>	The tn_info field of the associated rPCBLA tuple.
<i>Good Data =</i>	The procnum key of rPCBLA.
<i>Logical Index =</i>	The ns_id key of rINS_TN.
<i>Dump 1 =</i>	The rINS_TN tuple.
<i>Dump 2 =</i>	The rPCBLA tuple.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A one to one linkage exists from NSTN to PSUUI when the uui_info indicates no linkage should exist.
Possible Error Effect:	Lost resources for call processing
Corrective Action Taken:	The NSTN tuple is zeroed and placed on the idle list of DBHEADC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of rINS_TN.
<i>Bad Data =</i>	The physical key of the rPSUUI tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The ns_id of rINS_TN.
<i>Dump =</i>	rINS_TN tuple.

Table 3 Error Code: INV

Error Code:	INV
Error Description:	The user field of NS_TN tuple is set to an unexpected value.
Possible Error Effect:	The call is torn down.
Corrective Action Taken:	The tuple is placed on the idle list of RLDBHEADC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rINS_TN tuple.
<i>Bad Data =</i>	The user field of NS_TN tuple.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The ns_id key of the rINS_TN tuple.
<i>Dump =</i>	The rINS_TN tuple.

Table 4 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	An invalid one to one linkage exists from NS_TN to PSUUI. The uui_info flag indicates an active valid linkage should exist.
Possible Error Effect:	Lost resources for call processing
Corrective Action Taken:	The audit zaps all linkages and fields and places the tuple on the idle list of headcell.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of aINS_SDN.
<i>Bad Data</i> =	The physical key of the rIPSUUI tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The ns_id of aINS_SDN.
<i>Dump</i> =	The aINS_SDN tuple.

Table 5 Error Code: LNK2ILST

Error Code:	LNK2ILST
Error Description:	No one to one linkage exists to PCBLA when the tuple is on the busy list of headcell.
Possible Error Effect:	Lost resources for call processing
Corrective Action Taken:	The audit zaps all linkages and fields and places the tuple on the idle list of headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rINS_TN tuple.
<i>Bad Data</i> =	The st_link of the rIRS_TN tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The ns_id of rINS_TN.
<i>Dump</i> =	The rINS_TN tuple being audited.

Table 6 Error Code: NS_IDLE_USER

Error Code:	NS_IDLE_USER
Error Description:	The NS_TN tuple user field is set to DMNS_IDLE indicating the tuple should be on the idle list of RLDBHEADC. This tuple is marked as something other than idle.
Possible Error Effect:	Call processing losing resources of NS_TN tuple. The call is torn down.
Corrective Action Taken:	The tuple is placed on the idle list of RLDBHEADC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rINS_TN tuple.
<i>Bad Data</i> =	The owner linkage of the rINS_TN tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The ns_id key of rINS_TN.
<i>Dump</i> =	The rINS_TN tuple being audited.

Table 7 Error Code: STUCK

Error Code:	STUCK
Error Description:	The user field (dumped in Bad Data) indicates tuple should be idle, but the tuple is on the busy list of the headcell (DBHEADC).
Possible Error Effect:	The call is torn down.
Corrective Action Taken:	The audit zaps all linkages and fields and places the tuple on the idle list of the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rINS_TN tuple.
<i>Bad Data</i> =	The user field of rINS_TN tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The ns_id of rINS_TN.
<i>Dump</i> =	The rINS_TN tuple currently being audited.

Table 8 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The user field (dumped in Bad Data) indicates tuple should be busy, but the tuple is on the idle list of the headcell(DBHEADC).
Possible Error Effect:	Call processing data inconsistency
Corrective Action Taken:	The audit zaps all linkages and fields and places the tuple on the idle list of the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rINS_TN tuple.

<i>Bad Data</i> =	The user field of rINS_TN tuple.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The ns_id of rINS_TN.
<i>Dump</i> =	The rINS_TN tuple currently being audited.

23. O - PCC

OAOTDB

Software Release: 5E10 and later

1. OSPS ORIGINATING TREATMENT DATA BLOCK (OAOTDB) AUDIT

SS7 signaling for incoming OSPS calls from a 4ESSTM switch provides the following additional toll and assistance OSPS call types when the calls are received by AT&T's version of SS7 signaling:

- 800 access calls;
- Direct international originations (DIOR) calls;
- Inward calls; and,
- International billed number screening - release 2 (IBNS-2) calls.

The RLOAOTDB relation contains data which is used by call processing for incoming Toll and Assistance Operator Services Position System (T&A OSPS) type calls. This relation is audited by the OAOTDB audit, which resides in the switching module (SM) and which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

The OAOTDB audit checks rLOAOTDB tuples to verify that the proper number of tuples are allocated to the database, that the tuples are linked to the correct busy or idle list, and that all tuples which are marked busy are associated with live processes. The OAOTDB audit uses the following fields in the RLOAOTDB relation to perform these checks:

<i>oaotkey</i>	= The key of the rLOAOTDB tuple.
<i>quarstate</i>	= The tuple's quarantine state bit.
<i>busyidle.st_ownk</i>	= The tuple's owner linkage to the RLDB_HEADC head cell owner tuple (DBOAOTDBIDLE).
<i>busyidle.st_link</i>	= The tuple's linkage to an owning rPCBLA busy list tuple.

3. AUDIT CHECKS

The OAOTDB audit performs the following checks: If the OAOTDB audit gets entered:

ZERO_LINK	= Verify that all other linkages are unused when the rLOAOTDB tuple is linked to the RLDB_HEADC idle list.
NOPCBLA	= Verify that, when the owner linkage to RLDB_HEADC is NULL, the linkage to RLPCBLA is being used.
DEADPID	= Verify that the process for the rPCBLA tuple linked to the rLOAOTDB tuple is a live process.
INVOWNER	= Verify that the owner linkage for a rLOAOTDB tuple is either marked to DBOAOTDBIDLE or DALNKNUL.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error

code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	A rIOAOTDB tuple has been found associated with a dead process. For rIOAOTDB tuples linked to rIPCBLA tuples (rIOAOTDB.busyidle.st_link - DBSFPCBLA = rIPCBLA.procnum), the rIPCBLA tuple should be associated with a live terminal process.
Possible Error Effect:	Possible exhaustion of T&A OSPS resources.
Corrective Action Taken:	The rIOAOTDB tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOAOTDB tuple.
<i>Bad Data =</i>	The process number (rIPCBLA.procnum) found marked in the tuple (rIOAOTDB.busyidle.st_link - DBSFPCBLA).
<i>Good Data =</i>	NA
<i>Logical Index =</i>	The key of the rIOAOTDB tuple (<i>oaotkey</i>).
<i>Dump =</i>	The contents of the rIOAOTDB tuple.

Table 2 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	A rIOAOTDB tuple has been found with a corrupted owner linkage (rIOAOTDB.busyidle.st_ownk). The owner linkage must be marked with either the appropriate RLDB_HEADC key for the idle list (DBOAOTDBIDLE + DBSFDB HEA) or should be NULL (DALNKNUL).
Possible Error Effect:	Possible exhaustion of T&A OSPS resources.
Corrective Action Taken:	The rIOAOTDB tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOAOTDB tuple.
<i>Bad Data =</i>	The contents of the owner linkage (rIOAOTDB.busyidle.st_ownk).
<i>Good Data =</i>	NA
<i>Logical Index =</i>	The key of the rIOAOTDB tuple (<i>oaotkey</i>).
<i>Dump =</i>	The contents of the rIOAOTDB tuple.

Table 3 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	A rIOAOTDB tuple has been found with a NULL owner linkage (rIOAOTDB.busyidle.st_ownk = DALNKNUL) however, there is no rIPCBLA tuple linked to it (rIPCBLA.procnum = rIOAOTDB.busyidle.st_link - DBSFPCBLA). All busy rIOAOTDB tuples must have the <i>st_link</i> component of the owner linkage marked with the key of a rIPCBLA tuple when the <i>st_ownk</i> component is NULL (DALNKNUL).
Possible Error Effect:	Possible exhaustion of T&A OSPS resources.
Corrective Action Taken:	The rIOAOTDB tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOAOTDB tuple.
<i>Bad Data =</i>	The value found in the linkage to RLPCBLA (DALNKNUL).
<i>Good Data =</i>	The RLDB HEADC idle list head cell (DBOAOTDBIDLE).
<i>Logical Index =</i>	The key of the rIOAOTDB tuple (<i>oaotkey</i>).
<i>Dump =</i>	The contents of the rIOAOTDB tuple.

Table 4 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rIOAOTDB tuple has been found linked to the RLDB_HEADC idle list, however one or more other mapped linkages are not NULL. Idle rIOAOTDB tuples (rIOAOTDB.busyidle.st_ownk - DBSFDB_HEA = DBOAOTDBIDLE) should have all other mapped linkages marked NULL (DALNKNUL).
Possible Error Effect:	Possible exhaustion of T&A OSPS resources.
Corrective Action Taken:	The rIOAOTDB tuple is recovered to an idle condition (zeroed out and relinked to the RLDB HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOAOTDB tuple.
<i>Bad Data =</i>	The value in the first linkage field which was found to be non-NULL.
<i>Good Data =</i>	The contents of the linkage to the RLDB_HEADC idle list head cell (rIOAOTDB.busyidle.st_ownk - DBSFDB_HEA).
<i>Logical Index =</i>	The key of the rIOAOTDB tuple (<i>oaotkey</i>).
<i>Dump =</i>	The contents of the rIOAOTDB tuple.

OCRECORD

Software Release: 5E10 and later

1. OSPS CALL RECORD (OCRECORD) AUDIT

The operator services position system (OSPS) uses features provided by a host 5ESS® switch and dedicated peripheral equipment to support operator and attendant services. Directory, toll, and assistance services are provided by operators. These services include completion of toll calls, such as collect, person-to-person, and third-party billing. Data related to these OSPS services is stored in the RLOCRECORD relation, which is audited by the OCRECORD audit.

2. DATA STRUCTURES

The OCRECORD audit checks rLOCRECORD tuples to determine if the tuples are validly marked busy or idle. In addition, the audit verifies that all OSPS call records are associated with valid "base" call records stored in the RLCRECORD relation. The audit makes these checks based on the following data:

<i>crnum</i>	= Tuple key. There should exist a corresponding rLCRECORD tuple linked to its RLDB_HEADC busy list head cell, for all busy OSPS call records.
<i>ocrlink.st_ownk</i>	= Owner linkage field to the RLDB_HEADC idle list head cell. Idle rLOCRECORD tuples should have this field marked to DBOCRECIDLE + DBSFDB_HEA. For busy rLOCRECORD tuple, this field should always be DALNKNUL.
<i>ocrlink.st_link</i>	= One-to-one linkage with an auxiliary call record tuple in the RLACRECORD relation. It is valid for this field to not be used (DALNKNUL), but if used it must point to a valid tuple in the RLACRECORD relation, and the rLOCRECORD tuple must be busy.

The rLOCRECORD tuple and related data structures referenced by the OCRECORD audit are shown in Figure 1 .

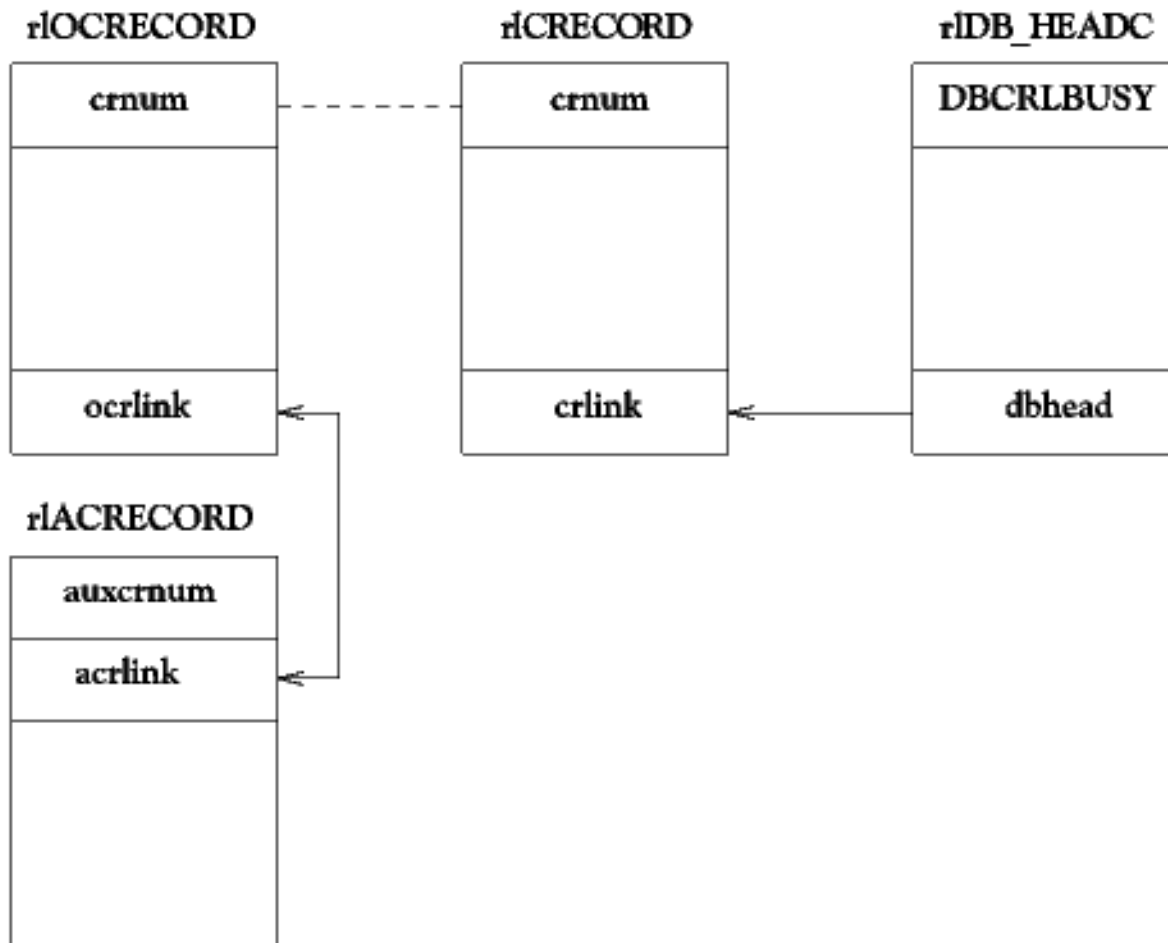


Figure 1 (OCRECORD-1) Data Structures of the OCRECORD Audit

3. AUDIT CHECKS

For each tuple in the RLOCRECORD relation, the OCRECORD audit makes the following checks: If the OCRECORD audit gets entered:

- CORRUPT_IDLE** = Check that the one-to-one linkage with RLACRECORD is not used (DALNKNUL) if the rIOCRECORD tuple is idle (rIOCRECORD.ocrlink.st_ownk == DBOCRECIDLE + DBSFDB_HEA).
- INVOWNER** = Check that busy rIOCRECORD tuples have an associated "base" call record tuple in the relation RLCRECORD, where rIOCRECORD.crnum == rLCRECORD.crnum.
- NOTBUSY** = Check that the "base" call record tuple in the RLCRECORD relation is busy (linked to its RLDB_HEADC idle list head cell) for a given busy rIOCRECORD tuple.
- BUSYIDLE** = Check that the linkage to a tuple in the RLACRECORD relation is valid, if the rIOCRECORD tuple is busy, and the RLACRECORD linkage is not idle (DALNKNUL).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not

found.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	A busy rIOfRECORD tuple has an invalid linkage to a rIACRECORD auxiliary call record tuple. For busy rIOfRECORD tuple, if the rIOfRECORD.ocrlink.st_link field is in use (not DALNKNUL), it must point to a valid tuple in the RLACRECORD relation (where rIOfRECORD.ocrlink.st_link == rIACRECORD.auxcrnum + DBSFACRECO).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The linkage to the RLACRECORD relation is idled out (marked DALNKNUL).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOfRECORD tuple.
<i>Bad Data =</i>	The logical key of the rIACRECORD tuple the audit attempted to access (rIOfRECORD.ocrlink.st_link - DBSFACRECO).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIOfRECORD tuple (rIOfRECORD.crnnum).
<i>Dump =</i>	The contents of the rIOfRECORD tuple.

Table 2 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	An idle rIOfRECORD tuple was found linked to a rIACRECORD tuple. For idle rIOfRECORD tuples (rIOfRECORD.ocrlink.st_ownk = DBOCRECIDLE + DBSFOCRECO), the rIOfRECORD.ocrlink.st_link linkage should be DALNKNUL.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rIOfRECORD tuple is recovered to an idle condition. Any RLACRECORD data inconsistencies are then recovered by the ACRECORD audit which is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOfRECORD tuple.
<i>Bad Data =</i>	The contents of the linkage to RLACRECORD (rIOfRECORD.ocrlink.st_link).
<i>Good Data =</i>	The contents of the rIOfRECORD tuple linkage to the RLDB_HEADC idle list head cell (rIOfRECORD.ocrlink.st_ownk - DBSFDB HEA).
<i>Logical Index =</i>	The key of the rIOfRECORD tuple (rIOfRECORD.crnnum).
<i>Dump =</i>	The contents of the rIOfRECORD tuple.

Table 3 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The corresponding call record (rIOfRECORD) tuple could not be accessed for a given tuple in the RLOfRECORD relation. For every tuple in the rIOfRECORD relation, there should also exist a tuple in the RLCRECORD relation where rIOfRECORD.crnnum == rIOfRECORD.crnnum.
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rIOfRECORD tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOfRECORD tuple.
<i>Bad Data =</i>	The key of the rIOfRECORD tuple that was found, if any. (rIOfRECORD.crnnum).
<i>Good Data =</i>	The key of the rIOfRECORD tuple under audit (rIOfRECORD.crnnum).
<i>Logical Index =</i>	The key of the rIOfRECORD tuple (rIOfRECORD.crnnum).
<i>Dump =</i>	The contents of the rIOfRECORD tuple.

Table 4 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The corresponding call record (rICRECORD) tuple for a busy tuple in the RLOCRECORD relation was found to be in an idle condition. All busy rIOCRECORD tuples must be associated with a busy "base" call record tuple (where rIOCRECORD.cnum == rICRECORD.cnum AND rICRECORD.crlink.st_ownk == DBCRECBUSY + DBSFECRECOR).
Possible Error Effect:	Possible exhaustion of call record billing resources, incorrect billing, and incorrect billing measurements.
Corrective Action Taken:	The rIOCRECORD tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIOCRECORD tuple.
<i>Bad Data =</i>	Contents of the head cell linkage to RLDB_HEADC (rIOCRECORD.ocrlink.st_ownk - DBSFDB_HEA).
<i>Good Data =</i>	The key of the RLDB_HEADC busy list head cell (DBCRECBUSY) for the RLCRECORD relation.
<i>Logical Index =</i>	The key of the rIOCRECORD tuple (rIOCRECORD.cnum).
<i>Dump =</i>	The contents of the rIOCRECORD tuple.

OHTIMER

Software Release: 5E12 and later

1. OWNING HEADCELL FOR TIMERS (OHTIMER) AUDIT

The RLOHTIMER relation is used to implement idle, timeout and timing lists for general purpose function timers (implemented using relation RLGPFTIMER).

2. DATA STRUCTURES

For details on the fields contained in the RLOHTIMER relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description."

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

OLOGINID

Software Release: 5E10 and later

1. OPERATOR LOGIN ID (OLOGINID) AUDIT

The OLOGINID audit runs in both the automatic call distributor management information system (ACD/MIS) and the operator services position system (OSPS). Both the ACD/MIS and OSPS features are similar. Both features are used to present queued calls to agents that are logged into the 5ESS[®] switch. In the OSPS feature, the operator would log into the 5ESS[®] switch through a basic services terminal (BST) and handle requests for operator services. In the ACD/MIS feature, the operator would log into the 5ESS[®] switch through an integrated digital services network (ISDN) position having full business and residential customer services (BRCS) features. In both features, calls are routed to available agent positions for processing. In addition, supervisors are able to monitor all agent activities. Each agent is part of a serving team that is capable of handling certain types of calls.

The RLOLOGINID relation contains tuples that define both OSPS and ACD/MIS operators in the 5ESS[®] office. Each tuple contains information about the agent's ID, the agent's serving team, and the login password as well as other fields. RLOLOGINID is a hashed relation that is keyed by the **op_id** field.

The purpose of the OLOGINID audit is to ensure that there is a one for one correspondence between the static rLOGINID tuples and the dynamic rLOGINID tuples in the communications module processor (CMP). The OLOGINID audit will also insure the integrity of the **op_id** in the RLOGINID relation. The OLOGINID audit runs in the CMP in the operational kernel process (OKP) in segmented mode. It is part of the OPAMBACD software package.

2. DATA STRUCTURES

For details on the fields contained in the RLOLOGINID relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

The OLOGINID audit uses the DRSTATIC audit driver to traverse through each tuple in the RLOLOGINID relation. For each tuple, the OLOGINID audit will make the following checks:

If the OLOGINID audit is entered:

- LOSTTUP = Check to make sure that the rLOGINID tuple has a corresponding dynamic rLOGINID tuple).
- PTR1 = Check to make sure the audit has a valid pointer to an rLOGINID tuple when updating a corrupt **op_id** field in the rLOGINID tuple).
- UNEQUAL = Check to make sure that the **op_id** fields match between the rLOGINID tuple and its corresponding rLOGINID tuple).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a static rILOGINID tuple that does not have a corresponding dynamic rILOGINID tuple.
Possible Error Effect:	This may prevent calls from being presented to the agent represented by this rILOGINID tuple.
Corrective Action Taken:	The audit will insert the missing dynamic rILOGINID tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rILOGINID tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The value of the intopid field in the rILOGINID tuple.
<i>Logical Index =</i>	The logical key of the rILOGINID tuple.
<i>Dump =</i>	The rILOGINID tuple.

Table 2 Error Code: PTR1

Error Code:	PTR1
Error Description:	The audit was not able to obtain a pointer the corresponding rILOGINID tuple when trying to update a corrupt op_id value in the rILOGINID tuple.
Possible Error Effect:	This may prevent any hunted calls from being presented to the ACD agent position represented by the rILOGINID tuple.
Corrective Action Taken:	None. The audit will only report the error and rely on the LOGININD audit to be scheduled to clean up the rILOGINID tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rILOGINID tuple.
<i>Bad Data =</i>	The value of the intopid field in the buffer used to read the rILOGINID tuple.
<i>Good Data =</i>	The value of the intopid field in the rILOGINID tuple.
<i>Logical Index =</i>	The logical key of the rILOGINID tuple.
<i>Dump 1 =</i>	The rILOGINID tuple.
<i>Dump 2 =</i>	The buffer used to read rILOGINID tuple.

Table 3 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The op_id fields do not match between the static rILOGINID tuple and its corresponding dynamic rILOGINID tuple.
Possible Error Effect:	This may result in the possible loss of rILOGINID tuple resources.
Corrective Action Taken:	The op_id field in the rILOGINID tuple is updated with the value obtained from the static rILOGINID tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rILOGINID tuple.
<i>Bad Data =</i>	The value of the op_id field in the rILOGINID tuple before correction.
<i>Good Data =</i>	The value of the op_id field in the rILOGINID tuple.
<i>Logical Index =</i>	The logical key of the rILOGINID tuple.
<i>Dump 1 =</i>	The rILOGINID tuple.
<i>Dump 2 =</i>	The rILOGINID tuple.

OPNDC

Software Release: 5E10 and later

1. OPEN DICTIONARY (OPNDC) AUDIT

The open relation dictionary, a data structure controlled by the Data Base Manager program, is closely connected with the transaction dictionary, the data base lock dictionary, and the relation queue; for additional information, refer to the TRNDC, DBLOCK and RELQ audit descriptions. The initiation of a transaction binds the data base so updates to it from sources other than the current transaction do not affect either the database or the transaction. The binding is done through a multi-level locking (MLL) mechanism, which includes relation level locking (RLL) and tuple level locking (TLL). The open relation dictionary associates the proper version of the data base under RLL with a transaction. The open dictionary audit (OPNDC) ensures that the open dictionary entries are not lost when errors occur.

The OPNDC audit is part of the data base audits subgroup. This audit runs in the operational kernel process (OKP) in the administrative module (AM), in the operating system for distributed switching-switching module (OSDS-SM), and in the communications module processor (CMP) environments in the segmented mode.

2. DATA STRUCTURES

The primary data structure of the OPNDC audit is the open dictionary array. Refer to Figure 1.

The tuples of this array are open dictionary blocks. The system allocates one open dictionary block for each relation under RLL associated with a database transaction; the relations are accessed through the transaction dictionary array (RLDB_TRNDC). In the figure, transaction 1 (T1) needs access to three relations (RL1, RL6, and RL11) under RLL; the system therefore allocates three open dictionary blocks (blocks 1, 2, and 4) to access the relations under RLL. These blocks are linked by an owner-member linkage to T1. Transaction 2 (T2) also needs access to r11 under RLL. The linkage between the T2 tuple of R1DB_TRNDC and BLOCK3 is also owner-member, even though the owner (T2) only links to a single member under RLL. Each open dictionary block is allocated with a database lock dictionary block with a one-to-one link. The database lock dictionary blocks are then put in the relation queue with a owner-member linkage. While a transaction is in progress, it may need to access other relations. In this case, the binding is through TLL. A database lock dictionary block is allocated, and put in the relation queue. No open dictionary block is allocated in the TLL case. As shown in Figure 1, Transaction 2 (T2) also needs to access r11 under TLL. Note that there is no linkage between the transaction dictionary and the data base lock dictionary. The broken owner-member linkage (#.....*) as shown in Figure 1 indicates that there may be other database lock dictionary blocks in between the queue, which are used by other transactions through TLL.

Each open dictionary block contains pointers to an access block in the access dictionary, the relation dictionary, and the dynamic relation dictionary along with its owner-member linkage to RLDB_TRNDC and the one-to-one linkage to RLDB_LOCK as shown in Figure 2.

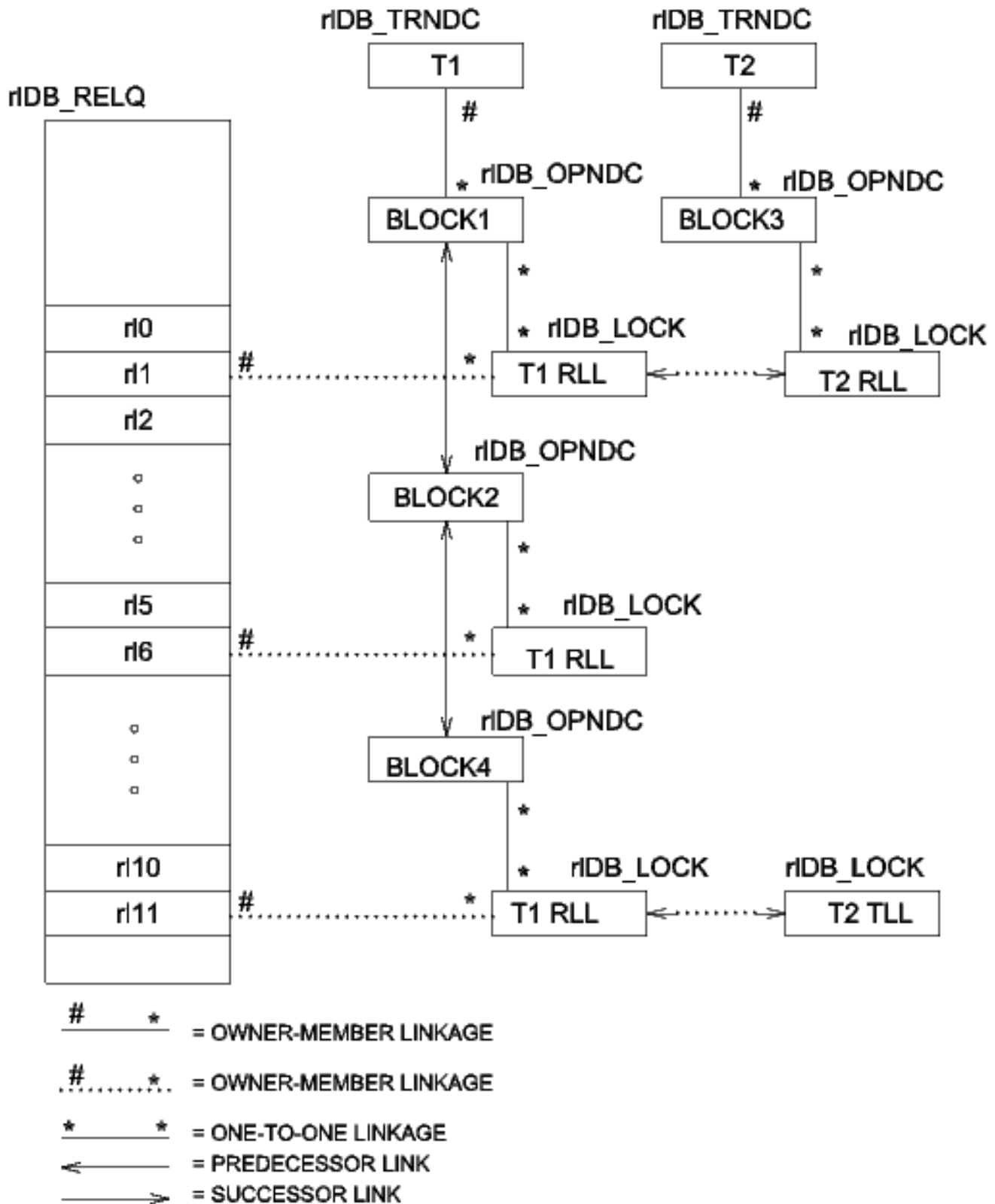


Figure 1 (OPNDC-1) Data Structures of the OPNDC Audit

rIDB_OPNDC

OPNDC ID (KEY)
TRNDC ID
RELATION ID
POINTER TO rIDB_ACCDICT
POINTER TO rIDB_RELDICT
POINTER TO rIDB_DRLDICT
MEMBER LINK TO rIDB_TRNDC
ONE-ONE LINK TO rIDB_LOCK

Figure 2 (OPNDC-2) Open Dictionary Entry**3. AUDIT CHECKS**

The audit performs syntactic and semantic checks. It calls the linkage driver (DRLINK) to traverse linkages between the open dictionary array and the database lock dictionary and transaction dictionary. The audit then performs the following semantic checks on active open dictionary entries:

- (a) If the OPNDC entry is idle, the audit ensures that the OPNDC entry is owned by the OPNDC head cell in the transaction dictionary.
- (b) The audit ensures that the transaction ID of the OPNDC entry is the same as the transaction ID in the transaction dictionary.
- (c) The audit examines the pointer to the access dictionary (RLDB_ACCDICT). If the pointer is set, the audit performs a range check. If the pointer is in range, the audit checks the relation ID field in the OPNDC entry against the relation ID fields in the access dictionary entry.
- (d) The audit checks whether the pointer to the dynamic relation dictionary (RLDB_DRLDICT) is set. If it is set, the audit performs a range check. If the pointer is in range, the audit checks the relation ID of the OPNDC entry against the relation ID in the dynamic relation dictionary.

4. ERROR HANDLING

The audit takes one of the following actions when it detects an error. If it detects a broken linkage, it aborts all the transactions and reclaims the lost resources. If the semantic check fails, only the owning transaction is aborted and the related resources are reclaimed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not

found below.

The audit produces the following error code(s).

Table 1 Error Code: ACPTRBAD

Error Code:	ACPTRBAD
Error Description:	The relation ID in the open dictionary entry does not match the one in the corresponding access dictionary or the pointer to the access dictionary is out-of-range.
Possible Error Effect:	The open dictionary entry is not accessing valid information. The transaction cannot complete correctly.
Corrective Action Taken:	The audit will idle this open dictionary and reinitialize it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_OPNDC tuple.
<i>Bad Data</i> =	Relation ID in open dictionary.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDB_OPNDC.
<i>Dump</i> =	rIDB_OPNDC tuple.

Table 2 Error Code: CRPTRBAD

Error Code:	CRPTRBAD
Error Description:	The relation ID in the rIDB_OPNDC entry does not match the one in the corresponding dynamic relation dictionary entry or the pointer to the access dictionary is out-of-range.
Possible Error Effect:	The open dictionary entry is not accessing valid information. The transaction cannot complete correctly.
Corrective Action Taken:	This audit will idle that erroneous rIDB_OPNDC entry and reinitialize it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_OPNDC tuple.
<i>Bad Data</i> =	Relation ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDB_OPNDC tuple.
<i>Dump</i> =	rIDB_OPNDC tuple.

Table 3 Error Code: OIDBAD

Error Code:	OIDBAD
Error Description:	The idle open dictionary entry is linked to the wrong idle list.
Possible Error Effect:	Other transactions cannot complete.
Corrective Action Taken:	This audit will idle the open dictionary entry and reinitialize it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_OPNDC tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Open dictionary key.
<i>Dump</i> =	The rIDB_OPNDC tuple.

Table 4 Error Code: OWNER

Error Code:	OWNER
Error Description:	The point-to/point-back linkage check for this rIDB_OPNDC tuple has failed.
Possible Error Effect:	Possible loss of resources due to rIDB_OPNDC tuples in limbo. The transaction cannot complete correctly.
Corrective Action Taken:	This audit will idle that erroneous rIDB_OPNDC entry and reinitialize it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_OPNDC tuple.
<i>Bad Data</i> =	Transaction ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDB_OPNDC tuple (transaction ID).

<i>Dump 1</i> =	rIDB_OPNDC tuple.
<i>Dump 2</i> =	Related rIDB_TRNDC tuple.
<i>Dump 3</i> =	Successor rIDB_OPNDC tuple.
<i>Dump 4</i> =	Predecessor rIDB_OPNDC tuple.

Table 5 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	The relation ID in the rIDB_OPNDC entry does not match the one in the corresponding relation dictionary or the pointer to the access dictionary is out-of-range.
Possible Error Effect:	The open dictionary entry is not accessing valid information. The transaction cannot complete correctly.
Corrective Action Taken:	The erroneous rIDB_OPNDC entry is put back onto the idle list and reinitialized by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the erroneous rIDB_OPNDC entry.
<i>Bad Data</i> =	Relation ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_OPNDC key.
<i>Dump</i> =	The rIDB_OPNDC tuple.

Table 6 Error Code: TIDBAD

Error Code:	TIDBAD
Error Description:	The owning transaction ID does not match the owner linkage.
Possible Error Effect:	The open dictionary entry is not accessing valid information. The transaction cannot complete correctly.
Corrective Action Taken:	The open dictionary entry is put back onto the idle list and reinitialized by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_OPNDC tuple.
<i>Bad Data</i> =	Transaction ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIDB_OPNDC key.
<i>Dump</i> =	The rIDB_OPNDC tuple.

OSBCB

Software Release: 5E10 and later

1. OPERATING SYSTEM MESSAGE BROADCAST AND REPLY COLLECTION CONTROL BUFFERS (OSBCB) AUDIT

The message broadcast and reply collection mechanism allows a user process in operational kernel process (OKP) environment in the administrative module (AM) or the communications module processor (CMP) to broadcast a message to all or a selected subset of the switching modules (SMs) in an office. The collection mechanism provides an interface that allows the user to process the replies at operating system for distributed switching (OSDS) interject level.

The broadcast and reply mechanism allocates a tuple of the broadcast control buffer relation (RLOSBCB) when a job is created. This tuple stores the message to be broadcast, the SM selection array, the broadcast identifier, the process identifiers of the other user, the destination in the SMs and control parameters for throttling messages. Some rLOSBCB tuples are preassigned to provide fast broadcast service. Other tuples are shared and the remainder are either in use or on an idle list.

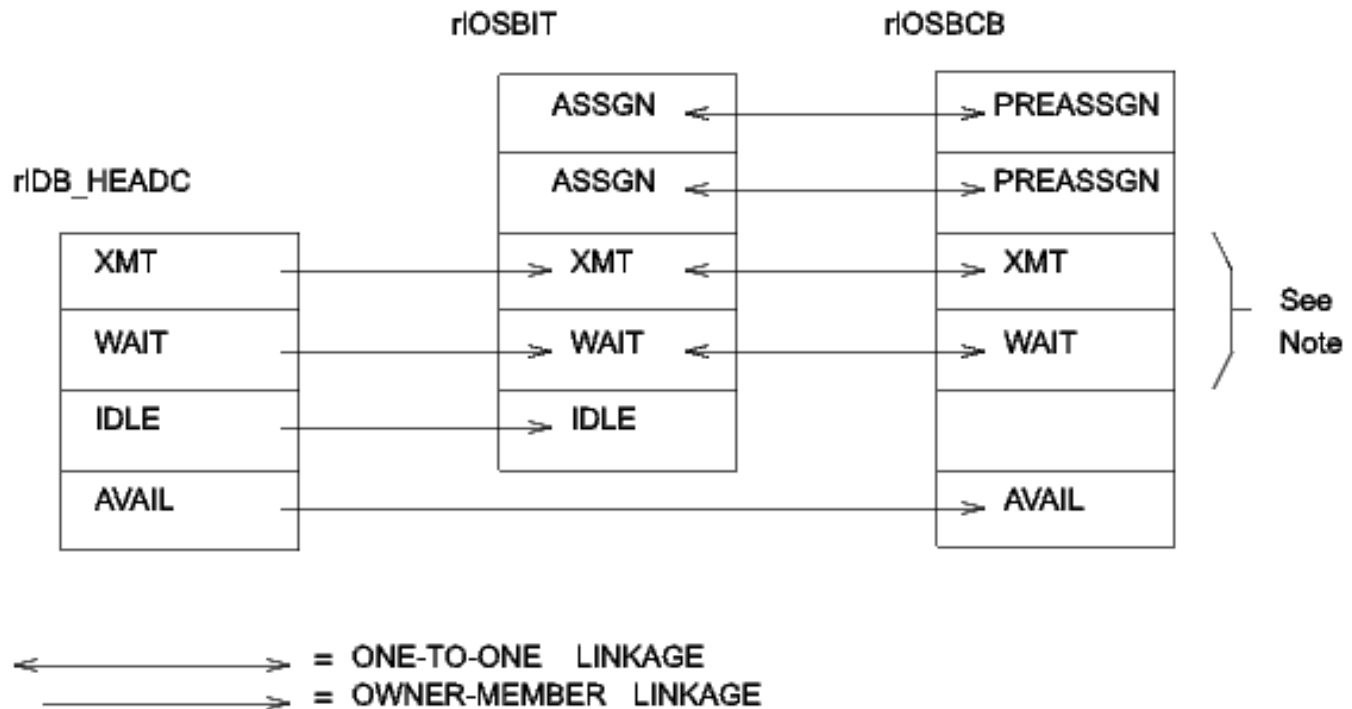
This audit runs in the OKP environment in the AM and the CMP.

2. DATA STRUCTURES

The OSBCB audit makes use of the following global data:

- Relation RLOSBCB.
- Relation RLOSBIT.
- Relation RLDB_HEADC.
- OSblastasg - Last assigned broadcast ID (maintained by operational code).

The idle list of rLOSBCB tuples is owned by an rLDB_HEADC tuple. Preassigned rLOSBCB tuples are linked one-to-one to rLOSBIT tuples. Idle shared tuples (not preassigned) are not owned. Figure 1 shows these data structures.



Note: when active rIOSBCB tuples are not on any list.

Figure 1 (OSBCB-1) Data Structures of the OSBCB Audit

3. AUDIT CHECKS

The OSBCB audit invokes the DRLINK driver to check the relations linkages. The following list gives a basic indication of the order of the checks made on each tuple after the driver has verified the linkages. For details regarding each error code see the comment blocks given.

If the tuple is owned by the broadcast available list:

WRONGLIST = Check if the tuple belongs on this list.

If the tuple is owned by a different list:

BADFLAG = Check of letlive flag.

DEADPID = Check status of a broadcast job's owning process.

NOTBUSY = Check if the tuple points to an rIOSBIT tuple.

DLOSTTUP = Check if there is not a process ID and the tuple is not preassigned.

If the tuple is preassigned:

MATELINK = Check linkage to associated rIOSBIT tuple.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	A non-idle rIOSBCB tuple's letlive flag is not set but should be. That is, rIOSBCB.from contains a dead process ID but the broadcast job should be allowed to finish (rIOSBCB.letlive should equal DBYES) because the associated tuple of RLOSBIT (key is in rIOSBCB.bcbi.link) is on the transmit list (rIOSBIT.state == OSBXMT) and the rIOSBCB.parm and rIOSBCB.timeout fields are set appropriately (rIOSBCB.parm.xmtdly = 0, parm.xmtmax = 0, parm.rsvb7 = DBNO, parm.rplynbr = 0 and rIOSBCB.timeout 0).
Possible Error Effect:	A broadcast job may be killed when it should be allowed to complete.
Corrective Action Taken:	The letlive flag is corrected (rIOSBCB.letlive = DBYES).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIOSBCB tuple in error.
<i>Bad Data</i> =	rIOSBCB tuple's letlive field.
<i>Good Data</i> =	OSblastasg.
<i>Logical Index</i> =	Key to the rIOSBCB tuple in error.
<i>Dump</i> =	The rIOSBCB tuple.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The user process ID (that is, rIOSBCB.from) stored in a non-idle rIOSBCB tuple is no longer valid and the letlive flag (rIOSBCB.letlive) is set to DBNO.
Possible Error Effect:	A broadcast job, initiated by the now dead process, may still be considered active by the broadcast system process.
Corrective Action Taken:	The rIOSBCB tuple, found to contain the dead process ID, is zeroed and returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIOSBCB tuple in error.
<i>Bad Data</i> =	Dead process ID.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to the rIOSBCB tuple in error.
<i>Dump 1</i> =	The rIOSBCB tuple.
<i>Dump 2</i> =	The associated process control block (indexed by <i>Bad Data</i>).

Table 3 Error Code: DLOSTTUP

Error Code:	DLOSTTUP
Error Description:	This tuple is not linked to any list (rIOSBCB.st_ownk), it does not have a valid process ID (rIOSBCB.bcb_ptr→from.procno) and it is not preassigned (rIOSBCB.bcb_ptr→bufx > OSblastasg).
Possible Error Effect:	If too many tuples are lost there may not be enough resources for the broadcast system process to execute a requested job.
Corrective Action Taken:	The rIOSBCB tuple which is lost is re-initialized and returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIOSBCB tuple in error (bcb_ptr).
<i>Bad Data</i> =	The rIOSBCB tuple's owner field (bcb_ptr→hcbcm.st_ownk).
<i>Good Data</i> =	The last assigned broadcast tuple (OSblastasg).
<i>Logical Index</i> =	Key to the rIOSBCB tuple in error (bcb_ptr→bufx).
<i>Dump</i> =	The rIOSBCB tuple.

Table 4 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	A preassigned broadcast ID (rIOSBCB) tuple does not point to a rIOSBIT tuple (that is, rIOSBCB.bufox is corrupt).
Possible Error Effect:	A user, expecting to have broadcast buffers ready to use, may not find them available. This could cause delays in processing that user's messages.
Corrective Action Taken:	The rIOSBCB tuple in error is zeroed and a linkage to the associated rIOSBIT tuple is established.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIOSBCB tuple in error.
<i>Bad Data</i> =	Pointer to the associated rIOSBIT tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to the rIOSBCB tuple in error.
<i>Dump</i> =	The rIOSBCB tuple.

Table 5 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An rIOSBCB tuple contains a valid process number (rIOSBCB.from) but is not linked to a corresponding rIOSBIT tuple (that is, rIOSBCB.bcbl.link = 0).
Possible Error Effect:	A broadcast job will be lost to the system.
Corrective Action Taken:	For preassigned broadcast ids, the rIOSBCB tuple is reinitialized and relinked to its associated rIOSBIT tuple. In all other cases, the tuple is reinitialized and returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIOSBCB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rIOSBCB tuple.
<i>Dump</i> =	The rIOSBCB tuple.

Table 6 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	An rIOSBCB tuple belongs to the idle list according to its owner field (rIOSBCB.hbcm.st_ownk) but appears to be preassigned (fields bufox, bcbl.link and from are consistent with this).
Possible Error Effect:	A broadcast job may be lost or deleted.
Corrective Action Taken:	The rIOSBCB tuple is reinitialized and relinked to its associated rIOSBIT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rIOSBCB tuple in error.
<i>Bad Data</i> =	rIOSBCB tuple's owner field.
<i>Good Data</i> =	Value of the correct idle list.
<i>Logical Index</i> =	Key to the rIOSBCB tuple in error.
<i>Dump</i> =	The rIOSBCB tuple.

OSBIT

Software Release: 5E10 and later

1. OPERATING SYSTEM MESSAGE BROADCAST AND REPLY COLLECTION IDENTIFIER TABLE (OSBIT) AUDIT

The message broadcast and reply collection mechanism allows a user process in the operational kernel process (OKP) environment in the administrative module (AM) or the communications module processor (CMP) to broadcast a message to all or a selected subset of the switching modules (SMs) in an office. The collection mechanism, which is implemented by setting bits in the operating system broadcast identifier table (RLOSBIT) relation, provides an interface allowing the user to process replies at the operating system for distributed switching (OSDS) interject level. Every broadcast job in the system occupies a tuple of rLOSBIT during the lifetime of the job.

This audit runs in the OKP environment in the AM and the CMP.

2. DATA STRUCTURES

The OSBIT audit makes use of the following global data:

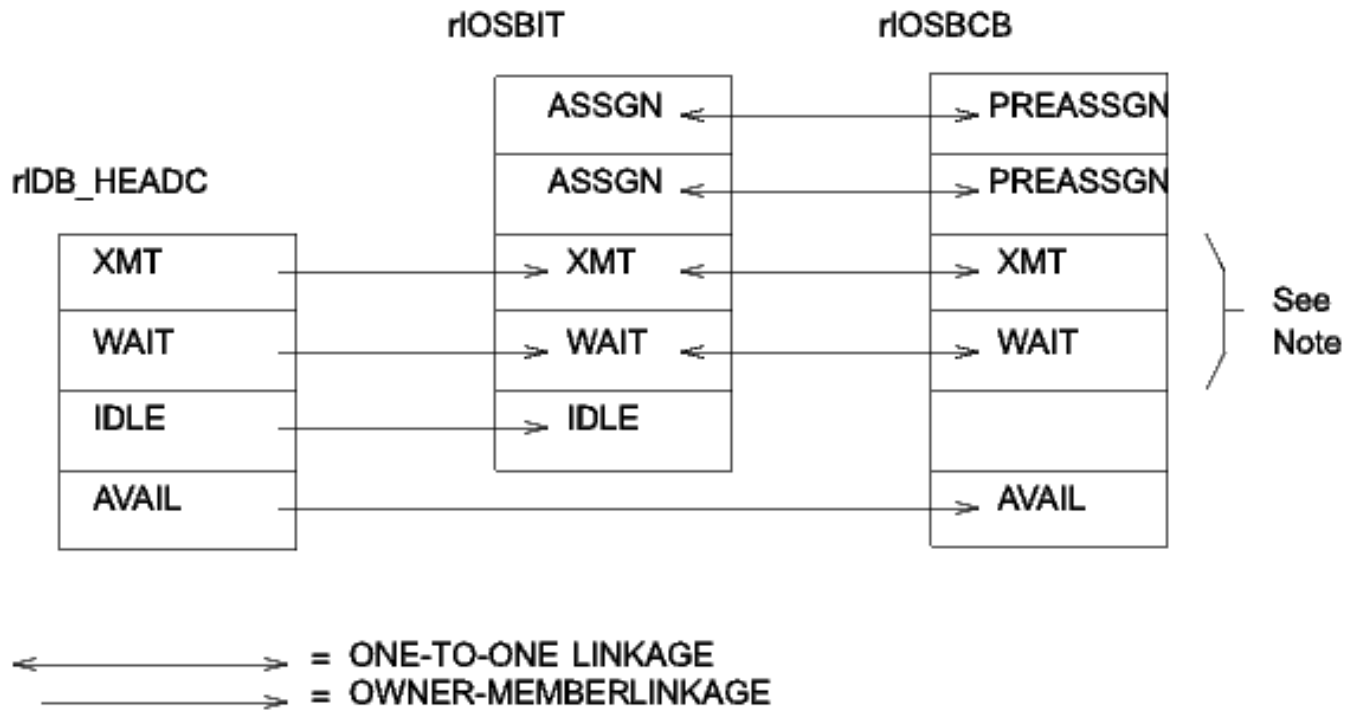
- Relation RLOSBIT.
- Relation RLOSBCB.
- Relation RLDB_HEADC.
- OSblastasg - Last assigned broadcast id (maintained by operational code).

The idle list of rLOSBIT tuples is owned by an rLDB_HEADC tuple. The following table shows the four states rLOSBIT tuples may be in:

Idle	• Not associated with a job. • Available for new job requests. • Linked to the idle list.
Preassigned inactive	Accessed by key only. • Not linked to any list. • Assigned to a certain user and not on any list. • Unused preassigned tuples are considered assigned.
Transmit	• In use by an active job in the transmitting phase.
Wait	• In use by a job that has finished transmitting and is waiting for replies to complete.

Preassigned tuples are used only for specific applications to provide fast broadcast service. A one-to-one linkage always exists between a preassigned rLOSBIT tuple and its associated rLOSBCB tuple.

Users initiating a broadcast job choose either an idle broadcast ID (BID) from the idle list or a BID preassigned to the application. Figure 1 shows the data structures of the OSBIT audit.



Note: when active, rIOSBCB tuples are not on any list.

Figure 1 (OSBIT-1) Data Structures of the OSBIT Audit

3. AUDIT CHECKS

The OSBIT audit invokes the DRLINK driver to check the relation's linkages. The following list gives a basic indication of the order of the checks made on each tuple after the driver has verified the linkages. For details regarding each error code see the comment block given.

If the tuple is in an idle state:

WRONGLIST = Check if the tuple belongs on the idle list.

STATE = check if the tuple is owned by the idle list.

If the tuple is in an assigned state:

WRONGLIST = Check if the tuple should be in an assigned state.

If the tuple is in a transmit state:

WRONGLIST = Check if the tuple is owned by the transmit list.

If the tuple is in a waiting state:

WRONGLIST = Check if the tuple is owned by the wait list.

Otherwise:

STATE_VAR = Check the tuple's state field.

If the tuple is assigned:

MATELINK = Check rIOSBIT tuple's link to rIOSBCB tuple.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	A preassigned rIOSBIT tuple points to an rIOSBCB tuple that does not point back (that is, rIOSBIT.bid and rIOSBCB.buflx do not agree).
Possible Error Effect:	A user, expecting to have broadcast buffers ready to use, may not find them available. This causes delays in processing the user's messages.
Corrective Action Taken:	The rIOSBIT tuple in error is zeroed and a linkage to an rIOSBCB tuple is established.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIOSBIT tuple in error.
<i>Bad Data</i> =	Address of the associated rIOSBCB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to the rIOSBIT tuple in error.
<i>Dump</i> =	rIOSBIT tuple.

Table 2 Error Code: STATE

Error Code:	STATE
Error Description:	An rIOSBIT tuple was in an idle state (rIOSBIT.state = OSBIDLE) but the tuple's owner (rIOSBIT.st_ownk) is not the idle list.
Possible Error Effect:	If operational code tries to process this tuple they will not be able to complete the broadcast job since the tuple is probably idle. This may also cause asserts.
Corrective Action Taken:	The rIOSBIT tuple is zeroed. If it is not preassigned it will be linked to the idle list. If it is preassigned, the linkage with its associated rIOSBCB tuple will be reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIOSBIT tuple in error.
<i>Bad Data</i> =	The rIOSBIT tuple's owner field.
<i>Good Data</i> =	The correct value for an idle rIOSBIT tuple's owner field (DBBCIDLE).
<i>Logical Index</i> =	Key to the rIOSBIT tuple in error.
<i>Dump</i> =	The rIOSBIT tuple.

Table 3 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	An rIOSBIT tuple is in an invalid state (rIOSBIT.state).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIOSBIT tuple is zeroed. If it is not preassigned it will be linked to the idle list. If it is preassigned, the linkage with its associated rIOSBCB tuple will be

	reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIOSBIT tuple in error.
<i>Bad Data</i> =	The rIOSBIT tuple's state field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to the rIOSBIT tuple in error.
<i>Dump</i> =	The rIOSBIT tuple.

Table 4 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	If the rIOSBIT's state (rIOSBIT.state, given in "Bad Data") is idle (OSBIDLE) then it's broadcast ID (rIOSBIT.bid) is $\leq$ the last assigned BID (OSblastasg). This means that that the tuple was preassigned and should not be in the idle state. If the tuple's state in an assigned state (rIOSBIT.state = OSBASGN) then its broadcast ID is $>$ the last assigned BID. This means that the tuple was not preassigned and should not be in the assigned state. In the remaining two cases (rIOSBIT.state = OSBXMT or OSBWAIT) the state and owner (rIOSBIT.st_ownk.owner) are inconsistent.
Possible Error Effect:	Varies according to conditions. If the tuple is truly on the wrong list (it doesn't just have a corrupted state field), operational code will not be able to complete broadcast jobs involving this tuple. Asserts may also be fired.
Corrective Action Taken:	The rIOSBIT tuple is zeroed. If it is not preassigned it will be linked to the idle list. If it is preassigned, the linkage with its associated rIOSBCB tuple will be reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIOSBIT tuple in error.
<i>Bad Data</i> =	The rIOSBIT tuple's state field.
<i>Good Data</i> =	Last assigned broadcast id (OSblastasg).
<i>Logical Index</i> =	Key to the rIOSBIT tuple in error.
<i>Dump</i> =	The rIOSBIT tuple.

PAGBUF

Software Release: 5E10 and later

1. PAGING BUFFER (PAGBUF) AUDIT

Diagnostics are run in the switching modules (SMs) as much as possible because they are closest to the hardware being tested. However, SM diagnostic programs do not reside in the SMs. The programs reside in the administrative module (AM) and are paged to SM memory when needed. The SM paging buffer consists of contiguous blocks of memory called **segments**. Paging buffer segments are allocated dynamically to paged programs. The paging supervisor loads the diagnostic programs into the paging buffer and the diagnostic supervisor executes them.

If a paged program fails or aborts, the paging supervisor will not deallocate the paging buffer memory used to store the program. The memory is therefore lost to the system. To prevent lost memory, the paging buffer audit (PAGBUF) verifies that each paging buffer segment that stores a paged program (or a portion of a program) is associated with a valid process. The audit also ensures the internal consistency of the data structures used by the diagnostic supervisor.

This audit is part of the Diagnostic Supervisor audits subgroup. It runs in the OSDS-M environment in the segmented mode. It is invoked periodically by the routine audit process or manually by means of an input message.

2. DATA STRUCTURES

The data structures checked by the PAGBUF audit are the paging buffer status table (DGpbst) and the request table (DGrt). The DGpbst contains status information about the paging buffer segments. The DGrt contains information about the users requesting the paged programs. Each entry in the request table has an index to the first entry in the paging buffer status table that contains its paged program. A one-to-one correspondence exists between entries in the paging buffer status table and the paging buffer segments. Figure 1 illustrates an entry in the paging buffer status table. Figure 2 illustrates an entry in the request table. The PAGBUF audit performs the following checks on the paging buffer status table (DGpbst) and the request table (DGrt). Figure 3 illustrates the correspondence between DGpbst, DGrt, and the paging buffer.

DGpbst

PAGED PRODUCT ID
NUMBER OF USERS (usecount)
CHECK COUNT (chkcount)
.
.
.
STATUS
NUMBER OF SEGMENTS (nseg)
SEGMENT ORDINAL (segn)

Figure 1 (PAGBUF-1) Paging Buffer Status Table

DGr1

PROCESS ID {pid}
.
.
.
PAGING BUFFER STATUS TABLE INDEX (pbsti)

Figure 2 (PAGBUF-2) Request Table

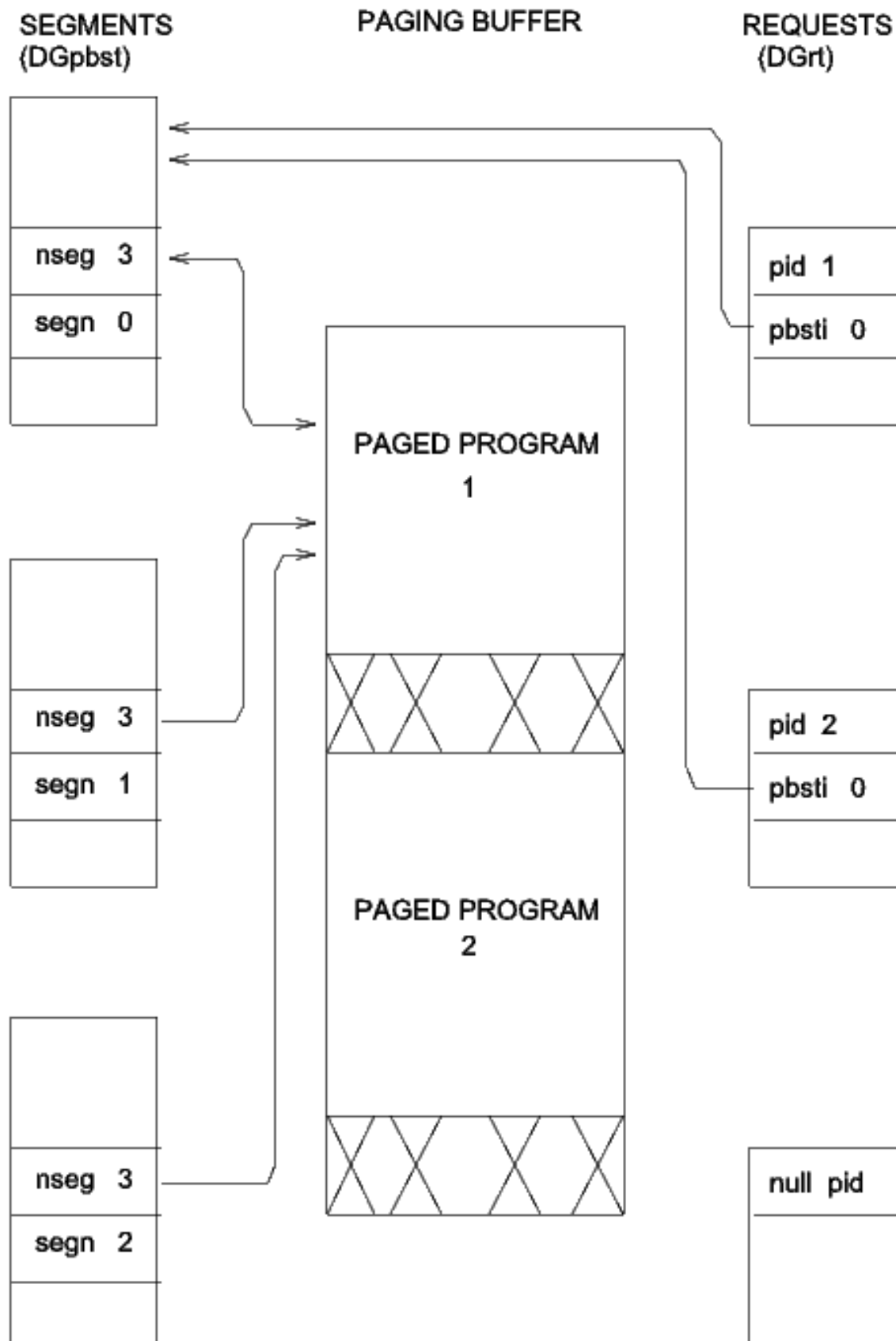


Figure 3 (PAGBUF-3) Paging Buffer Audit**3. AUDIT CHECKS**

- (a) The audit steps through the paging buffer status table and checks the following fields:
- (1) The audit verifies that the number of segments field (nseg) is consistent with the number of segments in the paging buffer. The number must be greater than zero and less than or equal to the number of segments in the table.
 - (2) The audit verifies that the segment ordinal field (segn) contains consecutive numbers starting at zero.
 - (3) The audit checks that the data in the status field is within the designated range.
- (b) The audit steps through the request table and checks the following fields:
- (1) The audit verifies that the process ID field (pid) contains a valid process name.
 - (2) The audit checks that the data in the paging buffer status table index field (pbsti) is within the designated range. As the audit checks each pbsti field, it also increments the check count (chkcount) field in the paging buffer status table.
- (c) The audit makes a final pass through the paging buffer status table. It compares the check count field (chkcount) and the number of users field (usecount) to find inconsistencies.
- (1) If the fields are equal, the audit zeroes the chkcount field to prepare it for the next time the audit executes.
 - (2) If the fields are not equal, the audit reports the error and starts recovery action.

4. ERROR HANDLING

All errors are handled by reinitializing both the status table and the request table to their original values and purging the paging supervisor.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DGMP

Error Code:	DGMP
Error Description:	A process that has an active paging request has disappeared.
Possible Error Effect:	Paging buffer memory may be unnecessarily tied up.
Corrective Action Taken:	A MGIN_TPGNE message is sent to IM-PAGS so it will deallocate all requests for the missing process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of DGrt[rt i].
<i>Bad Data =</i>	Process ID from the request table.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rt i.
<i>Dump 1 =</i>	Request table entry with the missing OSPID.
<i>Dump 2 =</i>	Associated paging buffer status table entry.

Table 2 Error Code: DGNL

Error Code:	DGNL
Error Description:	There is an invalid null entry in the Paging Buffer Status Table.
Possible Error Effect:	The paging buffer status table is in error. The paging supervisor may not know what is in the paging buffer.
Corrective Action Taken:	The SM paging supervisor is single process purged and the Request table and the paging buffer status tables are reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of DGpbst[pbst_i].
<i>Bad Data =</i>	A paging buffer status table entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	pbst_i.
<i>Dump =</i>	DGpbst[pbst_i] = The paging buffer status entry in error.

Table 3 Error Code: DGRN

Error Code:	DGRN
Error Description:	Invalid data in a request table entry.
Possible Error Effect:	The request tables and paging buffer status tables are in error. The paging supervisor may not know what is in the paging buffer or which processes currently have paged products.
Corrective Action Taken:	The SM paging supervisor is single process purged and the request table and the paging buffer status tables are reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of DGrt[rt_i].
<i>Bad Data =</i>	Time or to_warn or pbsti.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rt_i.
<i>Dump =</i>	DGrt[rt_i] = The request table entry in error.

Table 4 Error Code: DGRT

Error Code:	DGRT
Error Description:	Invalid data in a request table entry.
Possible Error Effect:	The request tables and paging buffer status tables are in error. The paging supervisor may not know what is in the paging buffer or which processes currently have paged products.
Corrective Action Taken:	The SM paging supervisor is single process purged and the request table and the paging buffer status tables are reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of DGrt[rt_i].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rt_i.
<i>Dump =</i>	DGrt[rt_i] = The request Table entry in error.

Table 5 Error Code: DGSN

Error Code:	DGSN
Error Description:	Segments numbers are invalid in the paging buffer status table.
Possible Error Effect:	The paging buffer status table is in error. The paging supervisor may not know what is in the paging buffer.
Corrective Action Taken:	The SM paging supervisor is single process purged and the request table and the paging buffer status tables are reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of DGpbst[pbst_i].
<i>Bad Data =</i>	nseq or segn in the paging buffer status table.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	pbst_i.

<i>Dump 1</i> =	DGpbst[pbst_i] = The paging buffer status entry in error.
<i>Dump 2</i> =	DGpbst[pbst_ibg] = The entry for the beginning of the block that contains the error.

Table 6 Error Code: DGST

Error Code:	DGST
Error Description:	Bad paging buffer status table status.
Possible Error Effect:	The paging buffer status table is in error. The paging supervisor may not know what is in the paging buffer.
Corrective Action Taken:	The SM paging supervisor is single process purged and the request table and the paging buffer status tables are reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of DGpbst[pbst_ibg].
<i>Bad Data</i> =	DGpbst[pbst_ibg].status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	pbst_ibg.
<i>Dump</i> =	DGpbst[pbst_ibg] = The paging buffer status entry in error.

Table 7 Error Code: DGUC

Error Code:	DGUC
Error Description:	The use count in the paging buffer status table does not agree with the number of active requests that are using the product.
Possible Error Effect:	The request tables and paging buffer status tables are in error. The paging supervisor may not know what is in the paging buffer or which processes currently have paged products.
Corrective Action Taken:	The SM paging supervisor is single process purged and the request table and the paging buffer status tables are reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of DGpbst[pbst_i].
<i>Bad Data</i> =	usecount.
<i>Good Data</i> =	chkcount.
<i>Logical Index</i> =	pbst_i = Paging buffer status table entry where the error was found.
<i>Dump 1</i> =	DGpbst[pbst_i] = The paging buffer status table entry.
<i>Dump 2</i> =	DGr = The entire request table.

PARKLOT

Software Release: 5E10 and later

1. CALL PARK TABLE ENTRY (PARKLOT) AUDIT

Call park is a custom ISDN and analog centrex feature with characteristics similar to call hold and call pickup. Call park is useful when a party receiving a call wishes to physically move to another location for convenience. In this way, a party may park a call, walk to another office or another part of the factory, then pick up the parked call. A party also may park a call, make a succession of calls, and then pick up the parked call later. The call park service includes basic call park, directed call park, retrieve, answer back, ring-back on parked calls exceeding a designated time limit, and 50B and ISDN attendant (ISAT) park operation.

2. DATA STRUCTURES

The RLPARK_LOT dynamic relation serves as a table for parked calls. This relation contains a tuple entry for each DN on a given SM that has a call parked against it. RLPARK_LOT is a hashed relation. Tuple entries are created as calls get parked, and are subsequently deleted as calls get unparked. Each tuple contains data related to the parked DN. This data includes *locdn* (key) which is the parked DN, *park_pid* which is the call park terminal process, *dn_port* which is the port of the DN the call is parked against, and *touch* which is the keep alive touch bit.

Call park processes and RLPARK_LOT table entries need not reside on the same SM. For instances where the RLPARK_LOT table entry and the call park process reside on the *same* SM, there exist the following implied linkages:

- Between the *pk_dn* field represented in the RLMDb PARK_DATA feature data block and the *locdn* (key) field in RLPARK_LOT;
- Between *pk_mod* in the RLMDb PARK_DATA feature data block and *park_pid.pcrd* in RLPARK_LOT; and,
- Between *park_pid.procno* in RLPARK_LOT and *rIPCBLA.procnum* in the associated rIPCBLA tuple.

All of these implied linkages are audited by the PARKLOT audit only for cases where the call park terminal process and RLPARK_LOT table entries are local to the same SM. For cases where this data resides on different SMs, these linkages are audited by call park keep alive.

The *parked* bit in RLPcBLA indicates when the rIPCBLA tuple is being used for a call park process, and is always marked DBYES for call park feature models. This bit is audited by the PARKLOT audit when the call park terminal process and the RLPARK_LOT table entry reside on the same SM. The integrity of the bit is maintained by call park keep alive, for processes and table entries residing on different SMs.

A call park data structure scenario is presented in Figure 1 .

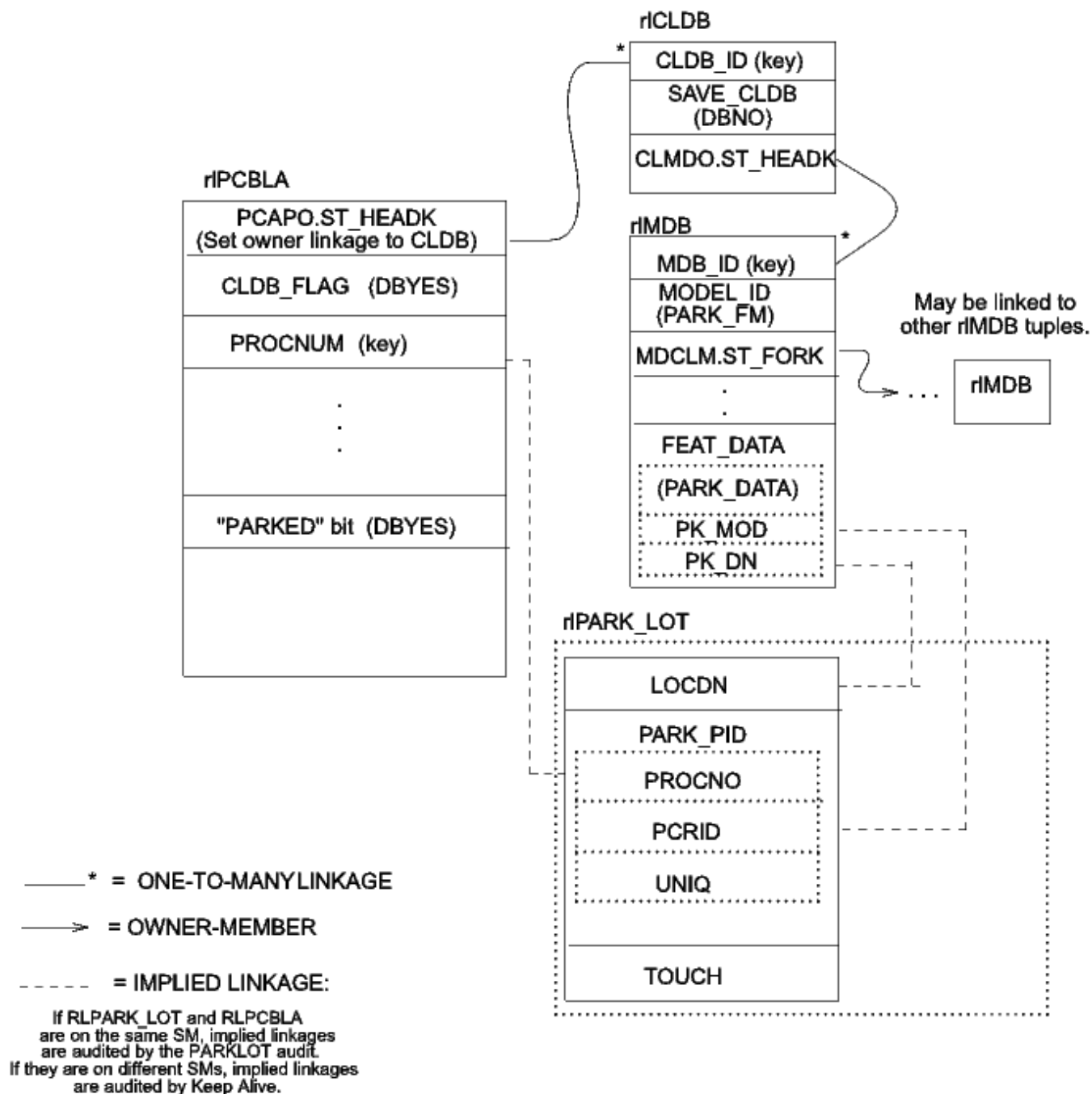


Figure 1 (PARKLOT-1) RLPARKLOT Linkages

3. AUDIT CHECKS

The PARKLOT audit uses the DRSTATIC audit driver. For each rPARK_LOT tuple found in the relation, the audit makes the following checks:

If the PARKLOT audit is entered:

OUTOFRANGE = Check that the SM marked in *park_pid* is valid.

If the RLPARK_LOT table entry and the call park process reside in the same SM:

PID = Check that *park_pid* in RLPARK_LOT is a live process 2D.

PMDBUSE = Check that an associated rIMDB tuple exists.

KEYFUNC = Check *locdn* (key) in RLPARK_LOT against *pk_dn* in RLMDDB.

PCRID = Check *park_pid.pcrid* against *pk_mod* in RLMDDB.

NOPCBLA = Check that an associated rIPCBLA tuple exists.

PSTATUS = Check that *parked* bit in RLPCBLA is DBYES.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: KEYFUNC

Error Code:	KEYFUNC
Error Description:	A RLPARK_LOT table entry is not linked to an rIMDB park feature model tuple. If the call park terminal process and the RLPARK_LOT table entry reside on the same SM (rIPARK_LOT.park_pid.pcrid = GI_pcrid), then there should also exist a rIMDB tuple used as the park feature model (rIMDB.model_id = PARK_FM) which is part of the list of rIMDB tuples linked off of an owning rICLDB tuple attached to the process (RLPCBLA), where (rIPCBLA.procnum = rIPARK_LOT.park_pid.procno).
Possible Error Effect:	No call model exists for the "parked" DN as represented by rIPARK_LOT.locdn. Thus, calls cannot be parked or unparked against this DN.
Corrective Action Taken:	The rIPARK_LOT tuple is deleted. If accessible, the rICLDB, and rIPCBLA tuples are recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical address of the rIPARK_LOT table entry.
<i>Bad Data</i> =	GLFAIL.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1</i> =	The contents of the RLPARK_LOT table entry.
<i>Dump 2</i> =	The contents of the <i>locdn</i> structure which serves as the logical key of the RLPARK_LOT table entry.
<i>Dump 3</i> =	The contents of the PARK_DATA <i>feat_data</i> union in the related rIMDB tuple.

Table 2 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	The rIPCBLA tuple associated with the RLPARK_LOT table entry under audit could not be accessed. If the call park terminal process and the RLPARK_LOT table entry reside on the same SM (rIPARK_LOT.park_pid.pcrid = GI_pcrid), then there should exist a corresponding rIPCBLA tuple for the related call park terminal process, where ((key)rIPCBLA.procnum = rIPARK_LOT.park_pid.procno).
Possible Error Effect:	The terminal process for the parked call cannot be accessed. Calls associated with the DN marked in the RLPARK_LOT table entry cannot be parked or unparked.

Corrective Action Taken:	The rIPARK_LOT tuple is deleted. If accessible, the rIMDB, and rICLDB tuples are recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The logical address of the rIPARK_LOT table entry.
<i>Bad Data =</i>	The process number <i>procnum</i> (key) of the rIPCBLA tuple that was found.
<i>Good Data =</i>	The process number <i>park_pid.procno</i> from the RLPARK_LOT table entry that was expected.
<i>Logical Index =</i>	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1 =</i>	The contents of the RLPARK_LOT table entry.

Table 3 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The SM marked in the <i>park_pid</i> field in a RLPARK_LOT table entry (rIPARK_LOT.park_pid.pcid) is either out of range or not in operation. This field should be in the legal range of SM values (between GL1STSM and GLLASTSM inclusive) and should reference an SM that is equipped and in operation.
Possible Error Effect:	Calls cannot be parked or unparked against the DN marked in the RLPARK_LOT table entry (rIPARK_LOT.locdn).
Corrective Action Taken:	The rIPARK_LOT tuple is deleted. If the Call Park terminal process is local to this SM, then the related rIMDB, rICLDB, and rIPCBLA tuples are recovered to an idle state, and the call park terminal process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The logical address of the rIPARK_LOT table entry.
<i>Bad Data =</i>	The contents of the <i>pcrid</i> field in the <i>park_pid</i> structure in the RLPARK_LOT table entry.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1 =</i>	The contents of the RLPARK_LOT table entry.
<i>Dump 2 =</i>	The contents of the <i>locdn</i> structure which serves as the logical key of the RLPARK_LOT table entry.

Table 4 Error Code: PCRID

Error Code:	PCRID
Error Description:	The SM number marked in a RLPARK_LOT table entry did not match the SM number marked in the PARK_DATA feature data union in an associated rIMDB tuple. If the call park terminal process and the RLPARK_LOT table entry reside on the same SM (rIPARK_LOT.park_pid.pcid = GI_pcid), then the SM number marked in the RLPARK_LOT table entry should match the SM number marked in the associated rIMDB tuple's PARK_DATA feature data union (rIPARK_LOT.park_pid.pcid = (PARK_DATA)rIMDB.feats_data.pk_mod) where ((PARK_DATA)rIMDB.feats_data.pk_dn = (key)rIPARK_LOT.locdn).
Possible Error Effect:	The SM where the call is parked cannot be accessed. Calls for the DN marked in the RLPARK_LOT table entry cannot be parked or unparked.
Corrective Action Taken:	The rIPARK_LOT tuple is deleted. If accessible, the rIMDB, rICLDB, and rIPCBLA tuples are recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The logical address of the rIPARK_LOT table entry.
<i>Bad Data =</i>	The SM number found in the rIPARK_LOT table entry (rIPARK_LOT.park_pid.pcid).
<i>Good Data =</i>	The SM number found in the associated rIMDB tuple ((PARK_DATA)rIMDB.feats_data.pk_mod).
<i>Logical Index =</i>	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1 =</i>	The contents of the RLPARK_LOT table entry.
<i>Dump 2 =</i>	The contents of the <i>locdn</i> structure which serves as the logical key of the RLPARK_LOT table entry.

Table 5 Error Code: PID

Error Code:	PID
Error Description:	The <i>park_pid</i> structure in the RLPARK_LOT table entry under audit did not contain the process ID for a live process. If the call park terminal process and the RLPARK_LOT table entry reside on the same SM (rIPARK_LOT.park_pid.pcrd = GI_pcrd), then the <i>park_pid</i> structure in the RLPARK_LOT table entry should contain the process ID for a live process in this SM.
Possible Error Effect:	Calls cannot be parked or unparked against the DN marked in the RLPARK_LOT table entry (rIPARK_LOT.locdn).
Corrective Action Taken:	The rIPARK_LOT tuple is deleted. If accessible, the related rIMDB, rICLDB, and rIPCBLA tuples are recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical address of the rIPARK_LOT table entry.
<i>Bad Data</i> =	The contents of the of the <i>park_pid</i> structure in the RLPARK_LOT table entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1</i> =	The contents of the RLPARK_LOT table entry.
<i>Dump 2</i> =	The contents of the <i>locdn</i> structure which serves as the logical key of the RLPARK_LOT table entry.

Table 6 Error Code: PMDBUSE

Error Code:	PMDBUSE
Error Description:	A RLPARK_LOT table entry is not linked to an rIMDB Park Feature Model tuple. If the call park terminal process and the RLPARK_LOT table entry reside on the same SM (rIPARK_LOT.park_pid.pcrd = GI_pcrd), then there should also exist a rIMDB tuple used as the park feature model (rIMDB.model_id = PARK_FM) which is part of the list of rIMDB tuples linked from an owning rICLDB tuple attached to the process (RLPCBLA), where (rIPCBLA.procnum = rIPARK_LOT.park_pid.procno).
Possible Error Effect:	No call model exists for the "parked" DN as represented by rIPARK_LOT.locdn. Thus, calls cannot be parked or unparked against this DN.
Corrective Action Taken:	The rIPARK_LOT tuple is deleted. If accessible, the rICLDB, and rIPCBLA tuples are recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical address of the rIPARK_LOT table entry.
<i>Bad Data</i> =	GLFAIL.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1</i> =	The contents of the RLPARK_LOT table entry.
<i>Dump 2</i> =	The contents of the <i>locdn</i> structure which serves as the logical key of the RLPARK_LOT table entry.

Table 7 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The status of the <i>parked</i> bit in the rIPCBLA tuple associated with a call park terminal process was found in an incorrect state. If the call park terminal process and the RLPARK_LOT table entry reside on the same SM (rIPARK_LOT.park_pid.pcrd = GI_pcrd), then there should exist a corresponding rIPCBLA tuple for the related call park terminal process, where ((key)rIPCBLA.procnum = rIPARK_LOT.park_pid.procno) and the <i>parked</i> bit in this rIPCBLA tuple should be set to DBYES.
Possible Error Effect:	The terminal process for the parked call cannot be accessed. Calls associated with the DN marked in the RLPARK_LOT table entry cannot be parked or unparked.
Corrective Action Taken:	The state of the <i>parked</i> bit in the rIPCBLA tuple is corrected to DBYES.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The logical address of the rIPARK_LOT table entry.
<i>Bad Data</i> =	The current state of the <i>parked</i> bit in the rIPCBLA tuple (DBNO).
<i>Good Data</i> =	The state the <i>parked</i> bit was corrected to in the rIPCBLA tuple (DBYES).
<i>Logical Index</i> =	The <i>noc</i> portion of the <i>locdn</i> structure in the RLPARK_LOT table entry.
<i>Dump 1</i> =	The contents of the RLPARK_LOT table entry.
<i>Dump 1</i> =	The contents of the rIPCBLA tuple associated with the RLPARK_LOT table entry.

PATHMAP

Software Release: 5E10 and later

1. SLC[®] 96 PATH MAP (PATHMAP) AUDIT

The digital carrier line unit (DCLU), a SLC[®] 96 circuit in the switching modules, transmits data to the data interface (DI) circuit in the module controller/time-slot interchanger (MC/TSI) over peripheral interface data buses (PIDBs). There are four PIDBs for each DCLU, two per service group; each PIDB contains 32 time slots. One PIDB time slot is assigned per customer. Two arrays keep track of how the PIDB time slots are allocated. The PATHMAP audit verifies the consistency of the information in these arrays.

The PATHMAP audit is part of the peripheral control audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented, nonsegmented, initialization, and post initialization modes.

2. DATA STRUCTURES

The PATHMAP audit verifies data in the two global arrays shown in Figure 1, the SLC[®] 96 data path array (pcS96PATH) and the SLC[®] 96 mapping array (pcS96MAP). The data path array describes the data paths (PIDB time slots) through the DCLUs to the module processors. The array contains an entry for all the PIDB time slots in the module processor. Its entries are computed by DCLU number (0-7), service group number (0-1), PIDB time slot number (0-31), and PIDB number (0-1). Valid values for the entries in the table include unavailable, idle, and if the PIDB time slot is selected in a call path, the SLC[®] 96 digital facilities interface (S-DFI) number. In Figure 1, DCLU number 0, PIDB number 0, service group 1, and PIDB time slot 0 is using S-DFI 12.

The mapping array, pcS96MAP, is a bit representation of pcS96PATH; instead of the full S-DFI numbers, pcS96MAP contains a busy/idle flag for each field in pcS96PATH. It allows faster access to available PIDB time slots. The path map array is accessed by an index made up of the DCLU number and the PIDB time slot number. Note that the bit corresponding to the entry for S-DFI 12 in pcS96PATH is set to one in pcS96MAP. An idle indication in pcS96PATH is a zero in the corresponding entry (bit position) in pcS96MAP.

The other data structures used by the PATHMAP audit are the circuit data block common area relation (RLCDBCOM), the SLC[®] 96 T1 usage relation (RLS96T1USG), and the SLC[®] 96 digital facility interface relation (RLS96DFI). The rLCDBCOM relation is the software entry into the call processing data structures of the module processor. There is a tuple of RLCDBCOM associated with each PIDB time slot. The RLS96T1USG relation keeps track of the T1 time slots in use. The RLS96DFI relation stores the shelf circuit names served by for the S-DFI circuits.

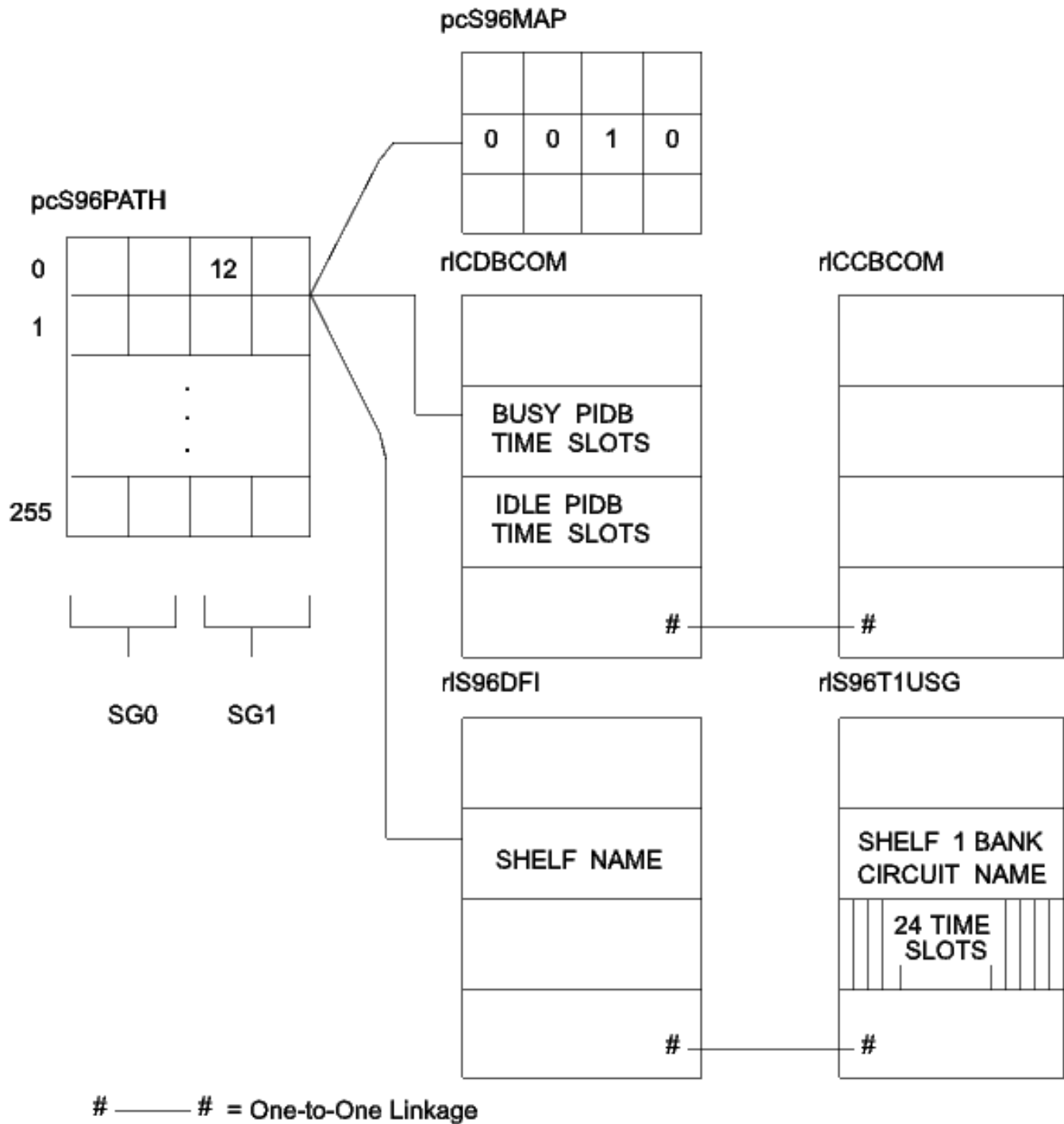


Figure 1 (PATHMAP-1) Data Structures of the PATHMAP Audit

3. AUDIT CHECKS

The audit invokes the path map driver (DRPHMAP) to step through each entry in pcS96PATH and pcS96MAP. The audit checks the consistency of the data path and path map arrays and verifies information about the PIDB time slots in use.

- (a) The audit checks the consistency of the two arrays. Idle entries in the data path array must have zeroes in the

corresponding bits of pcS96MAP; busy and unavailable entries must have ones. If the two arrays are consistent, the audit checks for idle, busy, and unavailable entries.

- (b) Idle entries must also have the tuples in rLCDBCOM associated with their PIDB time slots linked to the idle list. Unavailable entries in the data path array are flagged as busy in the path map array but they are not on the rLCDBCOM idle list.
- (c) The audit checks the busy data path entries as follows: Because each PIDB time slot has a corresponding tuple in rLCDBCOM, the tuples corresponding to busy PIDB time slots must not be linked to the rLCDBCOM idle list. Busy PIDB time slots have S-DFI numbers assigned to them; the audit checks their associated connections. The audit verifies the relationship between the T1 time slot and the PIDB time slot by checking the entry in the T1 usage relation. The audit needs the shelf name of the circuit to access rIS96T1USG; shelf names are stored in rIS96DFI, which is accessed by the S-DFI number stored in the pcS96PATH array. The audit examines each PIDB time slot in rIS96PATH and ensures that it is listed in rIS96T1USG. If the audit cannot find the time slot in rIS96T1USG, an error exists.

4. ERROR HANDLING

The audit treats all errors the same. It sets the entry in pcS96MAP to one and the entry in pcS96PATH to unavailable. The audit then idles the hardware connections in the DCLU. It sets the quarantine bit in RLCDBCOM and purges the related process. When the comparison between the status of an entry is inconsistent between pcS96PATH and pcS96MAP, the audit determines whether a tuple in RLCDBCOM exists for the associated PIDB time slot. If there is no corresponding tuple in RLCDBCOM, the entry in pcS96PATH is made unavailable and the corresponding bit in pcS96MAP is set. If there is a tuple in RLCDBCOM, the tuple is quarantined. The audit purges the process associated with the tuple of RLCDBCOM and invokes the circuit data audit (CKTDATA) to check RLCDBCOM and other data structures. The entry in pcS96PATH is set to idle and the corresponding bit in PCS96MAP is cleared. The associated hardware is put in the idle state and PIDB parity reporting is enabled for the DCLU. See also the T1USG audit description.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below..

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The entry in PCs96path[] says not idle, but the corresponding entry in PCs96map[] says idle.
Possible Error Effect:	When call processing tries to use this PIDB time slot, ASSERTs will be fired and the call may be dropped.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rLCDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the item in PCs96map[] global array being audited; note that each element of PCs96map[] contains 4 bits corresponding to service groups 0 and 1 (bit 1) and PIDB 0 and 1 (bit 0).
<i>Bad Data =</i>	Value the item in PCs96map[] being audited.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number.

	Bit 1 - Service group number.
	Bits 2-6 - PIDB time slot number.
	Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump</i> =	NA.

Table 2 Error Code: DFILOSTTUP

Error Code:	DFILOSTTUP
Error Description:	The audit is not successful in reading the dynamic relation RLS96DFI.
Possible Error Effect:	Calls may not be able to originate.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audits and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCs96path[].s96chan[] being audited.
<i>Bad Data</i> =	Value of PCs96path[].s96chan[] being audited.
<i>Good Data</i> =	Relation ID of rLS96DFI (this field should not be interpreted as good data but as an information field).
<i>Logical Index</i> =	S-DFI circuit name that is used as the key to read the dynamic relation RLS96DFI.
<i>Dump</i> =	A two byte number (short) that is interpreted as follows: bit 0 - Peripheral interface data bus number. bit 1 - Service group number within the digital carrier line unit. bits 2-6 - Peripheral interface data bus time slot number. bits 7-9 - Digital carrier line unit number.

Table 3 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The entry in PCs96path[] says idle, but the corresponding entry in PCs96map[] says not idle.
Possible Error Effect:	Call processing will not use the PIDB time slot because it is marked busy in PCs96map[]. If this error is not corrected, all the entries in PCs96map[] will eventually be busy and no calls can be processed.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the item in PCs96map[] global array being audited; note that each element of PCs96map[] contains 4 bits corresponding to service groups 0 and 1 (bit 1) and PIDB 0 and 1 (bit 0).
<i>Bad Data</i> =	Value the item in PCs96map[] being audited.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number.

	Bit 1 - Service group number.
	Bits 2-6 - PIDB time slot number.
	bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump</i> =	NA.

Table 4 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	The rCDBCOM associated with an idle PIDB time slot does not belong to the miscellaneous idle list.
Possible Error Effect:	It is not clear what the error effect is ????
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the CDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the CDBCOM tuple associated with the entry in PCs96path[] being audited.
<i>Bad Data</i> =	Logical key of the CCBCOM that the CDBCOM claims as its owner.
<i>Good Data</i> =	DBIDLMISC, value of the miscellaneous idle list.
<i>Logical Index</i> =	Circuit name of the pidb time slot.
<i>Dump 1</i> =	First 2 byte is the an entry in PCs96path[] to be audited; the second 2 byte is the item number which is interpreted as follows: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump 2</i> =	rCDBCOM tuple associated with the pidb time slot being audited.

Table 5 Error Code: INV

Error Code:	INV
Error Description:	The item in PCs96path[] that contains SLC [®] - digital facility interface (S-DFI) number associated with the call is not within the range of valid S-DFI numbers.
Possible Error Effect:	SLC [®] 96 call processing resources might be lost.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rCDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the item in PCs96path[].s96chan global array being audited.
<i>Bad Data</i> =	Value the item in PCs96path[].s96chan[] being audited.
<i>Good Data</i> =	Bits 0-3 should be less than 15. Bits 0-7 should be less than 32.
<i>Logical Index</i> =	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number.

	Bits 2-6 - PIDB time slot number.
	Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump =</i>	NA.

Table 6 Error Code: LNLOSTTUP

Error Code:	LNLOSTTUP
Error Description:	The audit cannot read the static relation RLPICB.
Possible Error Effect:	The call capacity is reduced because a DCLU is not known to the system. The audit will repeatedly report the error until it is fixed.
Corrective Action Taken:	This audit schedules the BASE, DFISHELF, T1USG and CDBCOM audits.
Manual Action Required:	The maintenance personnel need to RC/VER the RLPICB relation to correct or prevent future errors.
Dump Description:	
<i>Error Address =</i>	Address of PCs96path[].s96chan[] being audited.
<i>Bad Data =</i>	Value of PCs96path[].s96chan[] being audited.
<i>Good Data =</i>	Relation ID of rPICB (this field should not be interpreted as good data but as an information field).
<i>Logical Index =</i>	The picb key used to the read the static relation rPICB.
<i>Dump =</i>	A two byte number (short) that is interpreted as follows: Bit 0 - Peripheral interface data bus number. Bit 1 - Service group number within the digital carrier line unit. Bits 2-6 - Peripheral interface data bus time slot number. Bits 7-9 - Digital carrier line unit number.

Table 7 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Cannot access a tuple in the dynamic relation RLS96BASE.
Possible Error Effect:	May not be able to set up calls.
Corrective Action Taken:	The audit schedules the BASE, DFISHELF, T1USG and CDBCOM audits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCs96path[].s96chan[] being audited.
<i>Bad Data =</i>	Value of PCs96path[].s96chan[] being audited.
<i>Good Data =</i>	Relation ID of rLS96BASE (this is not really good data, the field is for information only and should not be interpreted as good data).
<i>Logical Index =</i>	Key used to read RLS96BASE dynamic relation.
<i>Dump =</i>	A two byte number (short) that is interpreted as follows: Bit 0 - Peripheral interface data bus number. Bit 1 - Service group number within the digital carrier line unit. Bits 2-6 - Peripheral interface data bus time slot number. Bits 7-9 - Digital carrier line unit number.

Table 8 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The audit cannot read the static relation RLCKTDATA.
Possible Error Effect:	Call processing in CONCENTRATED SLC [®] 96 cannot work and ASSERTs will be fired. The audit will repeatedly report the error until it is fixed.
Corrective Action Taken:	This audit schedules the BASE, DFISHELF, T1USG and CDBCOM audits.

Manual Action Required:	The maintenance personnel needs to recent change in the rICKTDATA relation to correct or prevent future errors.
Dump Description:	
<i>Error Address =</i>	Address of PCs96path[].s96chan[] being audited.
<i>Bad Data =</i>	Value of PCs96path[].s96chan[] being audited.
<i>Good Data =</i>	Relation ID of RLCKTDATA (this field should not be interpreted as good data but as an information field).
<i>Logical Index =</i>	The circuit name key used to the read the static relation rICKTDATA.
<i>Dump =</i>	A two byte number (short) that is interpreted as follows: Bit 0 - Peripheral interface data bus number. Bit 1 - Service group number within the digital carrier line unit. Bits 2-6 - Peripheral interface data bus time slot number. Bits 7-9 - Digital carrier line unit number.

Table 9 Error Code: M2T1_TSLot

Error Code:	M2T1_TSLot
Error Description:	The T1 time slot entry in the dynamic relation is idle for a CONCENTRATED SLC [®] 96 call that is supposed to be busy.
Possible Error Effect:	Call processing resources might not be re-used.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the item in PCs96path[].s96chan[] global array being audited.
<i>Bad Data =</i>	The value in the entry of the corresponding T1 time slot array in rIS96T1USG.
<i>Good Data =</i>	It is either the value printed or a byte value whose most significant bit is a one. The value printed is decoded as follows bits 0-4 peripheral interface data bus time slot number bit 5 peripheral interface data bus number bit 6 service group number within the digital concentration line unit.
<i>Logical Index =</i>	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. bits 2-6 - PIDB time slot number. bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump =</i>	The data in PCs96path[] being audited.

Table 10 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Cannot access the rIS96T1USG tuple.
Possible Error Effect:	The call may not release properly.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the item in PCs96path[].s96chan[] being audited.
<i>Bad Data</i> =	The value of the item in PCs96path[].s96chan[] being audited.
<i>Good Data</i> =	The field stores rIS96T1USG relation ID and should not be interpreted as the good data.
<i>Logical Index</i> =	rIS96SHELF circuit name used to read the relation RLS96T1USG.
<i>Dump 1</i> =	The value in the entry in PCs96path[].s96chan[] being audited.
<i>Dump 2</i> =	Item number which should be decoded as follows: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.

Table 11 Error Code: MODE

Error Code:	MODE
Error Description:	The mode field in RLS96DFI does not contain valid modes.
Possible Error Effect:	Call processing will fire ASSERTs and no calls can be processed for this S-DFI.
Corrective Action Taken:	This auditing of this entry is skipped and the BASE, DFISHELF, T1USG and CDBCOM audits are requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the mode attribute of the tuple in rIS96DFI.
<i>Bad Data</i> =	The mode found in the S-DFI tuple.
<i>Good Data</i> =	Either 0, 1 or 2.
<i>Logical Index</i> =	DFI circuit name used to read the relation rIS96DFI.
<i>Dump</i> =	rIS96DFI tuple.

Table 12 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit cannot read the static relation RLPICB.
Possible Error Effect:	The call capacity is reduced because a DCLU is not known to the system. The audit will repeatedly report the error until it is fixed.
Corrective Action Taken:	This audit schedules the BASE, DFISHELF, T1USG and CDBCOM audits.
Manual Action Required:	The maintenance personnel need to RC/VER the RLPICB relation to correct or prevent future errors.
Dump Description:	
<i>Error Address</i> =	Address of PCs96path[].s96chan[] being audited.
<i>Bad Data</i> =	Value of PCs96path[].s96chan[] being audited.
<i>Good Data</i> =	Relation ID of RLPICB (this field should not be interpreted as good data but as an information field).
<i>Logical Index</i> =	The picb key used to read the static relation RLPICB.
<i>Dump</i> =	A two byte number (short) that is interpreted as follows: Bit 0 - Peripheral interface data bus number. Bit 1 - Service group number within the digital carrier line unit. Bits 2-6 - Peripheral interface data bus time slot number. Bits 7-9 - Digital carrier line unit number.

Table 13 Error Code: OWNER

Error Code:	OWNER
Error Description:	The owner of a busy rICDBCOM is an rICCB COM that belongs to a head cell.
Possible Error Effect:	When the call is released, a link problem is found and ASSERTs will be fired.

Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM associated with the PIDB time slot.
<i>Bad Data =</i>	Owner of this rICDBCOM (a physical key of CCBCOM).
<i>Good Data =</i>	Any logical key in the rICBCOM that is greater than 0 is considered not a head cell.
<i>Logical Index =</i>	Peripheral interface data bus (PIDB) circuit name.
<i>Dump =</i>	Dump of rICDBCOM associated with the PIDB time slot.

Table 14 Error Code: PIDB

Error Code:	PIDB
Error Description:	For the CONCENTRATED SLC [®] 96, the PIDB number used in accessing the global array in PCs96path[] is not the same as the one stored in the T1 time slot array in the corresponding tuple in the dynamic relation RLS96T1USG.
Possible Error Effect:	Call processing resources might not be able be re-used.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the item in PCs96path[].s96chan[] global array being audited.
<i>Bad Data =</i>	PICB number which is the information used in accessing the global array PCs96path[].
<i>Good Data =</i>	PIDB number obtained from the T1 time slot array in the corresponding tuple in RLS96T1USG.
<i>Logical Index =</i>	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump 1 =</i>	T1 time slot number.
<i>Dump 2 =</i>	Path data entry in PCs96path[].s96chan[] being audited.

Table 15 Error Code: PIDBTS

Error Code:	PIDBTS
Error Description:	For the CONCENTRATED SLC [®] 96, the PIDB time slot number used in accessing the global array in PCs96path[] is not the same as the one stored in the T1 time slot array in the corresponding tuple in the dynamic relation RLS96T1USG.
Possible Error Effect:	Call processing resources might not be re-used.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	

<i>Error Address =</i>	Address of the item in PCs96path[].s96chan[] global array being audited.
<i>Bad Data =</i>	PIDB time slot no. which is the information used in accessing the global array PCs96path[].s96chan[].
<i>Good Data =</i>	PIDB time slot no. obtained from the T1 time slot array in the corresponding tuple in rls96T1USG.
<i>Logical Index =</i>	Valid values are: Bit 0 - Peripheral interface data bus(PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump 1 =</i>	T1 time slot number.
<i>Dump 2 =</i>	Path data entry in PCs96path[].s96chan[] being audited.

Table 16 Error Code: SGROUP

Error Code:	SGROUP
Error Description:	For the CONCENTRATED SLC [®] 96, the service group number used in accessing the global array in PCs96path[] is not the same as the one stored in the T1 time slot array in the corresponding tuple in the dynamic relation RLS96T1USG
Possible Error Effect:	Call processing resources might not be reused.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rCDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the item in PCs96path[].s96chan[] global array being audited.
<i>Bad Data =</i>	Service group no. is the information used in accessing the global array PCs96path[].s96chan[].
<i>Good Data =</i>	Service group no. obtained from the T1 time slot array in the corresponding tuple in RLS96T1USG.
<i>Logical Index =</i>	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump 1 =</i>	T1 time slot number.
<i>Dump 2 =</i>	Path data entry in PCs96path[].s96chan[] being audited.

Table 17 Error Code: T1_TIME_SLOT

Error Code:	T1_TIME_SLOT
Error Description:	For the NONCONCENTRATED SLC [®] 96, the service group number, the PIDB number and the PIDB time slot number used in accessing the global array in PCs96path[] is not the same as the ones stored in the T1 time slot array in the corresponding tuple in the dynamic relation RLS96T1USG.
Possible Error Effect:	Call processing resources might not be re-used.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the CDBCOM associated with the PIDB time slot is quarantined; any valid terminal

	process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the item in PCs96path[]s96chan[] global array being audited.
<i>Bad Data =</i>	The value in the entry of the global array PCs96path[]s96chan[] being audited.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump =</i>	rlS96T1USG tuple.

Table 18 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The rICDBCOM associated with an unavailable PIDB time slot belong to the miscellaneous idle list (which implies that this PIDB time slot can be used).
Possible Error Effect:	This call may be hung and the resources associated with the call cannot be used by the system.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PCs96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested, that is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple associated with the entry in PCs96path[] being audited.
<i>Bad Data =</i>	DBIDLISC, value of the miscellaneous idle list.
<i>Good Data =</i>	Any value other than the value in BAD DATA.
<i>Logical Index =</i>	Circuit name of the pidb time slot.
<i>Dump 1 =</i>	First 2 byte is the an entry in PCs96path[] to be audited; the second 2 byte is the item number which is interpreted as follows: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump 2 =</i>	rICDBCOM tuple associated with the PIDB time slot being audited.

Table 19 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The global array PCs96path[] does not indicate unavailability for a PIDB time slot that is not equipped.
Possible Error Effect:	When call processing tries to use this PIDB time slot, it cannot do it and will cause asserts to fire, the call might be dropped.
Corrective Action Taken:	The entry that is found in error is written with the available code in PCs96path[] and the busy code in PCs96map[], and the BASE, DFISHELF, T1USG and

	CDBCOM audits are requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the item in PCs96path[] global array being audited; note that each element of PCs96path[] contains 4 items corresponding to service groups 0 and 1 (bit 1) and PIDB 0 and 1 (bit 0).
<i>Bad Data</i> =	Value of the item in PCs96path[] being audited.
<i>Good Data</i> =	PCS96UNAV, the code that indicates unavailability.
<i>Logical Index</i> =	Valid values are: Bit 0 - Peripheral interface data bus (PIDB) number. Bit 1 - Service group number. Bits 2-6 - PIDB time slot number. Bits 7-9 - Digital carrier line unit (DCLU) number.
<i>Dump</i> =	NA.

Table 20 Error Code: ZAP

Error Code:	ZAP
Error Description:	An interrupt causes the audit to be entered in the ZAP manner.
Possible Error Effect:	The audit may repeated cause the interrupt to occur which might escalate to higher levels of initializations.
Corrective Action Taken:	The audit blind idles the last audited item.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of variable that stores the DCLU number.
<i>Bad Data</i> =	The digital carrier line unit (DCLU) number being audited. The valid range is 0 to 7. Note that the DCLU number should not necessary be bad. This field is an information field which gives the DCLU number of the item that causes the audit to be entered in the AUZAP mode.
<i>Good Data</i> =	0.
<i>Dump</i> =	NA.

PBCB

Software Release: 5E12 and later

1. PACKET BUS CONTROL BLOCK (PBCB) AUDIT

The PBCB audit was created for the PSU restructuring feature (99-5E-3114), which created a new data element, the PBCB, to contain data that is specific to control of the packet bus. This data had previously existed in the LLCB. This audit runs for virtually all PH types. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The PBCB audit uses the packet bus control block array (PSPbcb) and has linkages to the timer element array (PStelem).

3. AUDIT CHECKS

The audit makes semantic checks to ensure the correctness of data in PSPbcb, and that any associated timers are correctly linked.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BAD_FAILED

Error Code:	BAD_FAILED
Error Description:	PS has failed it's test of a PBCB.
Possible Error Effect:	Prints out a report and recovers the PBCB.
Corrective Action Taken:	The audit reports the error and recovers the PBCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PS return code from check function.
<i>Bad Data =</i>	PBCB's type field.
<i>Good Data =</i>	The type that this PBCB should be.
<i>Logical Index =</i>	PBCB index.
<i>Dump =</i>	PBCB.

Table 2 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The audit found a PBCB which has a timer running when it is not supposed to.
Possible Error Effect:	Prints out a report and recovers the PBCB.
Corrective Action Taken:	The audit reports the error and recovers the PBCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PBCB's type field.
<i>Bad Data =</i>	PBCB's tidx field.
<i>Good Data =</i>	PBCB's state field.
<i>Logical Index =</i>	PBCB index.
<i>Dump =</i>	PBCB.

Table 3 Error Code: QUAR

Error Code:	QUAR
Error Description:	The audit found a PBCB with its quarantine bit set.
Possible Error Effect:	Prints out a report and recovers the PBCB.
Corrective Action Taken:	The audit reports the error and recovers the PBCB to the free linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	PBCB's type field.

<i>Bad Data</i> =	PBCB's quar_bit field.
<i>Good Data</i> =	PBCB's state field.
<i>Logical Index</i> =	PBCB index.
<i>Dump</i> =	PBCB.

Table 4 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up effects of a possible SPP.
Possible Error Effect:	Prints a report and recovers the PBCB if the index is in range.
Corrective Action Taken:	Prints a report and recovers the PBCB if the index is in range.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSMAXPBCB - one more than the maximum value the PBCB index number can be.
<i>Logical Index</i> =	PBCB index.
<i>Dump</i> =	PBCB if the index is in range.

PCB

Software Release: 5E10 and later

1. PROCESS CONTROL BLOCK (PCB) AUDIT

A process consists of a process control block (PCB), and an stack control block (SCB). The PCB stores all the data required to represent and administer a process. This includes the process's state, priority, links to lists of timers and messages, a stack index, and linkages to other processes.

A fixed number of process control blocks is provided by operating system for distributed switching (OSDS) in each processor.

The PCB audit runs in the operational kernel process (OKP) and switch maintenance kernel process (SMKP) environments in the administrative module (AM), the switching module (SM) and the communications module processor (CMP).

2. DATA STRUCTURES

Figure 1 illustrates the layout of a process control block and shows the items checked by audits.

In addition to the PCBs themselves (OSbpcb[] in the administrative module and communication module processor, tuples of rIOSPCB in the switching module), the audit verifies and uses the following global data:

- The array of process ready queues (OSrdyq[]) each of which may own a list of PCBs.
- The global indexes OSffpcb and OSlfpcb which indicate the first and last members of the doubly linked PCB idle list.
- The global indexes OSfwpcb and OSlwpcb which indicate the first and last members of the doubly linked list of processes waiting for memory allocation.
- The static global table of terminal process program information (OSprogtab[]).

Figure 2 shows the organization of the ready queue.

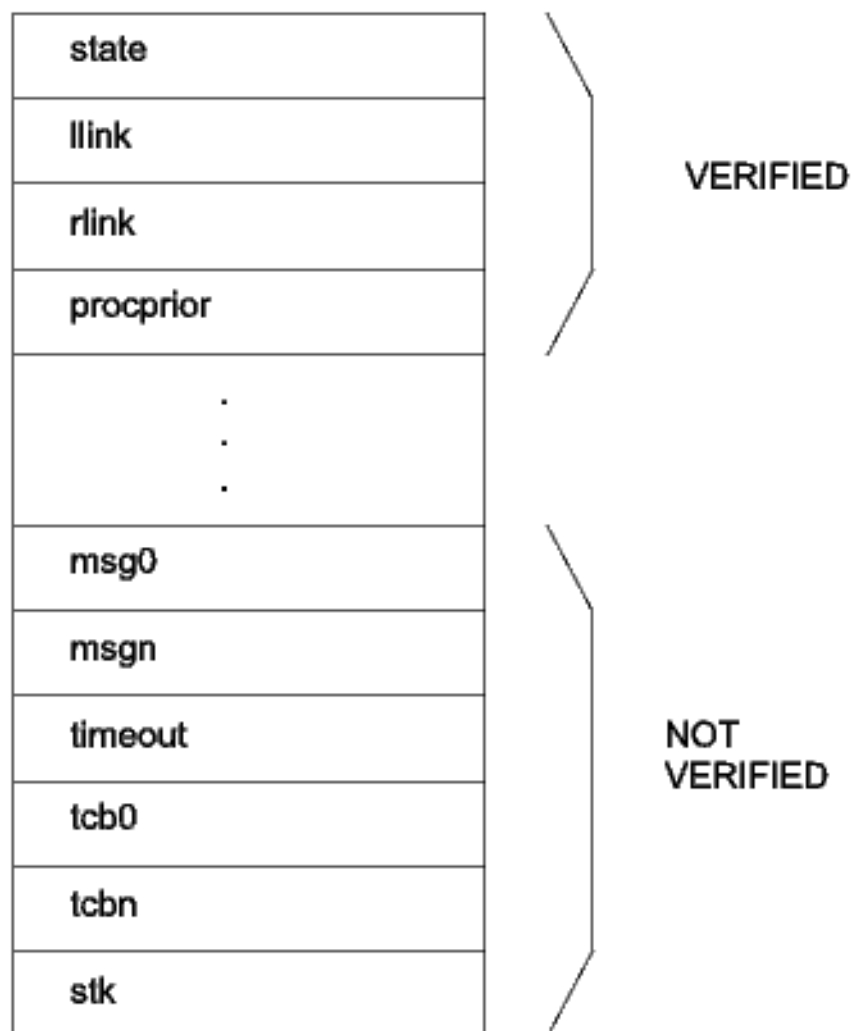


Figure 1 (PCB-1) Process Control Block

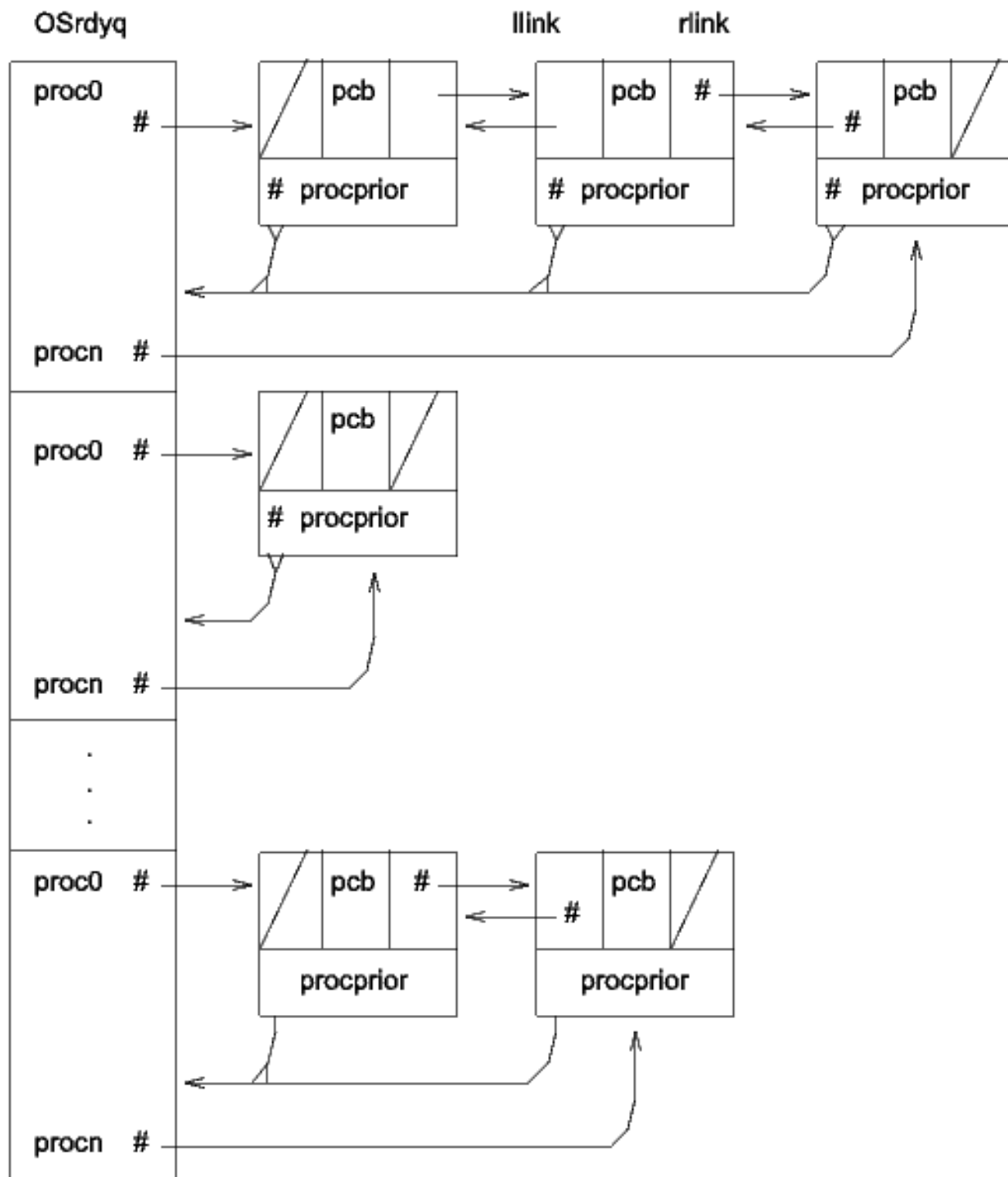


Figure 2 (PCB-2) Organization of the Ready Queue

3. AUDIT CHECKS

The PCB audit does not use any audit driver. Comment blocks for all error codes associated with this audit are

given. The following list gives a basic indication of the order of the checks made by this audit.

If the audit is being (re)entered:

ZAP = Check for system interrupt handling.

If the audit is attempting to access the PCB:

MISSING = Check for inaccessible PCBs.

If the audit is verifying the waiting for memory allocation list:

LNK2ALST = Range check OSlwpcb.

ALNK2QUE = Semantic check of OSlwpcb.

BLNKINQUE = Semantic check of OSfwpcb.

LNKIALST = Range check idle PCB's llink field.

PT2PT = Point-to/point-back check between PCBs.

STATE = Check PCB's state.

If the audit is verifying the idle list:

LNK2ALST = Range check OSlfpcb.

ALNK2QUE = Semantic check of OSlfpcb.

BLNKINQUE = Semantic check of OSffpcb.

LNKIALST = Range check idle PCB's llink field.

PT2PT = Point-to/point-back check between PCBs.

BADOWNER = Check PCB's procprior field.

NOTIDLE = Check PCB's state.

If the audit is verifying an PCB list attached to a ready queue:

LNK2ALST = Range check OSreadyq[].procn.

ALNK2QUE = Semantic check of OSreadyq[].procn.

BLNKINQUE = Semantic check of OSreadyq[].proc0.

LNKIALST = Range check PCB.llink.

PT2PT = Point-to/point-back check between PCBs.

BADOWNER = Check PCB.procprior.

STATE = Check PCB.state.

If the PCB is not idle:

TIMEOUT = Check for hung terminal processes.

If the audit is looking for lost PCBs:

KEY = Check PCB.pcbkey.

INVOWNER = Check potentially lost PCB's procprior field.

INV = Check for lost PCBs.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALNK2QUE

Error Code:	ALNK2QUE
Error Description:	An index to the last element in a list of process control blocks (PCBs) (OSIfpcb if the free list is being checked, OSIfwpcb if the WAITMEM list is being checked, otherwise OSrdyq[].proc0 for some ready queue PCB) indicates a PCB which either has an incorrect owner (not OSNULL for the free list, different from the ready queue priority in all other cases) or has a non-null (not OSNULL) rlink field.
Possible Error Effect:	For the free list this could mean lost PCBs (leading to overload situations) or non-idle PCBs being taken for new processes (affecting the process to which they rightfully belong). For PCB lists attached to a ready queue or the WAITMEM list, this problem could cause processes to never get scheduled. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the PCB audit in directed mode. It is also likely that PCB "INV" errors will be seen along with this possible error effect.
Corrective Action Taken:	In all cases, if the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying OSffpcb (index to the first free PCB) or OSfwpcb (index to the first WAITMEM PCB) or OSrdyq[].proc0, and verifying the list from the top down, if possible.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCB.
<i>Bad Data</i> =	The owner of the list we're trying to access (either OSNULL or a ready queue index).
<i>Good Data</i> =	Contents of the PCB's procprior field.
<i>Logical Index</i> =	The PCB's index (= OSIfpcb in the free list case, OSIfwpcb in the WAITMEM case and OSrdyq[].procn otherwise).
<i>Dump 1</i> =	The PCB.
<i>Dump 2</i> =	The owning ready queue or nothing if the free list or memory allocation list was being checked.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	A PCB's procprior field shows a different owner than the owner of the list in which it was found. Note that the PCB's rlink field has already been verified (PT2PT) and its llink has been checked to be in range (BLNKINQUE/LNKIALST).
Possible Error Effect:	Regardless of where the PCB was found, it is possible that it actually belongs on some ready queue list. In this case it could fail to be scheduled or be scheduled incorrectly. PCB INV errors are also likely to show up.
Corrective Action Taken:	The audit will relink the list around the bad message (using the indexes in PCB.llink and PCB.rlink). If the PCB truly belongs on the idle list, an INV error will fire later and the audit will then put it there.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCB.
<i>Bad Data =</i>	Contents of the PCB's procprior field.
<i>Good Data =</i>	The owner of the list the PCB was found in (either OSNULL or a ready queue index).
<i>Logical Index =</i>	The PCB's index (= OSlfpcb in the free list case, = OSrdyq[].procn otherwise).
<i>Dump =</i>	The PCB.

Table 3 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	While traversing a list (bottom up) a PCB was found with its llink field = OSNULL, but it was not the first PCB in the list according to the list owner. Specifically, if the PCB was found on the free list, it was not indicated by OSffpcb (index to the first free PCB). If the PCB was found on the WAITMEM list it was not indicated by OSfwpcb (index to the first WAITMEM PCB). If the PCB was found on a ready queue, it was not indicated by the ready queue's proc0 field (index to the first PCB of this priority).
Possible Error Effect:	The list is either broken, or the list owner's index to the first element is corrupt. This could have any of the effects of an ALNK2QUE, LNKIALST or PT2PT error.
Corrective Action Taken:	In all cases, if the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying either OSffpcb (index to the first free PCB), OSfwpcb (index to the first WAITMEM PCB) or OSrdyq[].proc0, and if it is correct, verifying the list as far as possible from the top down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCB.
<i>Bad Data =</i>	Index of the PCB, the list's owner says is first on the list.
<i>Good Data =</i>	The owner of the list the PCB was found in (either OSNULL or a PCB index).
<i>Logical Index =</i>	The PCB's index.
<i>Dump 1 =</i>	The PCB.
<i>Dump 2 =</i>	The owning PCB or nothing if the free list was being checked.

Table 4 Error Code: INV

Error Code:	INV
Error Description:	An PCB has been found which is not linked to any ready queue or the free list, and is not in a valid transient state.
Possible Error Effect:	The PCB is lost and inaccessible to the system. If enough PCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	The PCB is cleared and linked to the idle list. In particular, its procprior field is

	set to OSNULL and its state is set to OSEMPY.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCB.
<i>Bad Data</i> =	0 (Indicates that the audit did not find the PCB on any list).
<i>Good Data</i> =	1 (Indicates that the audit should have found the PCB on some list).
<i>Logical Index</i> =	The PCB's index.
<i>Dump</i> =	The PCB.

Table 5 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	A PCB has been found which the audit did not see on any list or in a valid transient state. Before taking an INV error, the audit will verify that the PCB is truly lost. For example, if the audit were running in the segmented mode, the PCB may have changed lists while the audit took a timebreak. In this case, the audit would not have seen the PCB before now. The audit uses the PCB's procprior field to determine which list to check. This error says that the procprior field is out of range (not OSNULL and not a ready queue index).
Possible Error Effect:	The PCB is effectively lost and inaccessible to the system. If enough PCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	If possible, the PCB is unlinked from the list it's in (the linkages of the PCBs it points to are reset). In any case, the PCB is cleared and linked to the idle list. In particular, its procprior field is set to OSNULL and state to OSEMPY.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCB.
<i>Bad Data</i> =	Contents of the PCB's procprior field.
<i>Good Data</i> =	OSNULL.
<i>Logical Index</i> =	The PCB's index.
<i>Dump</i> =	The PCB.

Table 6 Error Code: KEY

Error Code:	KEY
Error Description:	A PCB's key (PCB.pcbkey) did not match the key of the PCB tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	PCB.pcbkey is set to the tuple key value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Key from the PCB (PCB.pcbkey).
<i>Good Data</i> =	The key of the PCB tuple.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The PCB.

Table 7 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Either the tail of a PCB list was null when the head was not or the tail of a PCB list is out of range. The list owner (see good data) is either an OS (operating system) "ready queue" (entry in OSrdyq[]) or the WAITMEM list or the free list (OSNULL). If it is a ready queue, the out-of-range value is the queue's procn field. If the owner is the WAITMEM list, the out-of-range value is the last WAITMEM PCB (OSlwpcb). If the owner is the free list, the out of range value is the last free PCB index (OSlfpcb).
Possible Error Effect:	If the owner of the list is a ready queue or WAITMEM list, processes of the

	priority associated with that queue or on the WAITMEM list may be lost (fail to get scheduled). The results of this depend on the process(es) involved. If the free list is involved, this could result in lost PCBs which could, eventually, lead to OSDS PCB overloads.
Corrective Action Taken:	If the audit is running segmented, it will escalate to directed mode. If the audit is already running in directed mode it will attempt to rebuild the list by verifying either OSffpcb (index to the first free PCB), OSfwpcb (index to the first WAITMEM PCB) or OSrdyq[].proc0, and if it is correct, verifying the list as far as possible from the top down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The out of range PCB index.
<i>Good Data</i> =	The list owner (a ready queue index or OSNULL for the idle list).
<i>Logical Index</i> =	OSNULL.
<i>Dump</i> =	The ready queue (if any) which owns the list.

Table 8 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	While traversing a list, a PCB was found with an out of range llink field (not OSNULL and not a PCB index).
Possible Error Effect:	The list is broken. For the free list, this could mean lost PCBs (leading to overload situations) or non-idle PCBs being taken for new processes (affecting the process to which they rightfully belong). For PCB lists attached to a ready queue or the WAITMEM list, this problem could cause processes to never get scheduled. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the PCB audit in directed mode. It is also likely that PCB "INV" errors will be seen along with this possible error effect.
Corrective Action Taken:	In all cases, if the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying either OSffpcb (index to the first free PCB), OSfwpcb (index to the first WAITMEM PCB) or OSrdyq[].proc0, and if it is correct, verifying the list as far as possible from the top down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCB.
<i>Bad Data</i> =	The PCB's llink field.
<i>Good Data</i> =	OSPCBNUM (llink should have been OSNULL or less than this value).
<i>Logical Index</i> =	The PCB index.
<i>Dump</i> =	The PCB.

Table 9 Error Code: MISSING

Error Code:	MISSING
Error Description:	The meaning of this error depends on the value of <i>Error Address</i>: AUPCBI = The audit was unable to access a PCB tuple even though the key was valid. AUTCBI = The audit was unable to access a TCB tuple even though the key was valid.
Possible Error Effect:	Effectively, the tuple is lost to the system and can not be used.
Corrective Action Taken:	The audit will attempt to insert the missing tuple into the database.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Tuple type indicator.
<i>Bad Data</i> =	Value returned from database access attempt.
<i>Good Data</i> =	The current maximum valid tuple key + 1.
<i>Logical Index</i> =	Index of the tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A PCB was found correctly linked in the free list with its state equal to something other than OSEMPY.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PCB will be removed from the list and the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCB.
<i>Bad Data</i> =	The PCB's state.
<i>Good Data</i> =	PCB state "OSEMPY".
<i>Logical Index</i> =	Index of the PCB.
<i>Dump</i> =	The PCB.

Table 11 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	While traversing a list, a PCB was found whose llink field contained the index of a PCB whose rlink field was not equal to the first PCB's index.
Possible Error Effect:	The list is broken. For the free list, this could mean lost PCBs (leading to overload situations) or non-idle PCBs being taken for new processes (affecting the process to which they rightfully belong). For PCB lists attached to a ready queue or the WAITMEM list, this problem could cause the process to never get scheduled. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the PCB audit in directed mode. It is also likely that PCB "INV" errors will be seen along with this possible error effect.
Corrective Action Taken:	In all cases, if the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying either OSffpcb (index to the first free PCB), OSfwpcb (index to the first WAITMEM PCB) or OSrdyq[].proc0, and if it is correct, verifying the list as far as possible from the top down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCB.
<i>Bad Data</i> =	The PCB's llink field.
<i>Good Data</i> =	The rlink field of the PCB indexed by <i>Bad Data</i> .
<i>Logical Index</i> =	The PCB's index.
<i>Dump</i> =	The PCB.

Table 12 Error Code: STATE

Error Code:	STATE
Error Description:	One of the following is true: A process control block (PCB) found on a ready queue list has a state (PCB.state) equal to something other than READY.

	something other than WAITMEM or does not have an active timer (PCB.timeout = OSNULL).
Possible Error Effect:	The process associated with this PCB may be scheduled inappropriately.
Corrective Action Taken:	The PCB will be removed from the list and the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCB (if any).
<i>Bad Data =</i>	The PCB's state.
<i>Good Data =</i>	Either the PCB state READY (ready queue) or the number of priorities OSQNUM (WAITMEM list).
<i>Logical Index =</i>	Index of the PCB.
<i>Dump =</i>	The PCB.

Table 13 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	A non-idle (PCB.state != OSEMPY) terminal process whose mortality bit is set (OSprogtab[PCB.progij].mortality = 1) has existed longer than six hours (the maximum allowed for such a process).
Possible Error Effect:	Unavailability of all resources owned by this process.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCB.
<i>Bad Data =</i>	Number of milliseconds since the process was created (return from OSGETPTIME(...) for this process).
<i>Good Data =</i>	Six hours (in milliseconds).
<i>Logical Index =</i>	Index of the PCB.
<i>Dump =</i>	The PCB.

Table 14 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, an out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will idle the last PCB it was looking at (if any), abort, and be restarted in directed mode. Otherwise it will just be restarted in directed mode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCB (if any).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index of the PCB (if any).
<i>Dump =</i>	The PCB (if any).

PCBLA-A

Software Release: 5E10 - 5E12

1. PROCESSOR CONTROL BLOCK LINKAGE AREA (PCBLA) AUDIT

Call processing and terminal and switch maintenance are implemented by means of processes. Processes are associated with data blocks called process control block linkage areas. Call processing functions are implemented by terminal processes. One terminal process is associated with each terminal (line, trunk, and so forth) in a call. A terminal process control block linkage area contains data that describes the process, links it to other entities (such as ports and time slots), and controls the call itself (such as call waiting or forwarding data).

Incorrect data in a process control block linkage area can cause a variety of problems related to call processing. For example, broken linkage can cause processes to tie up terminals indefinitely when a call does not exist; or bad feature data can cause faulty processing of features (such as call forwarding and call waiting).

The process control block linkage area audit (PCBLA) verifies data in the process control block linkage areas resident in the switching modules (SMs). Syntactic checks are performed on all process control block linkage areas (terminal, terminal maintenance, switch maintenance, and system process process control block linkage areas). Semantic checks are performed only on terminal process control block linkage areas.

This audit is part of the feature control audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

The audit runs in the routine mode in the normal shift cycle and in the elevated mode when scheduled by a user request.

2. DATA STRUCTURES

The data structure checked by the PCBLA audit is the process control block linkage area relation (RLPCBLA). Figure 1 illustrates the layout of RLPCBLA and shows the items checked by the audit. The RLPCBLA relation has owner-member linkage with the coupler relation (RLCPLR), the path data block relation (RLPHDB), and the channel control block common area relation (RLCCBCOM). It also has a one-to-one linkage with the port linkage area relation (RLPORTLA), D-channel application linkage block (RLDALB), IDCU EOC level 7 administration remote terminal embedded operation channel linkage block (RLRTLb), and call buildup data block (RLCBDB). The following linkage check is performed by the PCBLA audit due to the addition of linkage structures for RLPCBLA.

- To verify that PCBLA owns the call leg data block (CLDB) or the association process block (APB).
- To verify that PCBLA owns the IDCU remote terminal linkage block (RLRTLb) if the rIPCBLA.rtlb_flag is set.

3. AUDIT CHECKS

This audit verifies the office database (ODB) call record flag with the presence of an ODB call process.

To verify the linkages associated with the process control block linkage areas, the PCBLA audit calls DRINASEK driver once for each audit segment. After verifying the owner-member linkages of each process control block linkage area, the DRINASEK driver invokes a semantic check function that performs the following checks:

- (a) All OSDS process control blocks that are not currently in use are called dead or inactive processes. The process control block linkage areas of dead processes must not be linked to any ports or to any other data structures. They should be on an idle list. The audit determines whether a process is dead and, if it is, checks to see that its process control block linkage area is not linked to anything.
- (b) Active processes are either transient or stable, depending on the content of the customer state (custat) item

of the process's process control block linkage area (for example, dialing is transient and talking is stable). Transient processes with a customer state that has not changed in over five minutes are considered to be in error.

- (c) The link state (state) item of the process control block linkage area contains a value that indicates that the process control block linkage area is in one of two states: the process control block linkage area is linked to a port or coupler, or the process control block linkage area is not linked to a port or coupler. For stable processes, the link state item value is compared to the actual linkage.
- (d) For active processes, the port item of the process control block linkage area contains the name of the linked port or, if the process control block linkage area is linked to more than one port, the name of the first linked port. For stable processes, the audit verifies that the process control block linkage area is actually linked to the port whose name is contained in the port item or, if the process control block linkage area is linked to more than one port, that the first port is the port indicated in the port item.
- (e) If the call waiting in progress (cw_inprog) item of the process control block linkage area is false, the customer state item (custat) may contain any value except CWTALK. If the cw_inprog item is true, custat may contain any value.
- (f) If the customer state (CUSTAT) is FCC_TALK and the class of service is FC_LNQU, then a related non-idle QM_MLHG queued member must exist. Or, if the class of service is FC_OHQU, then a related, non-idle QM_OGT queued member must exist. When either of these conditions exists, a customer is connected to an announcement or tone.
- (g) If the wide-band call attribute (rIPCBLA.wb_call) is DBYES and rIPHDB tuples are linked to the rIPCBLA tuple, the head rIPHDB.masterpath must be DBYES and it must be a wide-band type of path (rIPHDB.pathtyp = PHWIDEBAND).

The linkage checks between rIPCBLA and rCBDB, rDALB are covered by the AUDrinasek() drivers. In addition to the existing semantic checks, the following checks are done to accommodate the data structure changes mentioned:

- If the process is associated with the message traffic on IDCU EOC channels, the rIPCBLA.rtlb_flag is set. When the flag is set, a linkage should exist between the rIPCBLA and rRTLb. This is checked because the same linkage is used for rDALB tuples.
- If the process associated with this rIPCBLA tuple is not existent, this rIPCBLA tuple is not linked to any other data structures including CBDB and DALB.
- If this terminal process owns a B-port, verify that the DSL containing this B-port is the same as that specified in the lport field. The lport is logical port used to condense the key for a D-port. The new static relation (RLLPT-PT) can be used for mapping a logical port to the physical port.
- The logical port is moved from one process to another when two calls need to use the same B-channel. After the logical port is removed, the RTA subsystem will relocate the physical port (PORTLA) to another process and indicate the status of linkage as DBPSHOLD. At this moment, the PCBLA tuple associated with this call-on-hold should not have linkage to PORTLA and CCBCOM tuples. The PCBLA audit verifies this by checking the *state* field (DBPSHOLD) against the corresponding linkages.
- Verify logical port and party in the PCBLA from the terminal equipment data block (TEDB) which contains the feature subscribed by the user.
- If the ACDBRCS software package is loaded and this PCBLA represents a monitor bridge terminal process, the audit will verify that the **mon_pid** field in the corresponding rIACDPOSDB tuple contains a valid PID that is represented by this PCBLA tuple.

- If the PCBLA is used for default service origination, check that the service subscribed is in the static relation Digital Subscriber Line Equipage.
- Verify odbcall bit for on-demand-packet terminal process.

If any of these checks fail, control is returned to the DRINASEK driver with an error indication.

4. ERROR HANDLING

If the process control block linkage area of a dead process is found to be linked to anything, the linkage is broken. All other errors result in purging of the process whose process control block linkage area is in error. In all cases, errors are reported by means of output messages.

The audit produces the following error codes and those generated by the DRINASEK driver.

If there is an ODB call record, but no ODB call process owning the port, the ODB call will be torn down.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	For a given rIPCBLA tuple where the wide-band call attribute (rIPCBLA.wb_call) is DBYES and it is associated with one or more rIPHDB tuples, the head rIPHDB.pathtyp is not PHWIDEBAND.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIPCBLA or rIPHDB tuples.
Corrective Action Taken:	The rIPCBLA tuple is recovered to an idle state; related audits will be run to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPHDB tuple attribute pathtyp.
<i>Good Data =</i>	rIPCBLA tuple attribute wb_call.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIPHDB tuple.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The external information system link number in rIPCBLA tuple doesn't match the static rDLDSL tuple attribute xnamrl.
Possible Error Effect:	Listing services call processing isn't allowed on this EIS link.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_inlk_num with the static rDLDSL tuple attribute xnamrl.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_inlk_num.
<i>Good Data =</i>	rDLDSL tuple attribute xnamrl.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rDLDSL tuple.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The count value stored in a MSslnnode[] SM node to node pair array entry for the 2479 seizure count measurements (MSslnnode[].MS2479ac_inscz) was found to be inaccurate. This count should be the sum of the counts for this entry calculated by the PCBLA audit after sampling all tuples in the RLPCBLA relation, where: $MSslnnode[X].MS2479ac\_inscz = \sum MSnodeidx[OFFSET]$; for $OFFSET = (rIPCBLA.in\_ars\_idx * 2) - (rIPCBLA.data\_rt - 1)$.
Possible Error Effect:	Incorrect 2479 seizure count measurements for offices associated with SM node-to-node pairs in an FTS2000 network.
Corrective Action Taken:	The measurement count for the MSslnnode[].MS2479ac_inscz array entry is corrected to the value calculated after all rIPCBLA tuples have been sampled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the MSslnnode[] array entry found in error.
<i>Bad Data =</i>	The contents of the MSslnnode[].MS2479ac_inscz 2479 counts.
<i>Good Data =</i>	The correct value calculated by the audit and the number of note-to-node pairs associated with the office (calculated 2479 counts << 16 Gln_nodes).
<i>Logical Index =</i>	The index of the MSslnnode[] array entry found in error.
<i>Dump =</i>	NA.

Table 4 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	An rIPCBLA has an invalid customer state value.
Possible Error Effect:	The process would be purged the next time it receives a stimulus.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of custat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 5 Error Code: CUSTAT

Error Code:	CUSTAT
Error Description:	The PCBLA customer state indicates an MCC call, but has no CLDBs linked to it; conversely, the PCBLA customer state indicates a POTS call, but has CLDBs linked to it.
Possible Error Effect:	The CLDBs owned by a POTS call will be lost. An MCC call would be improperly executed.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of pcbla.
<i>Bad Data =</i>	Contents of custat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	pcbla number.
<i>Dump =</i>	pcbla.

Table 6 Error Code: DDL_MSG_FLD

Error Code:	DDL_MSG_FLD
Error Description:	The rIPCBLA tuple attribute eis_msg_dly doesn't match the static rEIS_DATA tuple attribute eis_msg_dly.
Possible Error Effect:	Call processing messages will be lost.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_msg_dly with static rEIS_DATA tuple attribute eis_msg_dly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_msg_dly.
<i>Good Data =</i>	rEIS_DATA tuple attribute eis_msg_dly.

<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.

Table 7 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The operating system believes the process associated with this rIPCBLA to be dead but the linkage fields are not null.
Possible Error Effect:	The next process to use this rIPCBLA may be mistakenly purged by an audit. A lost call will result.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Index of linkage map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	Process control block.

Table 8 Error Code: DFLG

Error Code:	DFLG
Error Description:	PCBLA is used for default service but lport and lpt_type in PCBLA are not set correctly.
Possible Error Effect:	CPE cannot use the subscribed default service.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	lport in PCBLA.
<i>Good Data</i> =	lport in DSLEQUIP.
<i>Logical Index</i> =	The key of the PCBLA tuple.
<i>Dump</i> =	The rIPCBLA tuple.

Table 9 Error Code: EISID

Error Code:	EISID
Error Description:	The rIPCBLA tuple attribute eis_id isn't equal to static rIDLDSL tuple attribute xnamid.
Possible Error Effect:	Listing services calls will not receive automatic treatment.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_id with static rIDLDSL tuple attribute xnamid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_id.
<i>Good Data</i> =	rIDLDSL tuple attribute xnamid.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rIDLDSL tuple.

Table 10 Error Code: EISSWID

Error Code:	EISSWID
Error Description:	The rIPCBLA tuple attribute eis_swid isn't equal to static rEIS_DATA tuple attribute eis_swid.
Possible Error Effect:	Sending the data to the wrong SM.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_swid with static rEIS_DATA tuple attribute eis_swid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_swid.
<i>Good Data</i> =	rEIS_DATA tuple attribute eis_swid.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).

<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.

Table 11 Error Code: EISTP

Error Code:	EISTP
Error Description:	The external information system terminal process number in found in rIPCBLA (procnum) doesn't match the terminal process number found in the rEIS_LNK tuple (eistp.procno).
Possible Error Effect:	The EIS link will not be placed in service to make calls. If an EIS link is in service the calls will be lost to the system.
Corrective Action Taken:	Purge the terminal process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rEIS_LNK tuple terminal process (eistp.procno).
<i>Good Data =</i>	rIPCBLA tuple terminal process (procnum).
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_LNK tuple.

Table 12 Error Code: EISVENDOR

Error Code:	EISVENDOR
Error Description:	Attribute rIPCBLA EIS vendor isn't equal to static rEIS_DATA tuple attribute eis_vendor.
Possible Error Effect:	Call processing messages will be lost while communicating with the wrong EIS vendor.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_vendor with static rEIS_DATA tuple attribute eis_vendor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_vendor.
<i>Good Data =</i>	rEIS_DATA tuple attribute eis_vendor.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.

Table 13 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The rIPCBLA attribute invalid is set to DBYES.
Possible Error Effect:	Receiving invalid call processing messages.
Corrective Action Taken:	Update rIPCBLA tuple attribute invalid with DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute invalid.
<i>Good Data =</i>	DBNO
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 14 Error Code: GPORT

Error Code:	GPORT
Error Description:	(1) currently active music on hold (MOH) process. (2) any entry of PCmptbl[] matched the currently active MOH process (MOH process ID == pcbla_ptr→procnum).
Possible Error Effect:	Customers may be using the MOH feature erroneously.
Corrective Action Taken:	Cleared duplicate correct entries or purged process if no valid entries found.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCmpttbl[] table.
<i>Bad Data =</i>	A bit field which marks the duplicate/erroneous entries in the PCmpttbl ("1" bits are bad).
<i>Good Data =</i>	A count of the number of incorrect entries in the PCmpttbl[].
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum) MOH process.
<i>Dump 1 =</i>	rIPCBLA tuple for MOH process.
<i>Dump 2 =</i>	The PCmpttbl[] table.

Table 15 Error Code: GPRT

Error Code:	GPRT
Error Description:	The rDLDSL tuple attribute servclass doesn't match FC_EIS_LNK or rDLDSL tuple attribute gdport.member doesn't match rIPCBLA tuple attribute port.member. There is no GOOD DATA in the Dump Description field, attribute GDPORT is displayed.
Possible Error Effect:	Call processing messages will be lost while calls are place on a non EIS link.
Corrective Action Taken:	Purge the terminal process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rDLDSL tuple attribute servclass.
<i>Good Data =</i>	rDLDSL tuple attribute gdport.
<i>Logical Index =</i>	rIPCBLA tuple logical key.
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.
<i>Dump 3 =</i>	rDLDSL tuple.

Table 16 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	There is an inconsistency between the rIPCBLA and the rDSLOWN. The dslown field in the rIPCBLA states that a rDSLOWN tuple should exist, but we could not find one.
Possible Error Effect:	A resource is being used but no rDSLOWN tuple exists for that resource. This will cause the resource in use to be used by another user.
Corrective Action Taken:	The process owning that resource is purged, and all related data structure is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	The rIPCBLA B/D port name.
<i>Good Data =</i>	The process number in rIPCBLA.
<i>Logical Index =</i>	The key of the rIPCBLA tuple (procnum).
<i>Dump =</i>	The rIPCBLA tuple.

Table 17 Error Code: LPORT

Error Code:	LPORT
Error Description:	lport and party in PCBLA are not the same as those specified in TEDB.
Possible Error Effect:	Application code may assert, and the process cannot access to the right data.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	lport in PCBLA.
<i>Good Data =</i>	lport in TEDB.
<i>Logical Index =</i>	The key of the PCBLA tuple.
<i>Dump =</i>	The rIPCBLA tuple.
	The rDALB tuple.
	The rTEDB tuple.

Table 18 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	The logical port specified in the PCBLA does not exist in the static relation RLLPT_PT. It could be an ODD problem or lport field populated with error.
Possible Error Effect:	The process cannot transmit the message over the D-channel specified by this lport.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Logical port (D-channel).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 19 Error Code: MASTER_OWN

Error Code:	MASTER_OWN
Error Description:	For a given rIPCBLA tuple where the wide-band call attribute (rIPCBLA.wb_call) is DBYES and it is associated with one or more rIPHDB tuples, the head rIPHDB.masterpath must be DBYES; in this case, it is not.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIPCBLA or rIPHDB tuples.
Corrective Action Taken:	The rIPCBLA tuple is recovered to an idle state; related audits will be run to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	The masterpath of the head rIPHDB tuple (rIPHDB.masterpath).
<i>Good Data =</i>	The wide-band call bit in the rIPCBLA tuple (rIPCBLA.wb_call).
<i>Logical Index =</i>	The key of the rIPCBLA tuple.
<i>Dump =</i>	The rIPCBLA and head rIPHDB tuples.

Table 20 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	This process should be linked to a rRTLB tuple. The one-to-one linkage is missing or invalid.
Possible Error Effect:	Application code may assert, and the process cannot access the right data from the rRTLB.
Corrective Action Taken:	Purge this process and recover related data structures to insure continued message traffic.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	rRTLB key.
<i>Good Data =</i>	The port associated with this process.(rIPCBLA.port.member)
<i>Logical Index =</i>	The key of the rIPCBLA tuple.
<i>Dump =</i>	The rIPCBLA tuple.

Table 21 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Dynamic TEDB tuple may be missing or the key of TEDB may be bad.
Possible Error Effect:	Application code may assert, and the process cannot access to the right data from TEDB.
Corrective Action Taken:	Let TEDB audit catch this error, this is the defensive check in PCBLA audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	tedb key.
<i>Good Data =</i>	The expected TEDB key.
<i>Logical Index =</i>	The key of the PCBLA tuple.
<i>Dump 1 =</i>	The rIPCBLA tuple.

<i>Dump 2 =</i>	The rIDALB tuple.

Table 22 Error Code: MSG_SET_ID

Error Code:	MSG_SET_ID
Error Description:	The rIPCBLA tuple attribute msg_set_id doesn't match static rIEIS_DATA tuple attribute msg_set_id.
Possible Error Effect:	Incoming messages will be lost to the system because the message set identification is invalid.
Corrective Action Taken:	Update rIPCBLA tuple attribute msg_set_id with static rIEIS_DATA tuple attribute msg_set_id.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute msg_set_id.
<i>Good Data =</i>	rIEIS_DATA tuple attribute msg_set_id.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIEIS_DATA tuple.

Table 23 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	rIPCBLA tuple is used for default service, but there is no rIDSLEQUIP, rIELDSL, or rIEIS_DATA tuple.
Possible Error Effect:	CPE cannot use the subscribed default service.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	Insert the missing rIDSLEQUIP, rIDLDSL, or rIEIS_DATA tuple with rIPCBLA logical key number.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIEIS_DATA tuple.
<i>Dump 3 =</i>	rIDSLEQUIP tuple.

Table 24 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit could not obtain a pointer to the corresponding rACDPOSDB (automatic call distributor position data block) tuple that is being monitored by this monitor bridges terminal process.
Possible Error Effect:	The monitoring of this agent position may not work properly.
Corrective Action Taken:	This error is probably due to corrupt linkage between the rITEDB and rACDPOSDB tuples. The error is reported, and the related ACDPOSDB audit is scheduled to clean up any linkage errors related to this associated rACDPOSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	The value of the tedb field in the associated rITEDB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rITEDB tuple.

Table 25 Error Code: PID

Error Code:	PID
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit found that the PIDs do not

	match between this rIPCBLA tuple and its associated rACDPOSDB (automatic call distributor position data block) tuple.
Possible Error Effect:	The monitoring of this agent position may not work properly.
Corrective Action Taken:	The process represented by this rIPCBLA tuple is purged. The related ACDPOSDB audit is scheduled to clean up the associated rACDPOSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	The value of the mon_pid.procno field in the rACDPOSDB tuple.
<i>Good Data =</i>	The value of the pcbla_pid.procno field in the rIPCBLA tuple.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rACDPOSDB tuple.

Table 26 Error Code: PID_MISMATCH

Error Code:	PID_MISMATCH
Error Description:	There is an inconsistency between the rIPCBLA and the rDSLOWN PID fields.
Possible Error Effect:	A resource is being used but no rDSLOWN tuple exists for that resource. This will cause the resource in use to be used by another user.
Corrective Action Taken:	The process owning that resource is purged, and all related data structure is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	The process number in the rIPCBLA tuple.
<i>Good Data =</i>	The process number in the rDSLOWN tuple.
<i>Logical Index =</i>	The key of the rIPCBLA tuple (pid).
<i>Dump =</i>	The rIPCBLA tuple.

Table 27 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	The port name in the port field does not match that of the rPORTLA to which the rIPCBLA is linked.
Possible Error Effect:	The wrong port name will be used in peripheral control requests for hardware actions.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of port.
<i>Bad Data =</i>	Contents of port.
<i>Good Data =</i>	Port to which the rIPCBLA is validly linked.
<i>Logical Index =</i>	PCBLA number.
<i>Dump =</i>	PCBLA tuple.

Table 28 Error Code: PROC_STATE

Error Code:	PROC_STATE
Error Description:	There is an inconsistency between the ODBCALL bit and the PCBLA tag enum which indicates the state the call is in.
Possible Error Effect:	Incorrect billing, or wasted resources for the call record.
Corrective Action Taken:	The process is purged, and all related data structure is torn down. If a call record is in use, it is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	The state of the ODB call bit (0).
<i>Good Data =</i>	The non-ODB stable state the port was in.
<i>Logical Index =</i>	The key of the rIPCBLA tuple (pid).
<i>Dump =</i>	The rIPCBLA tuple.

Table 29 Error Code: QUE

Error Code:	QUE

Error Description:	The PCBLA customer state indicates it is in the talk or audible state and should be on a queue, but is not.
Possible Error Effect:	PCBLA resource remains until the customer hangs up because there is no chance of being serviced.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of pcbla.
<i>Bad Data =</i>	Queued member.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	PCBLA number.
<i>Dump 1 =</i>	PCBLA tuple.

Table 30 Error Code: RMMOD

Error Code:	RMMOD
Error Description:	The rIPCBLA tuple EIS switching module number doesn't match the global processor ID (GI_pcrd) value.
Possible Error Effect:	The switching module is configured incorrectly along with containing the incorrect data in the SM.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_sm with the GI_pcrd.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_sm.
<i>Good Data =</i>	GI_pcrd.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 31 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The B-channel and D-channel, used by this process, are not in the same DSL.
Possible Error Effect:	The message and data for call processing will be erroneously transmitted.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Logical port (D-channel).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 32 Error Code: SIMLNK

Error Code:	SIMLNK
Error Description:	The rIPCBLA tuple attribute sm_rel_ink doesn't match the static rDLDSL tuple attribute sm_rel_ink.
Possible Error Effect:	Listing service calls will be mishandled, resulting in calls not receiving automatic treatment.
Corrective Action Taken:	Update rIPCBLA tuple attribute sm_rel_ink with static rDLDSL tuple attribute sm_rel_ink.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute sm_rel_ink.
<i>Good Data =</i>	rDLDSL tuple attribute sm_rel_ink.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rDLDSL tuple.

Table 33 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value for its use in an rIPCBLA.

Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the rIORTLA changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of state field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 34 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The state field indicated that the rIPCBLA is not linked to a rIORTLA or rICPLR, but the linkage is not null.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the process changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 35 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	The state field indicated that the rIPCBLA is linked to a rIORTLA, but there is no linkage.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the portla changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 36 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The state and hh_inprog fields indicated that the rIPCBLA is linked to a rIORTLA or rICPLR, but there is no linkage.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the process changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA.

Table 37 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rIPCBLA has been in a transient customer state for at least five minutes and has not changed state, or the customer state is stable but the pcbla audit flag was left in an invalid state.

Possible Error Effect:	The port involved, is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of custat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	Process control block.

Table 38 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	An rIPCBLA, which is being used for modem pooling, has been in a transient customer state for at least five minutes and has not changed state, or the customer state is stable but the pcbla audit flag was left in an invalid state.
Possible Error Effect:	The port involved is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of custat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	Process control block.

Table 39 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit could not obtain a pointer to the corresponding rITEDB tuple representing the position to be monitored.
Possible Error Effect:	Call processing will not work correctly for this port described by this rITEDB tuple.
Corrective Action Taken:	The process represented by this rIPCBLA tuple will be purged. The related TEDB audit is scheduled to clean up problems in the RLTEDB relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	The value of the tedb field obtained from the bad rITEDB pointer.
<i>Good Data =</i>	The value of the tedb field calculated by the RTLP2VPORT macro.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 40 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUold_time, used for transient timing may have occurred.
Possible Error Effect:	The audit's transient timing has been interrupted. Lost resources may take longer to recover because transient timing must be restarted.
Corrective Action Taken:	AUold_time will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the transient timing global variable (AUold_time).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Content of the transient timing variable (AUold_time).

<i>Dump 2 =</i>	Content of the current OSDS ITIME (OSnow).

Table 41 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The state field indicates that B-channel has been moved to another process, but the linkage fields are not NULL. The PCBLA tuple should not link to PORTLA and CCBCOM at this moment.
Possible Error Effect:	This B-channel cannot be used for other calls.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

PCBLA-B**Software Release:** 5E13 only**1. PROCESSOR CONTROL BLOCK LINKAGE AREA (PCBLA) AUDIT**

Call processing and terminal and switch maintenance are implemented by means of processes. Processes are associated with data blocks called process control block linkage areas. Call processing functions are implemented by terminal processes. One terminal process is associated with each terminal (line, trunk, and so forth) in a call. A terminal process control block linkage area contains data that describes the process, links it to other entities (such as ports and time slots), and controls the call itself (such as call waiting or forwarding data).

Incorrect data in a process control block linkage area can cause a variety of problems related to call processing. For example, broken linkage can cause processes to tie up terminals indefinitely when a call does not exist; or bad feature data can cause faulty processing of features (such as call forwarding and call waiting).

The process control block linkage area audit (PCBLA) verifies data in the process control block linkage areas resident in the switching modules (SMs). Syntactic checks are performed on all process control block linkage areas (terminal, terminal maintenance, switch maintenance, and system process process control block linkage areas). Semantic checks are performed only on terminal process control block linkage areas.

This audit is part of the feature control audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

The audit runs in the routine mode in the normal shift cycle and in the elevated mode when scheduled by a user request.

2. DATA STRUCTURES

The data structure checked by the PCBLA audit is the process control block linkage area relation (RLPCBLA). Figure 1 illustrates the layout of RLPCBLA and shows the items checked by the audit. The RLPCBLA relation has owner-member linkage with the coupler relation (RLCPLR), the path data block relation (RLPHDB), and the channel control block common area relation (RLCCBCOM). It also has a one-to-one linkage with the port linkage area relation (RLPORTLA), D-channel application linkage block (RLDALB), IDCU EOC level 7 administration remote terminal embedded operation channel linkage block (RLRTLb), and call buildup data block (RLCBDB). The following linkage check is performed by the PCBLA audit due to the addition of linkage structures for RLPCBLA.

- To verify that PCBLA owns the call leg data block (CLDB) or the association process block (APB).
- To verify that PCBLA owns the IDCU remote terminal linkage block (RLRTLb) if the rIPCBLA.rtlb_flag is set.

3. AUDIT CHECKS

This audit verifies the office database (ODB) call record flag with the presence of an ODB call process.

To verify the linkages associated with the process control block linkage areas, the PCBLA audit calls DRINASEK driver once for each audit segment. After verifying the owner-member linkages of each process control block linkage area, the DRINASEK driver invokes a semantic check function that performs the following checks:

- (a) All OSDS process control blocks that are not currently in use are called dead or inactive processes. The process control block linkage areas of dead processes must not be linked to any ports or to any other data structures. They should be on an idle list. The audit determines whether a process is dead and, if it is, checks to see that its process control block linkage area is not linked to anything.
- (b) Active processes are either transient or stable, depending on the content of the customer state (custat) item

of the process's process control block linkage area (for example, dialing is transient and talking is stable). Transient processes with a customer state that has not changed in over five minutes are considered to be in error.

- (c) The link state (state) item of the process control block linkage area contains a value that indicates that the process control block linkage area is in one of two states: the process control block linkage area is linked to a port or coupler, or the process control block linkage area is not linked to a port or coupler. For stable processes, the link state item value is compared to the actual linkage.
- (d) For active processes, the port item of the process control block linkage area contains the name of the linked port or, if the process control block linkage area is linked to more than one port, the name of the first linked port. For stable processes, the audit verifies that the process control block linkage area is actually linked to the port whose name is contained in the port item or, if the process control block linkage area is linked to more than one port, that the first port is the port indicated in the port item.
- (e) If the call waiting in progress (cw_inprog) item of the process control block linkage area is false, the customer state item (custat) may contain any value except CWTALK. If the cw_inprog item is true, custat may contain any value.
- (f) If the customer state (CUSTAT) is FCC_TALK and the class of service is FC_LNQU, then a related non-idle QM_MLHG queued member must exist. Or, if the class of service is FC_OHQU, then a related, non-idle QM_OGT queued member must exist. When either of these conditions exists, a customer is connected to an announcement or tone.
- (g) If the wide-band call attribute (rIPCBLA.wb_call) is DBYES and rIPHDB tuples are linked to the rIPCBLA tuple, the head rIPHDB.masterpath must be DBYES and it must be a wide-band type of path (rIPHDB.pathtyp = PHWIDEBAND).

The linkage checks between rIPCBLA and rCBDB, rDALB are covered by the AUDrinasek() drivers. In addition to the existing semantic checks, the following checks are done to accommodate the data structure changes mentioned:

- If the process is associated with the message traffic on IDCU EOC channels, the rIPCBLA.rtlb_flag is set. When the flag is set, a linkage should exist between the rIPCBLA and rRTLb. This is checked because the same linkage is used for rDALB tuples.
- If the process associated with this rIPCBLA tuple is not existent, this rIPCBLA tuple is not linked to any other data structures including CBDB and DALB.
- If this terminal process owns a B-port, verify that the DSL containing this B-port is the same as that specified in the lport field. The lport is logical port used to condense the key for a D-port. The new static relation (RLLPT-PT) can be used for mapping a logical port to the physical port.
- The logical port is moved from one process to another when two calls need to use the same B-channel. After the logical port is removed, the RTA subsystem will relocate the physical port (PORTLA) to another process and indicate the status of linkage as DBPSHOLD. At this moment, the PCBLA tuple associated with this call-on-hold should not have linkage to PORTLA and CCBCOM tuples. The PCBLA audit verifies this by checking the *state* field (DBPSHOLD) against the corresponding linkages.
- Verify logical port and party in the PCBLA from the terminal equipment data block (TEDB) which contains the feature subscribed by the user.
- If the ACDBRCS software package is loaded and this PCBLA represents a monitor bridge terminal process, the audit will verify that the **mon_pid** field in the corresponding rIACDPOSDB tuple contains a valid PID that is represented by this PCBLA tuple.

- If the PCBLA is used for default service origination, check that the service subscribed is in the static relation Digital Subscriber Line Equipage.
- Verify odbcall bit for on-demand-packet terminal process.

If any of these checks fail, control is returned to the DRINASEK driver with an error indication.

4. ERROR HANDLING

If the process control block linkage area of a dead process is found to be linked to anything, the linkage is broken. All other errors result in purging of the process whose process control block linkage area is in error. In all cases, errors are reported by means of output messages.

The audit produces the following error codes and those generated by the DRINASEK driver.

If there is an ODB call record, but no ODB call process owning the port, the ODB call will be torn down.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing of rIPCBLA.calltag attribute. This error indicates that the FITA macro to read a call tag from the call processing bit map has returned a unknown return code.
Possible Error Effect:	Since the current status of a call tag in the call processing bit map cannot be determined, the integrity of the bit map may be compromised resulting in inaccurate notifications of transferred call clearing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of rIPCBLA.calltag.
<i>Bad Data =</i>	The return code from the FITA macro.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	For a given rIPCBLA tuple where the wide-band call attribute (rIPCBLA.wb_call) is DBYES and it is associated with one or more rIPHDB tuples, the head rIPHDB.pathtyp is not PHWIDEBAND.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIPCBLA or rIPHDB tuples.
Corrective Action Taken:	The rIPCBLA tuple is recovered to an idle state; related audits will be run to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPHDB tuple attribute pathtyp.
<i>Good Data =</i>	rIPCBLA tuple attribute wb_call.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIPHDB tuple.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The external information system link number in rIPCBLA tuple doesn't match the static rDLDSL tuple attribute xnamrl.
Possible Error Effect:	Listing services call processing isn't allowed on this EIS link.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_inlk_num with the static rDLDSL tuple attribute xnamrl.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_inlk_num.
<i>Good Data</i> =	rDLDSL tuple attribute xnamrl.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rDLDSL tuple.

Table 4 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the AU bitmap AUtouch_bits[] against call tags in the call processing bitmap Ftag_map[]. This error indicates an inconsistency between the AU bitmap and the call processing bitmap.
Possible Error Effect:	If call tags have an incorrect status in the call processing bitmap, then call processing may run out of call tags which would prevent further notifications, or call processing may provide notification of transferred call clearing using an incorrect call tag.
Corrective Action Taken:	Recover the call tag in the call processing bitmap to the value in the AU bitmap.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value of call tag.
<i>Bad Data</i> =	The value returned from the FITA macro.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	0.
<i>Dump</i> =	NA.

Table 5 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The count value stored in a MSslnnode[] SM node to node pair array entry for the 2479 seizure count measurements (MSslnnode[].MS2479ac_inscz) was found to be inaccurate. This count should be the sum of the counts for this entry calculated by the PCBLA audit after sampling all tuples in the RLPCBLA relation, where: MSslnnode[X].MS2479ac_inscz = $\sum$ MSnodeidx[OFFSET]; for OFFSET = (rIPCBLA.in_ars_idx * 2) - (rIPCBLA.data_rt - 1).
Possible Error Effect:	Incorrect 2479 seizure count measurements for offices associated with SM node-to-node pairs in an FTS2000 network.
Corrective Action Taken:	The measurement count for the MSslnnode[].MS2479ac_inscz array entry is corrected to the value calculated after all rIPCBLA tuples have been sampled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the MSslnnode[] array entry found in error.
<i>Bad Data</i> =	The contents of the MSslnnode[].MS2479ac_inscz 2479 counts.
<i>Good Data</i> =	The correct value calculated by the audit and the number of note-to-node pairs associated with the office (calculated 2479 counts << 16 Glin_nodes).
<i>Logical Index</i> =	The index of the MSslnnode[] array entry found in error.
<i>Dump</i> =	NA.

Table 6 Error Code: CSTATE

Error Code:	CSTATE

Error Description:	An rPCBLA has an invalid customer state value.
Possible Error Effect:	The process would be purged the next time it receives a stimulus.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPCBLA.
<i>Bad Data</i> =	Contents of custat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rPCBLA number.
<i>Dump</i> =	rPCBLA tuple.

Table 7 Error Code: CUSTAT

Error Code:	CUSTAT
Error Description:	The PCBLA customer state indicates an MCC call, but has no CLDBs linked to it; conversely, the PCBLA customer state indicates a POTS call, but has CLDBs linked to it.
Possible Error Effect:	The CLDBs owned by a POTS call will be lost. An MCC call would be improperly executed.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of pcbla.
<i>Bad Data</i> =	Contents of custat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	pcbla number.
<i>Dump</i> =	pcbla.

Table 8 Error Code: DDL_MSG_FLD

Error Code:	DDL_MSG_FLD
Error Description:	The rPCBLA tuple attribute eis_msg_dly doesn't match the static rEIS_DATA tuple attribute eis_msg_dly.
Possible Error Effect:	Call processing messages will be lost.
Corrective Action Taken:	Update rPCBLA tuple attribute eis_msg_dly with static rEIS_DATA tuple attribute eis_msg_dly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPCBLA tuple.
<i>Bad Data</i> =	rPCBLA tuple attribute eis_msg_dly.
<i>Good Data</i> =	rEIS_DATA tuple attribute eis_msg_dly.
<i>Logical Index</i> =	rPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.

Table 9 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The operating system believes the process associated with this rPCBLA to be dead but the linkage fields are not null.
Possible Error Effect:	The next process to use this rPCBLA may be mistakenly purged by an audit. A lost call will result.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPCBLA.
<i>Bad Data</i> =	Index of linkage map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rPCBLA number.
<i>Dump 1</i> =	rPCBLA tuple.
<i>Dump 2</i> =	Process control block.

Table 10 Error Code: DFLG

Error Code:	DFLG

Error Description:	PCBLA is used for default service but lport and lpt_type in PCBLA are not set correctly.
Possible Error Effect:	CPE cannot use the subscribed default service.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	lport in PCBLA.
<i>Good Data</i> =	lport in DSLEQUIP.
<i>Logical Index</i> =	The key of the PCBLA tuple.
<i>Dump</i> =	The rIPCBLA tuple.

Table 11 Error Code: DUP

Error Code:	DUP
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the AU bitmap AUtouch_bits[] against call tags in the call processing bitmap Fltag_map[]. This error indicates a duplicate call tag has been found.
Possible Error Effect:	If call processing were allowed to assign and use duplicate call tags, then the notifications of transferred call clearing could not be trusted to each uniquely identify a single transfer.
Corrective Action Taken:	The process will be purged and any related data structures will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA.calltag.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	rIPCBLA logical key (procnum).
<i>Dump</i> =	rIPCBLA tuple.

Table 12 Error Code: EISID

Error Code:	EISID
Error Description:	The rIPCBLA tuple attribute eis_id isn't equal to static rIDLDSL tuple attribute xnamid.
Possible Error Effect:	Listing services calls will not receive automatic treatment.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_id with static rIDLDSL tuple attribute xnamid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_id.
<i>Good Data</i> =	rIDLDSL tuple attribute xnamid.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rIDLDSL tuple.

Table 13 Error Code: EISSWID

Error Code:	EISSWID
Error Description:	The rIPCBLA tuple attribute eis_swid isn't equal to static rEIS_DATA tuple attribute eis_swid.
Possible Error Effect:	Sending the data to the wrong SM.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_swid with static rEIS_DATA tuple attribute eis_swid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_swid.
<i>Good Data</i> =	rEIS_DATA tuple attribute eis_swid.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.

Table 14 Error Code: EISTP

Error Code:	EISTP
Error Description:	The external information system terminal process number in found in rIPCBLA (procnum) doesn't match the terminal process number found in the rEIS_LNK tuple (eistp.procno).
Possible Error Effect:	The EIS link will not be placed in service to make calls. If an EIS link is in service the calls will be lost to the system.
Corrective Action Taken:	Purge the terminal process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rEIS_LNK tuple terminal process (eistp.procno).
<i>Good Data =</i>	rIPCBLA tuple terminal process (procnum).
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_LNK tuple.

Table 15 Error Code: EISVENDOR

Error Code:	EISVENDOR
Error Description:	Attribute rIPCBLA EIS vendor isn't equal to static rEIS_DATA tuple attribute eis_vendor.
Possible Error Effect:	Call processing messages will be lost while communicating with the wrong EIS vendor.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_vendor with static rEIS_DATA tuple attribute eis_vendor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_vendor.
<i>Good Data =</i>	rEIS_DATA tuple attribute eis_vendor.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.

Table 16 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The rIPCBLA attribute invalid is set to DBYES.
Possible Error Effect:	Receiving invalid call processing messages.
Corrective Action Taken:	Update rIPCBLA tuple attribute invalid with DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute invalid.
<i>Good Data =</i>	DBNO
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 17 Error Code: GPORT

Error Code:	GPORT
Error Description:	(1) currently active music on hold (MOH) process. (2) any entry of PCmtpbl[] matched the currently active MOH process (MOH process ID == pcbla_ptr→procnum).
Possible Error Effect:	Customers may be using the MOH feature erroneously.
Corrective Action Taken:	Cleared duplicate correct entries or purged process if no valid entries found.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCmtpbl[] table.
<i>Bad Data =</i>	A bit field which marks the duplicate/erroneous entries in the PCmtpbl ("1" bits

	are bad).
<i>Good Data</i> =	A count of the number of incorrect entries in the PCmptbl[].
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum) MOH process.
<i>Dump 1</i> =	rIPCBLA tuple for MOH process.
<i>Dump 2</i> =	The PCmptbl[] table.

Table 18 Error Code: GPRT

Error Code:	GPRT
Error Description:	The rIDLSL tuple attribute servclass doesn't match FC_EIS_LNK or rIDLSL tuple attribute gdport.member doesn't match rIPCBLA tuple attribute port.member. There is no GOOD DATA in the Dump Description field, attribute GDPORT is displayed.
Possible Error Effect:	Call processing messages will be lost while calls are place on a non EIS link.
Corrective Action Taken:	Purge the terminal process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIDLSL tuple attribute servclass.
<i>Good Data</i> =	rIDLSL tuple attribute gdport.
<i>Logical Index</i> =	rIPCBLA tuple logical key.
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.
<i>Dump 3</i> =	rIDLSL tuple.

Table 19 Error Code: INDRANGEBAD

Error Code:	INDRANGEBAD
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the call processing bitmap Fltag_map[]. This error indicates a call tag is out of range.
Possible Error Effect:	If call processing were allowed to use out of range call tags, then the integrity of the call tags used for notification of transferred call clearing would be compromised.
Corrective Action Taken:	The process will be purged and any related data structures will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA.calltag.
<i>Good Data</i> =	The maximum value of a call tag.
<i>Logical Index</i> =	rIPCBLA logical key (procnum).
<i>Dump</i> =	rIPCBLA tuple.

Table 20 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	There is an inconsistency between the rIPCBLA and the rIDSLOWN. The dslow field in the rIPCBLA states that a rIDSLOWN tuple should exist, but we could not find one.
Possible Error Effect:	A resource is being used but no rIDSLOWN tuple exists for that resource. This will cause the resource in use to be used by another user.
Corrective Action Taken:	The process owning that resource is purged, and all related data structure is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	The rIPCBLA B/D port name.
<i>Good Data</i> =	The process number in rIPCBLA.
<i>Logical Index</i> =	The key of the rIPCBLA tuple (procnum).
<i>Dump</i> =	The rIPCBLA tuple.

Table 21 Error Code: LPORT

Error Code:	LPORT
Error Description:	lport and party in PCBLA are not the same as those specified in TEDB.
Possible Error Effect:	Application code may assert, and the process cannot access to the right data.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	lport in PCBLA.
<i>Good Data</i> =	lport in TEDB.
<i>Logical Index</i> =	The key of the PCBLA tuple.
<i>Dump</i> =	The rIPCBLA tuple. The rIDALB tuple. The rITEDB tuple.

Table 22 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	The logical port specified in the PCBLA does not exist in the static relation RLLPT_PT. It could be an ODD problem or lport field populated with error.
Possible Error Effect:	The process cannot transmit the message over the D-channel specified by this lport.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Logical port (D-channel).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 23 Error Code: MASTER_OWN

Error Code:	MASTER_OWN
Error Description:	For a given rIPCBLA tuple where the wide-band call attribute (rIPCBLA.wb_call) is DBYES and it is associated with one or more rIPHDB tuples, the head rIPHDB.masterpath must be DBYES; in this case, it is not.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIPCBLA or rIPHDB tuples.
Corrective Action Taken:	The rIPCBLA tuple is recovered to an idle state; related audits will be run to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	The masterpath of the head rIPHDB tuple (rIPHDB.masterpath).
<i>Good Data</i> =	The wide-band call bit in the rIPCBLA tuple (rIPCBLA.wb_call).
<i>Logical Index</i> =	The key of the rIPCBLA tuple.
<i>Dump</i> =	The rIPCBLA and head rIPHDB tuples.

Table 24 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	This process should be linked to a rRTLb tuple. The one-to-one linkage is missing or invalid.
Possible Error Effect:	Application code may assert, and the process cannot access the right data from the rRTLb.
Corrective Action Taken:	Purge this process and recover related data structures to insure continued message traffic.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	rRTLb key.
<i>Good Data</i> =	The port associated with this process.(rIPCBLA.port.member)
<i>Logical Index</i> =	The key of the rIPCBLA tuple.

<i>Dump =</i>	The rIPCBLA tuple.

Table 25 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Dynamic TEDB tuple may be missing or the key of TEDB may be bad.
Possible Error Effect:	Application code may assert, and the process cannot access to the right data from TEDB.
Corrective Action Taken:	Let TEDB audit catch this error, this is the defensive check in PCBLA audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	tedb key.
<i>Good Data =</i>	The expected TEDB key.
<i>Logical Index =</i>	The key of the PCBLA tuple.
<i>Dump 1 =</i>	The rIPCBLA tuple.
<i>Dump 2 =</i>	The rDALB tuple.

Table 26 Error Code: MSG_SET_ID

Error Code:	MSG_SET_ID
Error Description:	The rIPCBLA tuple attribute msg_set_id doesn't match static rEIS_DATA tuple attribute msg_set_id.
Possible Error Effect:	Incoming messages will be lost to the system because the message set identification is invalid.
Corrective Action Taken:	Update rIPCBLA tuple attribute msg_set_id with static rEIS_DATA tuple attribute msg_set_id.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute msg_set_id.
<i>Good Data =</i>	rEIS_DATA tuple attribute msg_set_id.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.

Table 27 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	rIPCBLA tuple is used for default service, but there is no rIDSLEQUIP, rELDSL, or rEIS_DATA tuple.
Possible Error Effect:	CPE cannot use the subscribed default service.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	Insert the missing rIDSLEQUIP, rDLDSL, or rEIS_DATA tuple with rIPCBLA logical key number.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.
<i>Dump 3 =</i>	rIDSLEQUIP tuple.

Table 28 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit could not obtain a pointer to the corresponding rACDPOSDB (automatic call distributor position data block) tuple that is being monitored by this monitor bridges terminal process.
Possible Error Effect:	The monitoring of this agent position may not work properly.
Corrective Action Taken:	This error is probably due to corrupt linkage between the rTEDB and rACDPOSDB tuples. The error is reported, and the related ACDPOSDB audit is

	scheduled to clean up any linkage errors related to this associated rIACDPOSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	The value of the tedb field in the associated rITEDB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rITEDB tuple.

Table 29 Error Code: PID

Error Code:	PID
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit found that the PIDs do not match between this rIPCBLA tuple and its associated rIACDPOSDB (automatic call distributor position data block) tuple.
Possible Error Effect:	The monitoring of this agent position may not work properly.
Corrective Action Taken:	The process represented by this rIPCBLA tuple is purged. The related ACDBRCS audit is scheduled to clean up the associated rIACDPOSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	The value of the mon_pid.procno field in the rIACDPOSDB tuple.
<i>Good Data =</i>	The value of the pcbla_pid.procno field in the rIPCBLA tuple.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIACDPOSDB tuple.

Table 30 Error Code: PID_MISMATCH

Error Code:	PID_MISMATCH
Error Description:	There is an inconsistency between the rIPCBLA and the rIDSLOWN PID fields.
Possible Error Effect:	A resource is being used but no rIDSLOWN tuple exists for that resource. This will cause the resource in use to be used by another user.
Corrective Action Taken:	The process owning that resource is purged, and all related data structure is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	The process number in the rIPCBLA tuple.
<i>Good Data =</i>	The process number in the rIDSLOWN tuple.
<i>Logical Index =</i>	The key of the rIPCBLA tuple (pid).
<i>Dump =</i>	The rIPCBLA tuple.

Table 31 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	The port name in the port field does not match that of the rIORTLA to which the rIPCBLA is linked.
Possible Error Effect:	The wrong port name will be used in peripheral control requests for hardware actions.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of port.
<i>Bad Data =</i>	Contents of port.
<i>Good Data =</i>	Port to which the rIPCBLA is validly linked.
<i>Logical Index =</i>	PCBLA number.
<i>Dump =</i>	PCBLA tuple.

Table 32 Error Code: PROC_STATE

Error Code:	PROC_STATE

Error Description:	There is an inconsistency between the ODBCALL bit and the PCBLA tag enum which indicates the state the call is in.
Possible Error Effect:	Incorrect billing, or wasted resources for the call record.
Corrective Action Taken:	The process is purged, and all related data structure is torn down. If a call record is in use, it is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	The state of the ODB call bit (0).
<i>Good Data</i> =	The non-ODB stable state the port was in.
<i>Logical Index</i> =	The key of the rIPCBLA tuple (pid).
<i>Dump</i> =	The rIPCBLA tuple.

Table 33 Error Code: QUE

Error Code:	QUE
Error Description:	The PCBLA customer state indicates it is in the talk or audible state and should be on a queue, but is not.
Possible Error Effect:	PCBLA resource remains until the customer hangs up because there is no chance of being serviced.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of pcbla.
<i>Bad Data</i> =	Queued member.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	PCBLA number.
<i>Dump 1</i> =	PCBLA tuple.

Table 34 Error Code: RMMOD

Error Code:	RMMOD
Error Description:	The rIPCBLA tuple EIS switching module number doesn't match the global processor ID (GI_pcrd) value.
Possible Error Effect:	The switching module is configured incorrectly along with containing the incorrect data in the SM.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_sm with the GI_pcrd.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_sm.
<i>Good Data</i> =	GI_pcrd.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump</i> =	rIPCBLA tuple.

Table 35 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The B-channel and D-channel, used by this process, are not in the same DSL.
Possible Error Effect:	The message and data for call processing will be erroneously transmitted.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Logical port (D-channel).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 36 Error Code: SIMLNK

Error Code:	SIMLNK
Error Description:	The rIPCBLA tuple attribute sm_rel_lnk doesn't match the static rLDLDSL tuple attribute sm_rel_lnk.
Possible Error Effect:	Listing service calls will be mishandled, resulting in calls not receiving automatic

	treatment.
Corrective Action Taken:	Update rIPCBLA tuple attribute sm_rel_lnk with static rIDLSL tuple attribute sm_rel_lnk.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute sm_rel_lnk.
<i>Good Data =</i>	rIDLSL tuple attribute sm_rel_lnk.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIDLSL tuple.

Table 37 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value for its use in an rIPCBLA.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the rPORTLA changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of state field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 38 Error Code: STATE_IDL

Error Code:	STATE IDL
Error Description:	The state field indicated that the rIPCBLA is not linked to a rPORTLA or rICPLR, but the linkage is not null.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the process changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 39 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	The state field indicated that the rIPCBLA is linked to a rPORTLA, but there is no linkage.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the portla changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 40 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The state and hh_inprog fields indicated that the rIPCBLA is linked to a rPORTLA or rICPLR, but there is no linkage.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the

	linkage when the process changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA.

Table 41 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rIPCBLA has been in a transient customer state for at least five minutes and has not changed state, or the customer state is stable but the pcbla audit flag was left in an invalid state.
Possible Error Effect:	The port involved, is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of custat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	Process control block.

Table 42 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	An rIPCBLA, which is being used for modem pooling, has been in a transient customer state for at least five minutes and has not changed state, or the customer state is stable but the pcbla audit flag was left in an invalid state.
Possible Error Effect:	The port involved is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	Contents of custat.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	Process control block.

Table 43 Error Code: TAG

Error Code:	TAG
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the AU bitmap AUtouch_bits[] against call tags in the call processing bitmap Fltag_map[]. This error indicates a call tag in the call processing bitmap is not set but should be.
Possible Error Effect:	If the call tag is not marked as busy in the call processing bit map, then call processing would not provide notification of transferred call clearing for that transfer since its call tag is marked as idle. Also, call processing may attempt to assign the call tag to another transfer resulting in a duplicate call tag and loss of integrity.
Corrective Action Taken:	AU will recover the call tag by setting it to busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of rIPCBLA.calltag.
<i>Bad Data =</i>	The value returned from the FITA macro.
<i>Good Data =</i>	The value of a busy tag (ISBUSY_TAG = 1).

<i>Logical Index</i> =	rIPCBLA logical key (procnum).
<i>Dump</i> =	rIPCBLA tuple.

Table 44 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit could not obtain a pointer to the corresponding rITEDB tuple representing the position to be monitored.
Possible Error Effect:	Call processing will not work correctly for this port described by this rITEDB tuple.
Corrective Action Taken:	The process represented by this rIPCBLA tuple will be purged. The related TEDB audit is scheduled to clean up problems in the RLTEDB relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	The value of the tedb field obtained from the bad rITEDB pointer.
<i>Good Data</i> =	The value of the tedb field calculated by the RTLP2VPORT macro.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 45 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUold_time, used for transient timing may have occurred.
Possible Error Effect:	The audit's transient timing has been interrupted. Lost resources may take longer to recover because transient timing must be restarted.
Corrective Action Taken:	AUold_time will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the transient timing global variable (AUold_time).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Content of the transient timing variable (AUold_time).
<i>Dump 2</i> =	Content of the current OSDS ITIME (OSnow).

Table 46 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The state field indicates that B-channel has been moved to another process, but the linkage fields are not NULL. The PCBLA tuple should not link to PORTLA and CCBCOM at this moment.
Possible Error Effect:	This B-channel cannot be used for other calls.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

PCBLA-C

Software Release: 5E14 and later

1. PROCESSOR CONTROL BLOCK LINKAGE AREA (PCBLA) AUDIT

Call processing and terminal and switch maintenance are implemented by means of processes. Processes are associated with data blocks called process control block linkage areas. Call processing functions are implemented by terminal processes. One terminal process is associated with each terminal (line, trunk, and so forth) in a call. A terminal process control block linkage area contains data that describes the process, links it to other entities (such as ports and time slots), and controls the call itself (such as call waiting or forwarding data).

Incorrect data in a process control block linkage area can cause a variety of problems related to call processing. For example, broken linkage can cause processes to tie up terminals indefinitely when a call does not exist; or bad feature data can cause faulty processing of features (such as call forwarding and call waiting).

The process control block linkage area audit (PCBLA) verifies data in the process control block linkage areas resident in the switching modules (SMs). Syntactic checks are performed on all process control block linkage areas (terminal, terminal maintenance, switch maintenance, and system process process control block linkage areas). Semantic checks are performed only on terminal process control block linkage areas.

This audit is part of the feature control audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

The audit runs in the routine mode in the normal shift cycle and in the elevated mode when scheduled by a user request.

2. DATA STRUCTURES

The data structure checked by the PCBLA audit is the process control block linkage area relation (RLPCBLA). Figure 1 illustrates the layout of RLPCBLA and shows the items checked by the audit. The RLPCBLA relation has owner-member linkage with the coupler relation (RLCPLR), the path data block relation (RLPHDB), and the channel control block common area relation (RLCCBCOM). It also has a one-to-one linkage with the port linkage area relation (RLPORTLA), D-channel application linkage block (RLDALB), IDCU EOC level 7 administration remote terminal embedded operation channel linkage block (RLRTLb), and call buildup data block (RLCBDB). The following linkage check is performed by the PCBLA audit due to the addition of linkage structures for RLPCBLA.

- To verify that PCBLA owns the call leg data block (CLDB) or the association process block (APB).
- To verify that PCBLA owns the IDCU remote terminal linkage block (RLRTLb) if the rIPCBLA.rtlb_flag is set.

rPCBLA

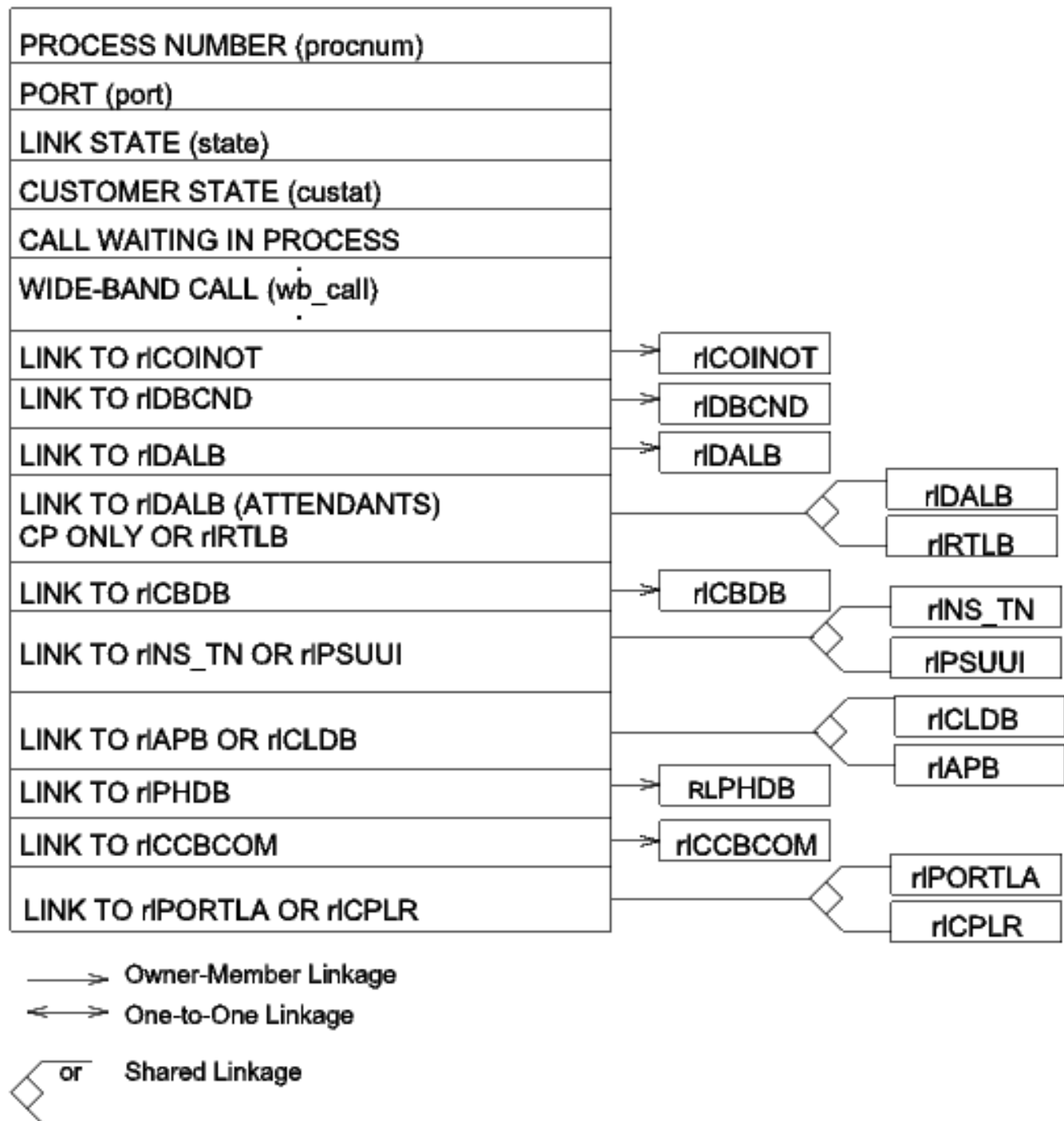


Figure 1 (PCBLA-1) Process Control Block Linkage Area

3. AUDIT CHECKS

This audit verifies the office database (ODB) call record flag with the presence of an ODB call process.

To verify the linkages associated with the process control block linkage areas, the PCBLA audit calls DRINASEK

driver once for each audit segment. After verifying the owner-member linkages of each process control block linkage area, the DRINASEK driver invokes a semantic check function that performs the following checks:

- (a) All OSDS process control blocks that are not currently in use are called dead or inactive processes. The process control block linkage areas of dead processes must not be linked to any ports or to any other data structures. They should be on an idle list. The audit determines whether a process is dead and, if it is, checks to see that its process control block linkage area is not linked to anything.
- (b) Active processes are either transient or stable, depending on the content of the customer state (custat) item of the process's process control block linkage area (for example, dialing is transient and talking is stable). Transient processes with a customer state that has not changed in over five minutes are considered to be in error.
- (c) The link state (state) item of the process control block linkage area contains a value that indicates that the process control block linkage area is in one of two states: the process control block linkage area is linked to a port or coupler, or the process control block linkage area is not linked to a port or coupler. For stable processes, the link state item value is compared to the actual linkage.
- (d) For active processes, the port item of the process control block linkage area contains the name of the linked port or, if the process control block linkage area is linked to more than one port, the name of the first linked port. For stable processes, the audit verifies that the process control block linkage area is actually linked to the port whose name is contained in the port item or, if the process control block linkage area is linked to more than one port, that the first port is the port indicated in the port item.
- (e) If the call waiting in progress (cw_inprog) item of the process control block linkage area is false, the customer state item (custat) may contain any value except CWTALK. If the cw_inprog item is true, custat may contain any value.
- (f) If the customer state (CUSTAT) is FCC_TALK and the class of service is FC_LNQU, then a related non-idle QM_MLHG queued member must exist. Or, if the class of service is FC_OHQU, then a related, non-idle QM_OGT queued member must exist. When either of these conditions exists, a customer is connected to an announcement or tone.
- (g) If the wide-band call attribute (rIPCBLA.wb_call) is DBYES and rIPHDB tuples are linked to the rIPCBLA tuple, the head rIPHDB.masterpath must be DBYES and it must be a wide-band type of path (rIPHDB.pathtyp = PHWIDEBAND).

The linkage checks between rIPCBLA and rCBDB, rDALB are covered by the AUdrinasek() drivers. In addition to the existing semantic checks, the following checks are done to accommodate the data structure changes mentioned:

- If the process is associated with the message traffic on IDCU EOC channels, the rIPCBLA.rtlb_flag is set. When the flag is set, a linkage should exist between the rIPCBLA and rRTLb. This is checked because the same linkage is used for rDALB tuples.
- If the process associated with this rIPCBLA tuple is not existent, this rIPCBLA tuple is not linked to any other data structures including CBDB and DALB.
- If this terminal process owns a B-port, verify that the DSL containing this B-port is the same as that specified in the lport field. The lport is logical port used to condense the key for a D-port. The new static relation (RLLPT-PT) can be used for mapping a logical port to the physical port.
- The logical port is moved from one process to another when two calls need to use the same B-channel. After the logical port is removed, the RTA subsystem will relocate the physical port (PORTLA) to another process and indicate the status of linkage as DBPSHOLD. At this moment, the PCBLA tuple associated with this

call-on-hold should not have linkage to PORTLA and CCBCOM tuples. The PCBLA audit verifies this by checking the *state* field (DBPSHOLD) against the corresponding linkages.

- Verify logical port and party in the PCBLA from the terminal equipment data block (TEDB) which contains the feature subscribed by the user.
- If the ACDBRCS software package is loaded and this PCBLA represents a monitor bridge terminal process, the audit will verify that the **mon_pid** field in the corresponding rIACDPOSDB tuple contains a valid PID that is represented by this PCBLA tuple.
- If the PCBLA is used for default service origination, check that the service subscribed is in the static relation Digital Subscriber Line Equipage.
- Verify odbcall bit for on-demand-packet terminal process.

If any of these checks fail, control is returned to the DRINASEK driver with an error indication.

4. ERROR HANDLING

If the process control block linkage area of a dead process is found to be linked to anything, the linkage is broken. All other errors result in purging of the process whose process control block linkage area is in error. In all cases, errors are reported by means of output messages.

The audit produces the following error codes and those generated by the DRINASEK driver.

If there is an ODB call record, but no ODB call process owning the port, the ODB call will be torn down.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing of rIPCBLA.calltag attribute. This error indicates that the FITA macro to read a call tag from the call processing bit map has returned an unknown return code.
Possible Error Effect:	Since the current status of a call tag in the call processing bit map cannot be determined, the integrity of the bit map may be compromised resulting in inaccurate notifications of transferred call clearing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of rIPCBLA.calltag.
<i>Bad Data =</i>	The return code from the FITA macro.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	For a given rIPCBLA tuple where the wide-band call attribute (rIPCBLA.wb_call) is DBYES and it is associated with one or more rIPHDB tuples, the head rIPHDB.pathtyp is not PHWIDEBAND.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIPCBLA or rIPHDB

	tuples.
Corrective Action Taken:	The rIPCBLA tuple is recovered to an idle state; related audits will be run to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPHDB tuple attribute pathtyp.
<i>Good Data =</i>	rIPCBLA tuple attribute wb_call.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rIPHDB tuple.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The external information system link number in rIPCBLA tuple doesn't match the static rDLDSL tuple attribute xnamrl.
Possible Error Effect:	Listing services call processing isn't allowed on this EIS link.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_ink_num with the static rDLDSL tuple attribute xnamrl.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_ink_num.
<i>Good Data =</i>	rDLDSL tuple attribute xnamrl.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rDLDSL tuple.

Table 4 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the AU bitmap AUtouch_bits[] against call tags in the call processing bitmap Fltag_map[]. This error indicates an inconsistency between the AU bitmap and the call processing bitmap.
Possible Error Effect:	If call tags have an incorrect status in the call processing bitmap, then call processing may run out of call tags which would prevent further notifications, or call processing may provide notification of transferred call clearing using an incorrect call tag.
Corrective Action Taken:	Recover the call tag in the call processing bitmap to the value in the AU bitmap.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of call tag.
<i>Bad Data =</i>	The value returned from the FITA macro.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	0.
<i>Dump =</i>	NA.

Table 5 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The count value stored in a MSslnnode[] SM node to node pair array entry for the 2479 seizure count measurements (MSslnnode[].MS2479ac_Inscz) was found to be inaccurate. This count should be the sum of the counts for this entry calculated by the PCBLA audit after sampling all tuples in the RLPCBLA relation, where: $MSslnnode[X].MS2479ac\_Inscz = \sum MSnodeidx[OFFSET]$; for $OFFSET = (rIPCBLA.in\_ars\_idx * 2) - (rIPCBLA.data\_rt - 1)$.
Possible Error Effect:	Incorrect 2479 seizure count measurements for offices associated with SM node-to-node pairs in an FTS2000 network.
Corrective Action Taken:	The measurement count for the MSslnnode[].MS2479ac_Inscz array entry is corrected to the value calculated after all rIPCBLA tuples have been sampled.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the MSslnnode[] array entry found in error.
<i>Bad Data</i> =	The contents of the MSslnnode[].MS2479ac_inscz 2479 counts.
<i>Good Data</i> =	The correct value calculated by the audit and the number of note-to-node pairs associated with the office (calculated 2479 counts << 16 Glnn_nodes).
<i>Logical Index</i> =	The index of the MSslnnode[] array entry found in error.
<i>Dump</i> =	NA.

Table 6 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	An rIPCBLA has an invalid customer state value.
Possible Error Effect:	The process would be purged the next time it receives a stimulus.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Contents of custat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 7 Error Code: CUSTAT

Error Code:	CUSTAT
Error Description:	The PCBLA customer state indicates an MCC call, but has no CLDBs linked to it; conversely, the PCBLA customer state indicates a POTS call, but has CLDBs linked to it.
Possible Error Effect:	The CLDBs owned by a POTS call will be lost. An MCC call would be improperly executed.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of pcbla.
<i>Bad Data</i> =	Contents of custat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	pcbla number.
<i>Dump</i> =	pcbla.

Table 8 Error Code: DDL_MSG_FLD

Error Code:	DDL_MSG_FLD
Error Description:	The rIPCBLA tuple attribute eis_msg_dly doesn't match the static rEIS_DATA tuple attribute eis_msg_dly.
Possible Error Effect:	Call processing messages will be lost.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_msg_dly with static rEIS_DATA tuple attribute eis_msg_dly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_msg_dly.
<i>Good Data</i> =	rEIS_DATA tuple attribute eis_msg_dly.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.

Table 9 Error Code: DEAD_LNKD

Error Code:	DEAD_LNKD
Error Description:	The operating system believes the process associated with this rIPCBLA to be dead but the linkage fields are not null.
Possible Error Effect:	The next process to use this rIPCBLA may be mistakenly purged by an audit. A lost call will result.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Index of linkage map.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	Process control block.

Table 10 Error Code: DFLG

Error Code:	DFLG
Error Description:	PCBLA is used for default service but lport and lpt_type in PCBLA are not set correctly.
Possible Error Effect:	CPE cannot use the subscribed default service.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	lport in PCBLA.
<i>Good Data</i> =	lport in DSLEQUIP.
<i>Logical Index</i> =	The key of the PCBLA tuple.
<i>Dump</i> =	The rIPCBLA tuple.

Table 11 Error Code: DUP

Error Code:	DUP
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the AU bitmap AUtouch_bits[] against call tags in the call processing bitmap Ftag_map[]. This error indicates a duplicate call tag has been found.
Possible Error Effect:	If call processing were allowed to assign and use duplicate call tags, then the notifications of transferred call clearing could not be trusted to each uniquely identify a single transfer.
Corrective Action Taken:	The process will be purged and any related data structures will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA.calltag.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	rIPCBLA logical key (procnum).
<i>Dump</i> =	rIPCBLA tuple.

Table 12 Error Code: EISID

Error Code:	EISID
Error Description:	The rIPCBLA tuple attribute eis_id isn't equal to static rIDLDSL tuple attribute xnamid.
Possible Error Effect:	Listing services calls will not receive automatic treatment.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_id with static rIDLDSL tuple attribute xnamid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_id.
<i>Good Data</i> =	rIDLDSL tuple attribute xnamid.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rIDLDSL tuple.

Table 13 Error Code: EISSWID

Error Code:	EISSWID
Error Description:	The rIPCBLA tuple attribute eis_swid isn't equal to static rEIS_DATA tuple attribute eis_swid.
Possible Error Effect:	Sending the data to the wrong SM.

Corrective Action Taken:	Update rIPCBLA tuple attribute eis_swid with static rEIS_DATA tuple attribute eis_swid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_swid.
<i>Good Data =</i>	rEIS_DATA tuple attribute eis_swid.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.

Table 14 Error Code: EISTP

Error Code:	EISTP
Error Description:	The external information system terminal process number in found in rIPCBLA (procnum) doesn't match the terminal process number found in the rEIS_LNK tuple (eistp.procno).
Possible Error Effect:	The EIS link will not be placed in service to make calls. If an EIS link is in service the calls will be lost to the system.
Corrective Action Taken:	Purge the terminal process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rEIS_LNK tuple terminal process (eistp.procno).
<i>Good Data =</i>	rIPCBLA tuple terminal process (procnum).
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_LNK tuple.

Table 15 Error Code: EISVENDOR

Error Code:	EISVENDOR
Error Description:	Attribute rIPCBLA EIS vendor isn't equal to static rEIS_DATA tuple attribute eis_vendor.
Possible Error Effect:	Call processing messages will be lost while communicating with the wrong EIS vendor.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_vendor with static rEIS_DATA tuple attribute eis_vendor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute eis_vendor.
<i>Good Data =</i>	rEIS_DATA tuple attribute eis_vendor.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.

Table 16 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The rIPCBLA attribute invalid is set to DBYES.
Possible Error Effect:	Receiving invalid call processing messages.
Corrective Action Taken:	Update rIPCBLA tuple attribute invalid with DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA tuple.
<i>Bad Data =</i>	rIPCBLA tuple attribute invalid.
<i>Good Data =</i>	DBNO
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 17 Error Code: GPORT

Error Code:	GPORT
Error Description:	

	(1) currently active music on hold (MOH) process.
	(2) any entry of PCmpttbl[] matched the currently active MOH process (MOH process ID == pcbla_ptr→procnum).
Possible Error Effect:	Customers may be using the MOH feature erroneously.
Corrective Action Taken:	Cleared duplicate correct entries or purged process if no valid entries found.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the PCmpttbl[] table.
<i>Bad Data =</i>	A bit field which marks the duplicate/erroneous entries in the PCmpttbl ("1" bits are bad).
<i>Good Data =</i>	A count of the number of incorrect entries in the PCmpttbl[].
<i>Logical Index =</i>	rPCBLA tuple logical key (procnum) MOH process.
<i>Dump 1 =</i>	rPCBLA tuple for MOH process.
<i>Dump 2 =</i>	The PCmpttbl[] table.

Table 18 Error Code: GPRT

Error Code:	GPRT
Error Description:	The rDLDSL tuple attribute servclass doesn't match FC_EIS_LNK or rDLDSL tuple attribute gdport.member doesn't match rPCBLA tuple attribute port.member. There is no GOOD DATA in the Dump Description field, attribute GDPORT is displayed.
Possible Error Effect:	Call processing messages will be lost while calls are place on a non EIS link.
Corrective Action Taken:	Purge the terminal process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPCBLA tuple.
<i>Bad Data =</i>	rDLDSL tuple attribute servclass.
<i>Good Data =</i>	rDLDSL tuple attribute gdport.
<i>Logical Index =</i>	rPCBLA tuple logical key.
<i>Dump 1 =</i>	rPCBLA tuple.
<i>Dump 2 =</i>	rEIS_DATA tuple.
<i>Dump 3 =</i>	rDLDSL tuple.

Table 19 Error Code: INDRANGE_BAD

Error Code:	INDRANGE_BAD
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the call processing bitmap Fltag_map[]. This error indicates a call tag is out of range.
Possible Error Effect:	If call processing were allowed to use out of range call tags, then the integrity of the call tags used for notification of transferred call clearing would be compromised.
Corrective Action Taken:	The process will be purged and any related data structures will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rPCBLA tuple.
<i>Bad Data =</i>	rPCBLA.calltag.
<i>Good Data =</i>	The maximum value of a call tag.
<i>Logical Index =</i>	rPCBLA logical key (procnum).
<i>Dump =</i>	rPCBLA tuple.

Table 20 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	There is an inconsistency between the rPCBLA and the rDSLOWN. The dslown field in the rPCBLA states that a rDSLOWN tuple should exist, but we could not find one.
Possible Error Effect:	A resource is being used but no rDSLOWN tuple exists for that resource. This will cause the resource in use to be used by another user.
Corrective Action Taken:	The process owning that resource is purged, and all related data structure is

	torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	The rIPCBLA B/D port name.
<i>Good Data</i> =	The process number in rIPCBLA.
<i>Logical Index</i> =	The key of the rIPCBLA tuple (procnum).
<i>Dump</i> =	The rIPCBLA tuple.

Table 21 Error Code: LPORT

Error Code:	LPORT
Error Description:	lport and party in PCBLA are not the same as those specified in TEDB.
Possible Error Effect:	Application code may assert, and the process cannot access to the right data.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	lport in PCBLA.
<i>Good Data</i> =	lport in TEDB.
<i>Logical Index</i> =	The key of the PCBLA tuple.
<i>Dump</i> =	The rIPCBLA tuple.
	The rIDALB tuple.
	The rITEDB tuple.

Table 22 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	The logical port specified in the PCBLA does not exist in the static relation RLLPT_PT. It could be an ODD problem or lport field populated with error.
Possible Error Effect:	The process cannot transmit the message over the D-channel specified by this lport.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Logical port (D-channel).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 23 Error Code: MASTER_OWN

Error Code:	MASTER_OWN
Error Description:	For a given rIPCBLA tuple where the wide-band call attribute (rIPCBLA.wb_call) is DBYES and it is associated with one or more rIPHDB tuples, the head rIPHDB.masterpath must be DBYES; in this case, it is not.
Possible Error Effect:	Failures or asserts when call processing attempts to use the rIPCBLA or rIPHDB tuples.
Corrective Action Taken:	The rIPCBLA tuple is recovered to an idle state; related audits will be run to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	The masterpath of the head rIPHDB tuple (rIPHDB.masterpath).
<i>Good Data</i> =	The wide-band call bit in the rIPCBLA tuple (rIPCBLA.wb_call).
<i>Logical Index</i> =	The key of the rIPCBLA tuple.
<i>Dump</i> =	The rIPCBLA and head rIPHDB tuples.

Table 24 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	This process should be linked to a rIRTLB tuple. The one-to-one linkage is

	missing or invalid.
Possible Error Effect:	Application code may assert, and the process cannot access the right data from the rRTLb.
Corrective Action Taken:	Purge this process and recover related data structures to insure continued message traffic.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	rRTLb key.
<i>Good Data</i> =	The port associated with this process.(rIPCBLA.port.member)
<i>Logical Index</i> =	The key of the rIPCBLA tuple.
<i>Dump</i> =	The rIPCBLA tuple.

Table 25 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	Dynamic TEDB tuple may be missing or the key of TEDB may be bad.
Possible Error Effect:	Application code may assert, and the process cannot access to the right data from TEDB.
Corrective Action Taken:	Let TEDB audit catch this error, this is the defensive check in PCBLA audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	tedb key.
<i>Good Data</i> =	The expected TEDB key.
<i>Logical Index</i> =	The key of the PCBLA tuple.
<i>Dump 1</i> =	The rIPCBLA tuple.
<i>Dump 2</i> =	The rDALB tuple.

Table 26 Error Code: MSG_SET_ID

Error Code:	MSG_SET_ID
Error Description:	The rIPCBLA tuple attribute msg_set_id doesn't match static rEIS_DATA tuple attribute msg_set_id.
Possible Error Effect:	Incoming messages will be lost to the system because the message set identification is invalid.
Corrective Action Taken:	Update rIPCBLA tuple attribute msg_set_id with static rEIS_DATA tuple attribute msg_set_id.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute msg_set_id.
<i>Good Data</i> =	rEIS_DATA tuple attribute msg_set_id.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.

Table 27 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	rIPCBLA tuple is used for default service, but there is no rIDSLEQUIP, rELDSL, or rEIS_DATA tuple.
Possible Error Effect:	CPE cannot use the subscribed default service.
Corrective Action Taken:	The process will be purged.
Manual Action Required:	Insert the missing rIDSLEQUIP, rDLDSL, or rEIS_DATA tuple with rIPCBLA logical key number.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rEIS_DATA tuple.
<i>Dump 3</i> =	rIDSLEQUIP tuple.

Table 28 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit could not obtain a pointer to the corresponding rACDPOSDB (automatic call distributor position data block) tuple that is being monitored by this monitor bridges terminal process.
Possible Error Effect:	The monitoring of this agent position may not work properly.
Corrective Action Taken:	This error is probably due to corrupt linkage between the rITEDB and rACDPOSDB tuples. The error is reported, and the related ACDPOSDB audit is scheduled to clean up any linkage errors related to this associated rACDPOSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	The value of the tedb field in the associated rITEDB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rITEDB tuple.

Table 29 Error Code: PID

Error Code:	PID
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit found that the PIDs do not match between this rIPCBLA tuple and its associated rACDPOSDB (automatic call distributor position data block) tuple.
Possible Error Effect:	The monitoring of this agent position may not work properly.
Corrective Action Taken:	The process represented by this rIPCBLA tuple is purged. The related ACDPOSDB audit is scheduled to clean up the associated rACDPOSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA tuple.
<i>Bad Data =</i>	The value of the mon_pid.procno field in the rACDPOSDB tuple.
<i>Good Data =</i>	The value of the pcbla_pid.procno field in the rIPCBLA tuple.
<i>Logical Index =</i>	rIPCBLA tuple logical key (procnum).
<i>Dump 1 =</i>	rIPCBLA tuple.
<i>Dump 2 =</i>	rACDPOSDB tuple.

Table 30 Error Code: PID_MISMATCH

Error Code:	PID_MISMATCH
Error Description:	There is an inconsistency between the rIPCBLA and the rDSLOWN PID fields.
Possible Error Effect:	A resource is being used but no rDSLOWN tuple exists for that resource. This will cause the resource in use to be used by another user.
Corrective Action Taken:	The process owning that resource is purged, and all related data structure is torn down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPCBLA where the error occurred.
<i>Bad Data =</i>	The process number in the rIPCBLA tuple.
<i>Good Data =</i>	The process number in the rDSLOWN tuple.
<i>Logical Index =</i>	The key of the rIPCBLA tuple (pid).
<i>Dump =</i>	The rIPCBLA tuple.

Table 31 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	The port name in the port field does not match that of the rPORTLA to which the rIPCBLA is linked.
Possible Error Effect:	The wrong port name will be used in peripheral control requests for hardware actions.

Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of port.
<i>Bad Data</i> =	Contents of port.
<i>Good Data</i> =	Port to which the rIPCBLA is validly linked.
<i>Logical Index</i> =	PCBLA number.
<i>Dump</i> =	PCBLA tuple.

Table 32 Error Code: PROC_STATE

Error Code:	PROC_STATE
Error Description:	There is an inconsistency between the ODBCALL bit and the PCBLA tag enum which indicates the state the call is in.
Possible Error Effect:	Incorrect billing, or wasted resources for the call record.
Corrective Action Taken:	The process is purged, and all related data structure is torn down. If a call record is in use, it is idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPCBLA where the error occurred.
<i>Bad Data</i> =	The state of the ODB call bit (0).
<i>Good Data</i> =	The non-ODB stable state the port was in.
<i>Logical Index</i> =	The key of the rIPCBLA tuple (pid).
<i>Dump</i> =	The rIPCBLA tuple.

Table 33 Error Code: QUE

Error Code:	QUE
Error Description:	The PCBLA customer state indicates it is in the talk or audible state and should be on a queue, but is not.
Possible Error Effect:	PCBLA resource remains until the customer hangs up because there is no chance of being serviced.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of pcbla.
<i>Bad Data</i> =	Queued member.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	PCBLA number.
<i>Dump 1</i> =	PCBLA tuple.

Table 34 Error Code: RMMOD

Error Code:	RMMOD
Error Description:	The rIPCBLA tuple EIS switching module number doesn't match the global processor ID (Gl_pcrd) value.
Possible Error Effect:	The switching module is configured incorrectly along with containing the incorrect data in the SM.
Corrective Action Taken:	Update rIPCBLA tuple attribute eis_sm with the Gl_pcrd.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute eis_sm.
<i>Good Data</i> =	Gl_pcrd.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump</i> =	rIPCBLA tuple.

Table 35 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The B-channel and D-channel, used by this process, are not in the same DSL.
Possible Error Effect:	The message and data for call processing will be erroneously transmitted.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.

<i>Bad Data</i> =	Logical port (D-channel).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 36 Error Code: SIMLNK

Error Code:	SIMLNK
Error Description:	The rIPCBLA tuple attribute sm_rel_Ink doesn't match the static rIDLDSL tuple attribute sm_rel_Ink.
Possible Error Effect:	Listing service calls will be mishandled, resulting in calls not receiving automatic treatment.
Corrective Action Taken:	Update rIPCBLA tuple attribute sm_rel_Ink with static rIDLDSL tuple attribute sm_rel_Ink.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA tuple.
<i>Bad Data</i> =	rIPCBLA tuple attribute sm_rel_Ink.
<i>Good Data</i> =	rIDLDSL tuple attribute sm_rel_Ink.
<i>Logical Index</i> =	rIPCBLA tuple logical key (procnum).
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	rIDLDSL tuple.

Table 37 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value for its use in an rIPCBLA.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the rIORTLA changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Contents of state field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 38 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The state field indicated that the rIPCBLA is not linked to a rIORTLA or rICPLR, but the linkage is not null.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the process changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 39 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	The state field indicated that the rIPCBLA is linked to a rIORTLA, but there is no linkage.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the portla changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA tuple.

Table 40 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	The state and hh_inprog fields indicated that the rIPCBLA is linked to a rPORTLA or rCPLR, but there is no linkage.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the process changes its associations.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump</i> =	rIPCBLA.

Table 41 Error Code: STUCK

Error Code:	STUCK
Error Description:	An rIPCBLA has been in a transient customer state for at least five minutes and has not changed state, or the customer state is stable but the pcbla audit flag was left in an invalid state.
Possible Error Effect:	The port involved, is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Contents of custat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	Process control block.

Table 42 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	An rIPCBLA, which is being used for modem pooling, has been in a transient customer state for at least five minutes and has not changed state, or the customer state is stable but the pcbla audit flag was left in an invalid state.
Possible Error Effect:	The port involved is out-of-service and at least one process is not available for use.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPCBLA.
<i>Bad Data</i> =	Contents of custat.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rIPCBLA number.
<i>Dump 1</i> =	rIPCBLA tuple.
<i>Dump 2</i> =	Process control block.

Table 43 Error Code: TAG

Error Code:	TAG
Error Description:	This error code is used for the two B-channel transfer notification to controller (TBCT NTC) feature auditing call tags in the AU bitmap AUtouch_bits[] against call tags in the call processing bitmap Fltag_map[]. This error indicates a call tag in the call processing bitmap is not set but should be.

Possible Error Effect:	If the call tag is not marked as busy in the call processing bit map, then call processing would not provide notification of transferred call clearing for that transfer since its call tag is marked as idle. Also, call processing may attempt to assign the call tag to another transfer resulting in a duplicate call tag and loss of integrity.
Corrective Action Taken:	AU will recover the call tag by setting it to busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of rIPCBLA.calltag.
<i>Bad Data =</i>	The value returned from the FITA macro.
<i>Good Data =</i>	The value of a busy tag (ISBUSY_TAG = 1).
<i>Logical Index =</i>	rIPCBLA logical key (procnum).
<i>Dump =</i>	rIPCBLA tuple.

Table 44 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The rIPCBLA tuple represents a monitor bridge terminal process associated with the ACDBRCS software package. However, the audit could not obtain a pointer to the corresponding rITEDB tuple representing the position to be monitored.
Possible Error Effect:	Call processing will not work correctly for this port described by this rITEDB tuple.
Corrective Action Taken:	The process represented by this rIPCBLA tuple will be purged. The related TEDB audit is scheduled to clean up problems in the RLTEDB relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	The value of the tedb field obtained from the bad rITEDB pointer.
<i>Good Data =</i>	The value of the tedb field calculated by the RTLP2VPORT macro.
<i>Logical Index =</i>	rIPCBLA number.
<i>Dump =</i>	rIPCBLA tuple.

Table 45 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUold_time, used for transient timing may have occurred.
Possible Error Effect:	The audit's transient timing has been interrupted. Lost resources may take longer to recover because transient timing must be restarted.
Corrective Action Taken:	AUold_time will be updated with the current OSDS ITIME (OSnow) and processing will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the transient timing global variable (AUold_time).
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Content of the transient timing variable (AUold_time).
<i>Dump 2 =</i>	Content of the current OSDS ITIME (OSnow).

Table 46 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The state field indicates that B-channel has been moved to another process, but the linkage fields are not NULL. The PCBLA tuple should not link to PORTLA and CCBCOM at this moment.
Possible Error Effect:	This B-channel cannot be used for other calls.
Corrective Action Taken:	The corresponding process is purged from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPCBLA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	riPCBLA number.
<i>Dump</i> =	riPCBLA tuple.

PCCB

Software Release: 5E10 and later

1. PERIPHERAL CHANNEL CONTROL BLOCK (PCCB) AUDIT

Peripheral channel control blocks represents a channel in the second message handler (MH) of the digital networking unit-synchronous optical network (SONET) (DNU-S). The DNU-S is a new peripheral for the 5ESS[®] switch that provides a cost of goods (COGS) reduction and synchronous optical network (SONET) capabilities. The second message handler provides all of the MH resident operational and maintenance software for establishing and maintaining communication paths using the peripheral control and timing (PCT) link between the SMP of SM-2000 and the DNU-S. Link access protocol for D-channel (LAPD) will be used to communicate to the DNU-S using the PCT links over three fixed timeslots as control to each common control (CC). An MH channel consists of these three time slots and a one-to-one mapping exist between an MH channel and a DNU-S CC side.

The PCCB audit runs in segmented mode and is part of the second MH base level loop. This base level loop provides for loading and unloading FIFOs used for communicating to DNU-S. It also provides time for routine work such as mate (CC) update, hashsum checks for active and mate MHs and for audits.

2. DATA STRUCTURES

The PCCB audit checks the peripheral channel control block array (MTpccb[]) made up of MTPCCB data structure elements. There is one MTPCCB data structure associated with the each channel in the second MH allowing for a maximum of 32 channels on a second MH. The MTPCCB data consists of:

- *teib[]* array for LLCB linkages
- *link_stat[]* array for logical link status
- *tidx* field for timer index
- *ttype* field for timer type
- *chan_act* for channel status

3. AUDIT CHECKS

The audit checks the MTpccb[] array and prints audit errors if needed as follows:

BADIDLE	= Make null and range checks for ttype and tidx fields.
BDPID	= Null checks for pid field and other related checks.
BDLINK	= Check the lnk_busy flag.
NOMATCH	= Check whether channel should be turned on or off
NONNULL	= Check that all the fields in the current PCCB are null
OUTOFRANGE	= Check ranges for tei and sapi2llcb fields.
STATUS	= Range check the status field.
TEIBAD	= Check linkages to Logical Link Control Blocks.
TIMERLNK	= Check linkages to timers

ZAP = Check for system interrupt handling.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The <i>ttype</i> and <i>tidx</i> fields failed the null and validity checks. When <i>ttype</i> field is null, the <i>tidx</i> field is not null and vice versa. Also, either <i>ttype</i> or <i>tidx</i> field has invalid values. Any of these conditions can trigger this error.
Possible Error Effect:	Incorrect operations of the channel, possible message failure.
Corrective Action Taken:	Null out the <i>ttype</i> , <i>tidx</i> and <i>pid</i> values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>tidx</i> field of the PCCB.
<i>Bad Data</i> =	The <i>ttype</i> field of the PCCB.
<i>Good Data</i> =	MTT_NULL, null value of <i>ttype</i> .
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit calculated the link busy flag value, checked it against the same value in the field <i>lnk_busy</i> found in the link status array entry, <i>link_stat[]</i> , in the PCCB and failed.
Possible Error Effect:	One of the links between the CC and the second MH may be down resulting in failure of messages between the DNU-S and SM.
Corrective Action Taken:	Correct the <i>lnk_busy</i> field value by the value calculated by audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the entry in the <i>link_stat[]</i> array in the PCCB.
<i>Bad Data</i> =	The <i>lnk_busy</i> field in the <i>link_stat[]</i> array.
<i>Good Data</i> =	The correct value of the <i>lnk_busy</i> field calculated by audit.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 3 Error Code: BDPID

Error Code:	BDPID
Error Description:	The <i>ttype</i> or <i>tidx</i> field failed either null or validity checks when there is a non-null processor id, <i>pid</i> , value for the channel.
Possible Error Effect:	Possible message failure between SM and DNU-S.
Corrective Action Taken:	The <i>pid</i> field is assigned the null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>tidx</i> field.
<i>Bad Data</i> =	The <i>ttype</i> field.
<i>Good Data</i> =	The process number, <i>pid.procno</i> .
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 4 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The audit searched the current channel being audited in the write protected MTtsb[] array that contains MH channel to timeslot mapping. If the channel

	number is not found in this array, then the channel should be turned off. When audit checked this result against the <i>chan_act</i> field, the check failed.
Possible Error Effect:	Possible message failure between DNU-S and SM.
Corrective Action Taken:	Correct the <i>chan_act</i> field value by the value calculated by audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The correct value of the <i>chan_act</i> field calculated by audit.
<i>Bad Data</i> =	The <i>chan_act</i> field.
<i>Good Data</i> =	The index in the MTtsb[] array where audit found the channel or the maximum number of the array index when audit did not find the channel.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 5 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	One of the fields of the PCCB under audit failed the null value check when either this was the 0th PCCB entry or audit determined during the <i>chan_act</i> check that this channel should be turned off.
Possible Error Effect:	Possible message failure between SM and DNU-S.
Corrective Action Taken:	Null the PCCB entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCCB.
<i>Bad Data</i> =	The local value calculated by audit that determines whether the channel should be turned on or off. This should be off.
<i>Good Data</i> =	The value of the <i>chan_act</i> field.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 6 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	Either the <i>tei</i> or the <i>sapi2llcb</i> field of the teib[] array entry failed the range check.
Possible Error Effect:	One of the links between the CC and the second MH may be down resulting in failure of messages between the DNU-S and SM.
Corrective Action Taken:	Correct the <i>tei</i> and <i>sapi2llcb</i> fields to null values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>sapi2llcb</i> field.
<i>Bad Data</i> =	The <i>tei</i> field.
<i>Good Data</i> =	The index of the entry in the teib[] array where this error was found.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 7 Error Code: STATUS

Error Code:	STATUS
Error Description:	The <i>status</i> field of the link_stat[] array entry failed the range check test.
Possible Error Effect:	One of the links between the CC and the second MH may be down resulting in failure of messages between the DNU-S and SM.
Corrective Action Taken:	The link_stat[] array entry where this error was found is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the entry in the link_stat[] array where the error was found.
<i>Bad Data</i> =	The <i>status</i> field.
<i>Good Data</i> =	The null value of the <i>status</i> field, MTDS_NULL.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 8 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	The audit failed linkage checks with the LLCB and PCCB. It accessed LLCB

	using the <i>sapi2llcb</i> field of PCCB and checked the <i>tei</i> and <i>teib_ix</i> fields against the same values in PCCB and failed the check.
Possible Error Effect:	One of the links between the CC and the second MH may be down resulting in failure of messages between the DNU-S and SM.
Corrective Action Taken:	Null out the current entry of the <i>teib[]</i> array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>teib_ix</i> field of LLCB linked to this PCCB.
<i>Bad Data</i> =	The <i>tei</i> value calculated by audit using <i>sapi2llcb</i> field and LLCB.
<i>Good Data</i> =	The index of the entry of the <i>teib[]</i> array that was in error.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 9 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The audit failed linkage checks with the timer element and PCCB. It accessed timer element using the <i>tidx</i> field from PCCB and checked the <i>sidx</i> and <i>stype</i> fields to reflect that it is indeed connected with this PCCB and failed the check.
Possible Error Effect:	Possible message failure between SM and DNU-S.
Corrective Action Taken:	Null out the <i>tidx</i> , <i>ttype</i> and <i>pid</i> fields.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>sidx</i> field of timer element.
<i>Bad Data</i> =	The <i>tidx</i> field.
<i>Good Data</i> =	The <i>stype</i> field of the timer element.
<i>Logical Index</i> =	The PCCB index.
<i>Dump</i> =	The PCCB being audited.

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audits will take this error as the result of a write protection violation, out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Audits may have caused the interrupt so action must be taken not to roll in the interrupts.
Corrective Action Taken:	If the PCCB index is in range, an interface will be called to reinitialize this channel.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	the value of SMmhrunning, telling which MH process is running.
<i>Bad Data</i> =	MTLLNULL, null value of logical link.
<i>Good Data</i> =	CM_SPYCHMAX, maximum number of channels.
<i>Logical Index</i> =	The PCCB index when the error occurred.
<i>Dump</i> =	None.

24. PCD - PE

PCD

Software Release: 5E10 and later

1. PERIPHERAL CONFIGURATION DATA (PCD) AUDIT

The peripheral configuration data audit (PCD) verifies the states of the hardware that is controlled by the administrative module (AM), as shown in Figure 1. This hardware includes the message switch (MSGS) complex, the network clock reference (NCREF) circuits, and the office network and timing complex common (ONTCCOM). The following units make up the message switch: the message switch control unit (MSCU), the pump peripheral controller (PPC), the foundation peripheral controller (FPC), the communications module processor (CMP), and module message processors (MMPs). The office network and timing complex common includes the following units: the network clock (NCLK), the message interface (MI), and the time-multiplexed switch (TMS).

The units of the message switch and the office network and timing complex (ONTC) make up the communication module (CM). The software that controls the communication module is part of the switch maintenance subsystem; it resides in the switch maintenance kernel process (SMKP). Another kernel process, the message switch kernel process (MSKP), is the communication link to the hardware. MSKP interacts with the hardware to detect changes in hardware states; it informs SMKP of the changes so that SMKP can order the appropriate actions.

The state data for each unit is kept in the peripheral configuration database in the administrative module. The PCD audit checks the consistency of this data and ensures that the states of the hardware agree with the states indicated by the controlling software. The audit, a part of the switch maintenance audits subgroup, runs in the SMKP environment in the segmented and directed modes.

2. DATA STRUCTURES

The static and dynamic relations and arrays checked in this audit make up the peripheral configuration database associated with existing units. The administrative module path data relation (RLPHDATA) contains the link numbers, status, and host switching module (HSM) number. Static entries and dynamic arrays exist for each unit except the MSCU, which has a dynamic entry. These arrays include the SMdmscu array for the message switch control unit, the SMdpc array for peripheral controllers, and the SMdcore array for the office network and timing complex. There are no audited linkages among these data structures.

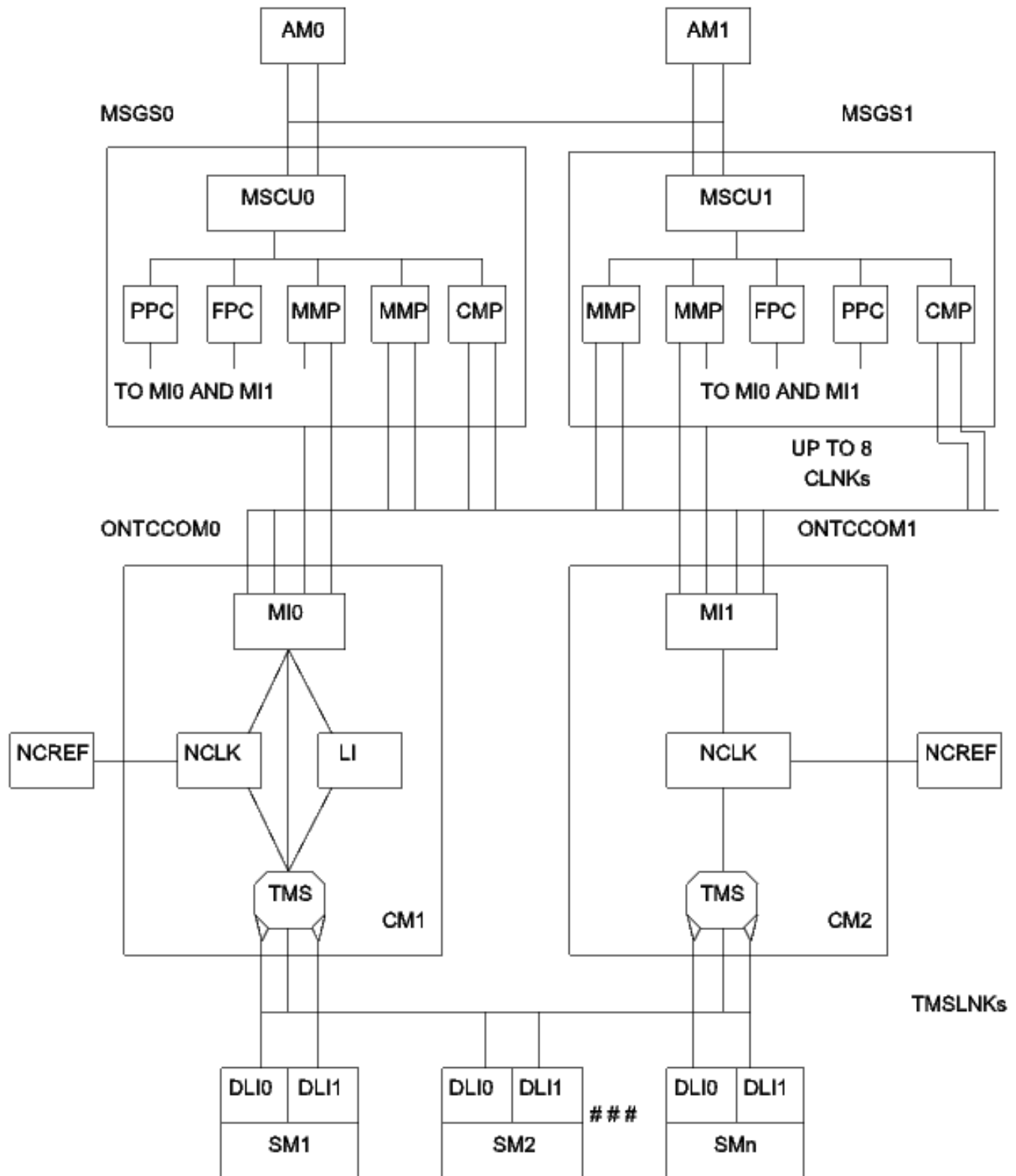


Figure 1 (PCD-1) PCD Hardware

3. AUDIT CHECKS

The audit makes similar checks on each of the units. These checks include a verification of the software state of the unit, a check to ensure that the mate unit is in an appropriate state, and whether deferred work is pending on out-of-service units. The variations in these checks result from the differences in the units, their functions, and the hardware architecture.

- (a) The audit first checks the MSCU. Along with the internal check the audit makes of the validity of the state of the MSCU, the audit checks whether the interrupt inhibit state of the MSCU is valid. Another check the audit performs ensures that both MSCUs are not out of service; both may be active, though. Finally, the audit determines whether additional work pending for an out-of-service MSCU is properly scheduled.
- (b) In the FPC, the audit checks whether the state of the FPC is legal. Since the FPC is a peripheral device under the MSCU, it must be out of service when the MSCU is out of service. FPCs run in the active/standby mode and now also in the out-of-service (OOS)/OOS mode for the duplex failure case; the audit ensures that the FPCs reflect valid combinations of this mode. The audit ensures that the interrupt inhibit state is set correctly (if the FPC is out of service, for example, the interrupt inhibit state must be INHIBIT), and performs configuration checks to ensure that the hardware configuration agrees with the state of the software. The audit also makes deferred work checks.
- (c) The checks the audit makes for the PPC are the same as those made for the foundation peripheral controller with one exception. Because the PPC is not a critical unit, both PPCs may be out of service at the same time.
- (d) The checks the audit makes for the MMPs are similar to the ones made for the FPC except that all the MMPs may be active at the same time. The audit does not perform a mate check for MMPs.
- (e) The checks the audit makes for the CMP are similar to the ones for the FPC. Some specific checks done for the CMP include a check of the INIT state of the CMP. At this time there should be a job on the switch maintenance analysis and recovery strategies (SMARS) queue.
- (f) The audit checks the office network and timing complex common. There is only one state for all the units within the ONTCCOM. The audit makes an internal check to ensure that this state is valid, and it checks to ensure that the mate's state is compatible. The ONTCCOM is not directly under the MSCU and may be in service when the MSCU is not. Legal states for the ONTCCOMs include active major and active minor, and active major and out-of-service. Invalid states occur when both ONTCCOMs are out of service or both are listed as active major. The other checks the audit makes on the ONTCCOM include a check of the interrupt inhibit state, a check of the hardware configuration, and a check for deferred work.
- (g) The network clock references the ONTCCOM through the NCLK. The audit verifies the states of the references, and ensures that the state of the references is compatible with that of the ONTCCOM. The audit checks the interrupt inhibit state of the references and the configuration of the hardware.
- (h) The audit verifies the states of the network clock oscillator (NCOSC), and ensures that its state is compatible with that of the ONTC.
- (i) The audit checks the mode of the network clock to ensure its validity. If the mode is normal, the clock is receiving timing from an external reference. In the holdover mode, and internal oscillator provides timing for the clock.
- (j) The audit checks the AM/CM re-synchronization flag SMrsmode. If the SMrsmode flag is set, a re-synchronization job should be pending on the SMARS job queue. If there is no job on the queue, the SMrsmode flag is written to normal.
- (k) The audit validates the pending state (DDPCMISOL) of the CM isolation data delivery (DD) status key DDSCMISOL. If the CM isolation DD request key DDRCMISOL is not normal, then a CM isolation request should be pending on the SMARS job queue. If the CM isolation DD request key DDRCMISOL is normal, then the AM/CM re-synchronization flag SMrsmode should be set. If either case is not true, then the

DDSCMISOL pending state is invalid and written to normal.

4. ERROR HANDLING

The type of error determines the actions the audit takes. Several errors, including the detection of a bad software state, the detection of an invalid mate state, and the check for deferred work showing that no work is pending when it should be, trigger a conditional restoration or switch of the hardware unit. A conditional restoration is possible only in the duplex mode; it removes the unit from service, diagnoses it, and restores it to service if the diagnostics pass.

Other errors are corrected by making the states agree. For example, if the interrupt inhibit state does not agree with the hardware, the audit sends a message to change the hardware state so it agrees with the state in the interrupt inhibit table.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CMP_CONFIG

Error Code:	CMP_CONFIG
Error Description:	The communications module processor (CMP) DDPCONFIGST key does not agree with the states of the CMP. Specifically, none of the following situations is true: - The CMP base state is SM5ACT while the mate is not SM5STBY and the DD key is not SMSWNOBLK. - The CMP base state is SM5STBY while the qual2 field is not SMQ2UPD and the DD key is either SMSWUPBLK or SMSWSWBLK.
Possible Error Effect:	Whoever accesses DDPCONFIGST will receive the wrong status.
Corrective Action Taken:	The DDPCONFIGST key is cleared until it is recovered by SM code.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, ffffff is printed if there is only one unit of this type on the side. For network clock references valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 2 Error Code: CMP_DMAM

Error Code:	CMP_DMAM
Error Description:	The CMP is in an out-of-service state where there should be some deferred work performed on it but, deferred maintenance administrator and monitor (DMAM) does not have an entry in its queue for this unit.

Possible Error Effect:	CMP is out-of-service with no automatic action pending to restore it to service.
Corrective Action Taken:	Audit requests a conditional restore of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, ffffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 3 Error Code: CMP_FE

Error Code:	CMP_FE
Error Description:	The CMP is OOS family of equipment (FE) when the parent MSCU is active.
Possible Error Effect:	A CMP is OOS when it should not be.
Corrective Action Taken:	The software state of the CMP is written to OOS AUTO RMV, and a conditional restore to active is requested on the CMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, ffffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 4 Error Code: CMP_IIA

Error Code:	CMP_IIA
Error Description:	The CMP software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, ffffff is printed if there is only one unit of this type on the side. For network clock references, valid values are:

	1 = The primary reference.
	2 = The secondary reference.
	3 = The cross couple.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 5 Error Code: CMP_INIT

Error Code:	CMP_INIT
Error Description:	The CMP is in a out-of-service Maintenance in progress state and there is no entry for it on the SMARS queue.
Possible Error Effect:	The CMP is out-of-service and there is no maintenance action pending on it.
Corrective Action Taken:	The CMP state is written to OOS AUTO RMV, and a conditional restore is requested on the CMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, fffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 6 Error Code: CMP_LLM0

Error Code:	CMP_LLM0
Error Description:	The logical link map entry for the CMP does not agree with the state of the FPC.
Possible Error Effect:	Call processing connect orders are not going to the active CMP. New intermodule calls are not being connected.
Corrective Action Taken:	The audit updates the CMP logical link map entry to agree with the active CMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, fffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 7 Error Code: CMP_PCD

Error Code:	CMP_PCD
Error Description:	The CMP has been found to be in an invalid state. The number of valid (and invalid) combinations of equipage states, base states, qual1 and qual2 values are too numerous to list here.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, ffffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 8 Error Code: CMP_SSWSTAT

Error Code:	CMP_SSWSTAT
Error Description:	The CMP DDPSSWSTAT key does not agree with the states of the unit. Specifically, none of the following situations are true: - The CMP base state is SM5ACT while the mate is not SM5STBY and the DD key is not SMSWNOSW. - The CMP base state is SM5STBY while the qual2 field is not SMQ2UPD and the DD key is either SMSWINGP or SMSWRCBLK.
Possible Error Effect:	The soft switch status for the CMP will be incorrect.
Corrective Action Taken:	The DDPSSWSTAT key is set to NOSSW.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the unit's number is printed; otherwise, ffffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 9 Error Code: LNK_PCD

Error Code:	LNK_PCD
Error Description:	A TMS NCT link has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Audit's copy of the data read.

Table 10 Error Code: MMP_CONFIG_SIDE1

Error Code:	MMP_CONFIG_SIDE1
Error Description:	The MMP is not receiving its timing from the active major ONTC side.
Possible Error Effect:	AM to SM communication links on this MMP could be taking errors.
Corrective Action Taken:	The audit has the MMP receive its timing from the ONTC on the system side zero, since this is the active major side.
Manual Action Required:	If the audit rolls in this error on the same MMP, then diagnose the MMP to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 11 Error Code: MMP_CONFIG_SIDE

Error Code:	MMP_CONFIG_SIDE
Error Description:	The MMP is not receiving its timing from the active major ONTC side.
Possible Error Effect:	AM to SM communication links on this MMP could be taking errors.
Corrective Action Taken:	The audit has the MMP receive its timing from the ONTC on system side one since this is the active major side.
Manual Action Required:	If the audit rolls in this error on the same MMP, then diagnose the MMP to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 12 Error Code: MMP_DMAM

Error Code:	MMP_DMAM
Error Description:	The MMP is in a out-of-service state where there should be some deferred work being done on it, but DMAM does not have an entry in its queue for this unit.
Possible Error Effect:	MMP is out-of-service with no automatic action pending to bring it back to

	service.
Corrective Action Taken:	Audit requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 13 Error Code: MMP_FE

Error Code:	MMP_FE
Error Description:	The MMP is OOS FE when the parent MSCU is active.
Possible Error Effect:	An MMP is OOS when it should not be.
Corrective Action Taken:	The software state of the MMP is written to OOS AUTO RMV and a conditional restore to active is requested on the MMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 14 Error Code: MMP_HDW_INH

Error Code:	MMP_HDW_INH
Error Description:	The MMP software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited.
Corrective Action Taken:	The audit that requests the hardware interrupt inhibit status be set to what the software interrupt inhibit status is. If both interrupts statuses are inhibit man, then the audit changes them to inhibit auto. The audit also asks for a conditional restore of the MMP.
Manual Action Required:	If the audit rolls in this error on the same MMP, then diagnose the MMP to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 15 Error Code: MMP_IIA

Error Code:	MMP_IIA
Error Description:	The MMP software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in the SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 16 Error Code: MMP_MIP

Error Code:	MMP_MIP
Error Description:	The MMP is in a OOS Maintenance in progress state and there is no entry for it on the SMARS queue.
Possible Error Effect:	The MMP is out-of-service and there is no maintenance action pending on it.

Corrective Action Taken:	The MMP state is written to OOS AUTO RMV and a conditional restore is requested on the MMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 17 Error Code: MMP_MSG_GAP

Error Code:	MMP_MSG_GAP
Error Description:	The message gapping time in the SDLC (protocol device) of MMP is not consistent with the gapping time specified in the RLSSDLC and RLMODATT relation. Each MMP can have up to 8 SLDC which connects to 8 SMs. For each SM, in order to determine the message gapping time from static ODD, read RLSSDLC to determine which SM is on this MMP. If the equipment status field (rISDLC.eqstat) is marked with either SMSGRO (operational growth state) or SMOPER (special growth state), then read RLMODATT for this SM to determine the message gapping value. If this SM has either of the following then it should have a gapping value of 5 milliseconds: (1) SMP20 hardware (rIMODATT.unitttype = SMSMP20); or, (2) TN1617 hardware (rIMODATT.mctsi0_cli & rIMODATT.mctsi1_cli & DGVS DLCINT = 1). Otherwise, this SM should have the gapping value of 0 milliseconds.
Possible Error Effect:	The message sent by this MMP will not be synchronous with the receiving SM.
Corrective Action Taken:	A request for a conditional restore will be sent for the MMP unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on this side, for example, the MMPs and TMS NCT links, then the unit number is printed. Otherwise, "ffffff" is printed.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 18 Error Code: MMP_PCD

Error Code:	MMP_PCD
Error Description:	The MMP has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 19 Error Code: MMP_TEMP

Error Code:	MMP_TEMP
Error Description:	The MMP is OOS AUTO TEMP and neither ONTC is in a diagnostic state.
Possible Error Effect:	An MMP is out-of-service when it should not be.

Corrective Action Taken:	The software state of the MMP is written to OOS AUTO RMV and a conditional restore to active is requested on the MMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The MMP number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 20 Error Code: MODATT

Error Code:	MODATT
Error Description:	Failure occurred when trying to read the rIMODATT tuple for the specified SM to determine what message gap value it should have.
Possible Error Effect:	A SM may be having communication problems.
Corrective Action Taken:	None.
Manual Action Required:	Missing rIMODATT tuple should be inserted into database. Using ODBE insert the missing tuple using the module field of the rIMODATT tuple dumped in the <i>Dump 2</i> of this error report as the key.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The error code received while attempting to access the database. Return code from DBfrdtup.
<i>Good Data =</i>	Communication link 1.
<i>Logical Index =</i>	Communication link 0.
<i>Dump 1 =</i>	The equipment ID for the unit.
<i>Dump 2 =</i>	rIMODATT tuple.
<i>Dump 3 =</i>	The state of the unit being checked (AUBIG ST structure).

Table 21 Error Code: MSCU_DMAM

Error Code:	MSCU_DMAM
Error Description:	The MSCU is in a OOS state where there should be some deferred work being done on it, but DMAM does not have an entry in its queue for this unit.
Possible Error Effect:	MSCU is out-of-service with no automatic action pending to restore it to service.
Corrective Action Taken:	Audit requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 22 Error Code: MSCU_FRCD

Error Code:	MSCU_FRCD
Error Description:	The MSCU is in a forced state, and the corresponding data delivery bits are not set.
Possible Error Effect:	MSCU is forced when it should not be.
Corrective Action Taken:	The audit removes the MSCU force state and requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 23 Error Code: MSCU_FRCD_DD

Error Code:	MSCU_FRCD_DD
Error Description:	The MSCU is not in a forced state and the corresponding data delivery bits are

	set.
Possible Error Effect:	MSCU data delivery force bits are set and they should not be.
Corrective Action Taken:	The audit clears the MSCU data delivery force bits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 24 Error Code: MSCU_HDW_INH

Error Code:	MSCU_HDW_INH
Error Description:	The MSCU software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed or allowed, when the software says they are inhibited.
Corrective Action Taken:	The audit requests that the hardware interrupt inhibit status to be set to what the software interrupt inhibit status is. If both interrupts statuses are inhibit man, then the audit changes them to inhibit auto. The audit also asks for a conditional restoration of the MSCU.
Manual Action Required:	If the audit rolls in this error on the same MSCU, then diagnose the MSCU side to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 25 Error Code: MSCU_IIA

Error Code:	MSCU_IIA
Error Description:	The MSCU software interrupt inhibit state is invalid.
Possible Error Effect:	None.
Corrective Action Taken:	The interrupt inhibit state is written to a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 26 Error Code: MSCU_PCD

Error Code:	MSCU_PCD
Error Description:	The MSCU has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 27 Error Code: MSCU_TEMP

Error Code:	MSCU_TEMP
Error Description:	The MSCU was in a TEMP state (helper unit for diagnostics) and an FPC, PPC or MMP was not in a diagnostic or helper state.

Possible Error Effect:	The MSCU is simplex.
Corrective Action Taken:	The software state of the MSCU is written to OOS AUTO RMV or (UNAV FRCD RMV if the MSCU is forced), and a condition restore is requested if the MSCU is not forced.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 28 Error Code: NCLK2_OSC_CONFIG

Error Code:	NCLK2_OSC_CONFIG
Error Description:	The network clock 2 oscillator hardware state does not agree with the network clock 2 oscillator software state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	Audit requests a conditional restore or remove of the oscillator so that the hardware is updated to agree with the software.
Manual Action Required:	Try to remove and restore the oscillator. If the audit continues rolling in this error, diagnose the ONTC side which the network clock 2 oscillator is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 29 Error Code: NCLK2_OSC_PCD

Error Code:	NCLK2_OSC_PCD
Error Description:	A network clock 2 oscillator has been found to be in an invalid state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	Try to remove and restore the oscillator. If the audit rolls in this error, diagnose the ONTC side which the network clock 2 oscillator is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 30 Error Code: NCLK2_OSCXC_CONFIG

Error Code:	NCLK2_OSCXC_CONFIG
Error Description:	The network clock 2 oscillator cross couple hardware state does not agree with the network clock 2 oscillator cross couple software state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	Audit requests a restore or remove of the oscillator cross couple so that the hardware is updated to agree with the software.
Manual Action Required:	Try to remove and restore the oscillator. If the audit continues rolling in this error, diagnose the ONTC side which the network clock 2 oscillator cross couple is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 31 Error Code: NCLK2_OSCXC_PCD

Error Code:	NCLK2_OSCXC_PCD
Error Description:	A network clock 2 oscillator cross couple has been found to be in an invalid state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	Try a remove and restore of the oscillator. If the audit rolls in this error, diagnose the ONTC side which the network clock 2 oscillator cross couple is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 32 Error Code: NC_MODE_CONFIG

Error Code:	NC_MODE_CONFIG
Error Description:	The hardware state of the network clock mode does not agree with the software state of the network clock mode.
Possible Error Effect:	System could be having timing problems.
Corrective Action Taken:	The audit requests the hardware state to be set to the software state.
Manual Action Required:	If the audit rolls in this error, diagnose the ONTC side which this network clock is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 33 Error Code: NC_MODE_PCD

Error Code:	NC_MODE_PCD
Error Description:	The network clock mode has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 34 Error Code: NC_REF_CONFIG

Error Code:	NC_REF_CONFIG
Error Description:	The network clock 2 reference hardware state does not agree with the network clock 2 reference software state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	Audit requests a restore or remove of the Reference so that the hardware is updated to agree with the software. Also, a phase lock of the network clock

	hardware is requested.
Manual Action Required:	If the audit rolls in this error, diagnose the ONTC side which the network clock 2 reference is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The reference number is printed.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 35 Error Code: NC_REF_FE

Error Code:	NC_REF_FE
Error Description:	A network clock 2 reference has been found to be in an invalid state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	Try a remove and restore of the reference. If the audit rolls in this error, diagnose the ONTC side which the network clock reference is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 36 Error Code: NC_REF_HDW_INH

Error Code:	NC_REF_HDW_INH
Error Description:	The network clock 2 reference software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited. Errors could be masked or allowed to report errors when unexpected.
Corrective Action Taken:	The audit requests that the hardware interrupt inhibit status to be set to what the software interrupt inhibit status is. If both interrupt statuses are inhibit man, the audit changes them to inhibit auto.
Manual Action Required:	If the audit rolls in this error, on the same network clock 2 reference, then diagnose the ONTC side which this reference is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The NCLK2 reference number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 37 Error Code: NC_REF_IIA

Error Code:	NC_REF_IIA
Error Description:	The network clock 2 reference software interrupt inhibit state is invalid.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited. Errors could be masked or allowed to report errors when unexpected.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The NCLK2 reference number.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).

<i>Dump</i> =	Audit's copy of the data read.

Table 38 Error Code: NC_REF_PCD

Error Code:	NC_REF_PCD
Error Description:	A network clock 2 reference has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The NCLK2 reference number.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 39 Error Code: ONTC_CONFIG_LI

Error Code:	ONTC_CONFIG_LI
Error Description:	The hardware state of the link interface (LI) does not agree with the software state of the ONTC.
Possible Error Effect:	ONTC fault recovery may be taking improper actions.
Corrective Action Taken:	The ONTC is duplex; the error is reported to fault recovery, and a conditional restore may be requested of the ONTC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 40 Error Code: ONTC_CONFIG_NCLK

Error Code:	ONTC_CONFIG_NCLK
Error Description:	The hardware state of the network clock and the software state of the ONTC do not agree.
Possible Error Effect:	Units could be getting bad timing.
Corrective Action Taken:	The ONTC is duplex; the error is reported to fault recovery, and a conditional restore may be requested of the ONTC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 41 Error Code: ONTC_CONFIG_TMS

Error Code:	ONTC_CONFIG_TMS
Error Description:	The hardware state of the TMS does not agree with the software state of the ONTC.
Possible Error Effect:	ONTC fault recovery may be taking improper actions.
Corrective Action Taken:	The error is report to fault recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 42 Error Code: ONTC_DF_MRK

Error Code:	ONTC_DF_MRK
Error Description:	The ONTC fault recovery counters are out of range.
Possible Error Effect:	ONTC fault recovery may be taking improper actions.
Corrective Action Taken:	The audit clears the counters.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 43 Error Code: ONTC_DMAM

Error Code:	ONTC_DMAM
Error Description:	The ONTC is in a out-of-service state where there should be some deferred work being done on it, but the DMAM does not have an entry in its queue for this unit.
Possible Error Effect:	ONTC is out-of-service with no automatic action pending to bring it back in service.
Corrective Action Taken:	Audit requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 44 Error Code: ONTC_FRCD

Error Code:	ONTC_FRCD
Error Description:	The ONTC is in a forced state and the corresponding data delivery bits are not set.
Possible Error Effect:	ONTC is forced when it should not be.
Corrective Action Taken:	The audit removes the ONTC force state and requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 45 Error Code: ONTC_FRCD_DD

Error Code:	ONTC_FRCD_DD
Error Description:	The ONTC is not in a forced state, and the corresponding data delivery bits are set.
Possible Error Effect:	ONTC data delivery force bits are set and they should not be.
Corrective Action Taken:	The audit clears the ONTC data delivery force bits.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 46 Error Code: ONTC_HDW_INH

Error Code:	ONTC_HDW_INH
Error Description:	The ONTC software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed

	or allowed, when the software says they are inhibited.
Corrective Action Taken:	The audit requests the hardware interrupt inhibit status to be set to what the software interrupt inhibit status is.
Manual Action Required:	If the audit rolls in this error, then diagnose the active FPC to see if there is a hardware problem. This is because the interrupts we are checking are located in the active FPC.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 47 Error Code: ONTC_IIA

Error Code:	ONTC_IIA
Error Description:	The ONTC software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in the SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 48 Error Code: ONTC_MIP

Error Code:	ONTC_MIP
Error Description:	The ONTC is in a Maintenance in progress (MIP) state, and the SMARS queue does not have an entry for the ONTC.
Possible Error Effect:	ONTC fault recovery may be taking improper actions.
Corrective Action Taken:	The audit reports the error to fault recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 49 Error Code: ONTC_MIP2

Error Code:	ONTC_MIP2
Error Description:	The ONTC is in a MIP state, and SMARS queue does not have an entry for the ONTC.
Possible Error Effect:	ONTC fault recovery may be taking improper actions.
Corrective Action Taken:	The audit reports the error to fault recovery.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 50 Error Code: ONTC_PCD

Error Code:	ONTC_PCD
Error Description:	The ONTC has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 51 Error Code: ONTC_SIMPLEX_LI

Error Code:	ONTC_SIMPLEX_LI
Error Description:	The hardware state of the LI does not agree with the software state of the ONTC.
Possible Error Effect:	ONTC fault recovery may be taking improper actions.
Corrective Action Taken:	The audit reports the error to fault recovery.
Manual Action Required:	ONTC is simplex; the out of service ONTC must be repaired and restored. The active side should be checked for LI errors.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 52 Error Code: ONTC_SIMPLEX_NCLK

Error Code:	ONTC_SIMPLEX_NCLK
Error Description:	The hardware state of the network clock, and the software state of the ONTC do not agree; the ONTC is simplex.
Possible Error Effect:	Units could be getting bad timing.
Corrective Action Taken:	The audit reports the error to ONTC fault recovery.
Manual Action Required:	The ONTC is simplex; the out of service ONTC must be repaired and restored. The active side should be checked for network clock errors.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 53 Error Code: ONTC_X2WCON

Error Code:	ONTC_X2WCON
Error Description:	The state of the ONTC and the call processing two way connect interface not agree.
Possible Error Effect:	New call processing connect orders are not sent to an active ONTC or are sent to an out-of-service ONTC.
Corrective Action Taken:	The call processing interface is updated to agree with the state of the ONTC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 54 Error Code: PCD

Error Code:	PCD
Error Description:	The audit has found an error which it cannot correct; this may be due to a bad ODD, bad static data, or the audit recovery table needs to be changed.
Possible Error Effect:	The audit will not recover this error.
Corrective Action Taken:	The audit will not recover this error.
Manual Action Required:	Check if the ODD is correct for the PCD. The relations this includes are:

	RLSMSCU = For the MSCU.
	RLSPC = For the MMPs, FPCs and PPCs.
	RLSCORE = For the ONTC.
	RLSNCREf = For the network clock references.
	RLSAUTISS = For AUTISS.
	RLSLNKTMS = For the TMSLNKs.
	RLMODATT = For the SMs.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	If there is more than one unit of this type on the side, for example the MMPs, then the units number is printed, or else fffff is printed there is only one unit of this type on the side.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 55 Error Code: PCD_SCORE

Error Code:	PCD_SCORE
Error Description:	The audit has found a problem with the RLSCORE relation in the ODD.
Possible Error Effect:	An ONTC may be out of service.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the RLSCORE relation for bad data.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	If there is more than one unit of this type on the side (for example, the MMPs and TMS NCT links), then the unit's number is printed; otherwise, fffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 56 Error Code: PCD_SLNKTMS

Error Code:	PCD_SLNKTMS
Error Description:	The audit has found a problem with the RLSLNKTMS relation in the ODD.
Possible Error Effect:	A TMS and NCT link may be out of service.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the RLSLNKTMS relation entry.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	If there is more than one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else fffff is printed if there is only one unit of this type on the side. For network clock references, valid values are:

	1 = The primary reference.
	2 = The secondary reference.
	3 = The cross couple.
<i>Logical Index</i> =	Side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Local buffer used in DBfrdtup() call.
<i>Dump 3</i> =	Audit's copy of the data read.

Table 57 Error Code: PCD_SNCREF

Error Code:	PCD_SNCREF
Error Description:	The audit has found a problem with the RLSNCREf relation in the ODD.
Possible Error Effect:	A network clock references may be out of service.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the RLSNCREf relation for bad data.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	If there is more then one unit of this type on the side, for example the MMPs then the units number is printed, or else fffff is printed if there is only one unit of this type on the side.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 58 Error Code: PCD_SPC

Error Code:	PCD_SPC
Error Description:	The audit has found a problem with the RLSPC relation in the ODD.
Possible Error Effect:	A MMP, FPC, or PPC may be out of service.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the RLSPC relation for bad data.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else fffff is printed if there is only one unit of this type on the side. For network clock references, valid values are: 1 = The primary reference. 2 = The secondary reference. 3 = The cross couple.
<i>Logical Index</i> =	Side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 59 Error Code: PPC_CONFIG_SIDE0

Error Code:	PPC_CONFIG_SIDE0
Error Description:	The PPC is not receiving its timing from the active major ONTC side.
Possible Error Effect:	Fast pump of SMs could be failing.
Corrective Action Taken:	The audit enables the PPC to receive its timing from the ONTC on system side zero, since this is the active major side.
Manual Action Required:	If the audit rolls in this error on the same PPC, then diagnose the PPC to see if there is a hardware problem.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 60 Error Code: PPC_CONFIG_SIDE1

Error Code:	PPC_CONFIG_SIDE1
Error Description:	The PPC is not receiving its timing from the active major ONTC side.
Possible Error Effect:	Fast pump of SMs could be failing.
Corrective Action Taken:	The audit enables the PPC to receive its timing from the ONTC on system side one, since this is the active major side.
Manual Action Required:	If the audit rolls in this error on the same PPC, then diagnose the PPC to see if there is a hardware problem.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 61 Error Code: PPC_DMAM

Error Code:	PPC_DMAM
Error Description:	The PPC is in a out-of-service state where there should be some deferred work being done on it, but DMAM does not have an entry in its queue for this unit.
Possible Error Effect:	PPC is out-of-service with no automatic action pending to bring it back to service.
Corrective Action Taken:	Audit requests a conditional restore on the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 62 Error Code: PPC_FE

Error Code:	PPC_FE
Error Description:	The PPC was in an OOS FE state when the parent SCMU is active.
Possible Error Effect:	The PPC is simplex.
Corrective Action Taken:	The software state of the PPC is written to OOS AUTO RMV, and a condition restore is requested on the PPC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 63 Error Code: PPC_HDW_INH

Error Code:	PPC_HDW_INH
Error Description:	The PPC software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited.
Corrective Action Taken:	The audit requests that the hardware interrupt inhibit status be set to what the software interrupt inhibit status is. If both the interrupts were inhibit man, the audit changes them to inhibit auto. The audit also asks for a conditional restore of the PPC.
Manual Action Required:	If the unit rolls in this error on the same PPC, then diagnose the PPC to see if there is a hardware problem.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 64 Error Code: PPC_IIA

Error Code:	PPC_IIA
Error Description:	The PPC software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in the SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 65 Error Code: PPC_PCD

Error Code:	PPC_PCD
Error Description:	The PPC has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 66 Error Code: PPC_TEMP

Error Code:	PPC_TEMP
Error Description:	The PPC was in a TEMP state (helper unit for ONTC diagnostics) and a ONTC was not in a diagnostic state.
Possible Error Effect:	The PPC is simplex.
Corrective Action Taken:	The software state of the PPC is written to OOS AUTO RMV, and a condition restore to STANDBY is requested on the PPC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 67 Error Code: RCMISOL

Error Code:	RCMISOL
Error Description:	The CM isolation DD status key DDSCMISOL is incorrectly set in the DDPCMISOL state. The CM isolation DD request key DDRCMISOL is not normal and entry into manual CM isolation should be in progress, however, there is no CM isolation request on the SMARS queue.
Possible Error Effect:	The DDSCMISOL DD key is stuck in what should be a transient pending state giving incorrect CM isolation status and stubbing off PCD auditing. Some communication module maintenance functionality may not be available.
Corrective Action Taken:	Write the DDSCMISOL key to DDNORMAL.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Value for DDRCMISOL.
<i>Bad Data</i> =	Value for DDSCMISOL.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Value for SMrsmode.
<i>Dump</i> =	NA.

Table 68 Error Code: RSMODE

Error Code:	RSMODE
Error Description:	The SMrsmode AM/CM re-synchronization flag is set, however, the DDSCMISOL key is normal and there is no entry on the SMARS queue indicating an AM/CM re-synchronization is in progress.
Possible Error Effect:	The SMrsmode flag is in an invalid state, falsely indicating a re-synchronization is in progress. Some communication module maintenance functionality may not be available.
Corrective Action Taken:	Write the SMrsmode flag to SMRSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value for DDRCMISOL.
<i>Bad Data</i> =	Value for DDSCMISOL.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Value for SMrsmode.
<i>Dump</i> =	NA.

Table 69 Error Code: SAUTISS

Error Code:	SAUTISS
Error Description:	The audit has found a problem with the RLSAUTISS relation in the ODD.
Possible Error Effect:	Automatic time slot switching (AUTISS) may not be in the right state.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the RLSAUTISS relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	The unit's identifier.
<i>Logical Index</i> =	The side the error was found on.
<i>Dump 1</i> =	Local buffer used in DBfrdtup() call.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 70 Error Code: SCMISOL

Error Code:	SCMISOL
Error Description:	The CM isolation DD status key DDSCMISOL is incorrectly set in the DDPCMISOL state. The CM isolation DD request key DDRCMISOL is normal and an AM/CM re-synchronization should be in progress, however, the SMrsmode re-synchronization flag is SMRSNULL.
Possible Error Effect:	The DDSCMISOL DD key is stuck in what should be a transient pending state giving incorrect CM isolation status and stubbing off PCD auditing. Some communication module maintenance functionality may not be available.
Corrective Action Taken:	Write the DDSCMISOL key to DDNORMAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value for DDRCMISOL.
<i>Bad Data</i> =	Value for DDSCMISOL.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Value for SMrsmode.
<i>Dump</i> =	NA.

Table 71 Error Code: SMSUCU

Error Code:	SMSUCU
Error Description:	The audit has found a problem with the RLSMSUCU relation in the ODD.
Possible Error Effect:	An MSCU may be out of service.
Corrective Action Taken:	None.

Manual Action Required:	If the error repeats, check the RLSMSCU relation for bad data.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DBfrdtup return code.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump 1 =</i>	Equipment identifier of the unit.
<i>Dump 2 =</i>	Local buffer used in DBfrdtup() call.
<i>Dump 3 =</i>	Audit's copy of the data read.

Table 72 Error Code: SSDLC

Error Code:	SSDLC
Error Description:	Failure occurred when trying to read the rSSDLC tuple to determine what SM are on the specified MMP during message gap value auditing.
Possible Error Effect:	This may cause asserts to be fired by application programs.
Corrective Action Taken:	None.
Manual Action Required:	Missing rMODATT tuple should be inserted into database.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The error code received while attempting to access the database.
<i>Good Data =</i>	Communication link 1.
<i>Logical Index =</i>	Communication link 0.
<i>Dump 1 =</i>	The equipment ID for the unit.
<i>Dump 2 =</i>	rMODATT tuple.
<i>Dump 3 =</i>	The state of the unit being checked (AUBIG_ST structure).

Table 73 Error Code: TIP_CONFIG

Error Code:	TIP_CONFIG
Error Description:	The primary network clock reference hardware state does not agree with the network clock reference software state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	Audit requests a restore or remove of the Reference so that the hardware is updated to agree with the software. Also a phase lock of the network clock mode is requested.
Manual Action Required:	If the audit rolls in this error, diagnose the ONTC side which the network clock references is on to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 74 Error Code: TIP_FE

Error Code:	TIP_FE
Error Description:	The primary network clock reference was in an out of service family state when the parent ONTC was in service.
Possible Error Effect:	The reference is out of service when it should not be.
Corrective Action Taken:	The reference is restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 75 Error Code: TIP_HDW_INH

Error Code:	TIP_HDW_INH
Error Description:	The primary network clock reference software and hardware interrupt inhibit states do not agree.

Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited.
Corrective Action Taken:	The audit requests the hardware interrupt inhibit status to be set to what the software interrupt inhibit status is. If both of the interrupts are inhibit man, the audit changes them to inhibit auto.
Manual Action Required:	If the audit rolls in this error on the same network clock Reference, then diagnose the ONTC side which this reference is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 76 Error Code: TIP_IIA

Error Code:	TIP_IIA
Error Description:	The primary network clock reference software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in the SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 77 Error Code: TIP_PCD

Error Code:	TIP_PCD
Error Description:	A primary network clock reference has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 78 Error Code: TIS_CONFIG

Error Code:	TIS_CONFIG
Error Description:	The secondary network clock reference hardware state does not agree with the network clock reference software state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	The audit requests a restore or remove of the Reference so that the hardware is updated to agree with the software. Also a phase lock of the network clock mode is requested.
Manual Action Required:	If the audit rolls in this error, then diagnose the ONTC side which the network clock references is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 79 Error Code: TIS_FE

Error Code:	TIS_FE
Error Description:	The secondary network clock reference was in an out of service family state when the parent ONTC was in service.
Possible Error Effect:	The reference is out of service when it should not be.
Corrective Action Taken:	The reference is restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 80 Error Code: TIS_HDW_INH

Error Code:	TIS_HDW_INH
Error Description:	The secondary network clock reference software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited.
Corrective Action Taken:	The audit requests the hardware interrupt inhibit status to be set to what the software interrupt inhibit status is. If both of the interrupt statuses are inhibit man, the audit changes them to inhibit auto.
Manual Action Required:	If the audit rolls in this error on the same network clock reference, then diagnose the ONTC side which this reference is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 81 Error Code: TIS_IIA

Error Code:	TIS_IIA
Error Description:	The secondary network clock reference software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in the SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 82 Error Code: TIS_PCD

Error Code:	TIS_PCD
Error Description:	A secondary network clock reference has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in the SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 83 Error Code: XC_CONFIG

Error Code:	XC_CONFIG

Error Description:	The network clock cross couple hardware state does not agree with the network clock reference software state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	The audit requests a restore or remove of the Reference so that the hardware is updated to agree with the software.
Manual Action Required:	If the audit rolls in this error, then diagnose the ONTC side which the network clock references is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 84 Error Code: XC_FE

Error Code:	XC_FE
Error Description:	The network clock cross couple reference was in an out of service family state when the parent ONTC was in service.
Possible Error Effect:	The reference is out of service when it should not be.
Corrective Action Taken:	The reference is restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 85 Error Code: XC_HDW_INH

Error Code:	XC_HDW_INH
Error Description:	The network clock cross couple software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited.
Corrective Action Taken:	The audit requests that the hardware interrupt inhibit status be set to what the software interrupt inhibit status is. If both the interrupt statuses are inhibit man, the audit changes them to inhibit auto.
Manual Action Required:	If the audit rolls in this error on the same network clock reference, then diagnose the ONTC side which this reference is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 86 Error Code: XC_IIA

Error Code:	XC_IIA
Error Description:	The network clock cross couple software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in the SMKP.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 87 Error Code: XC_PCD

Error Code:	XC_PCD
Error Description:	A network clock cross couple has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 88 Error Code: XC2_CONFIG

Error Code:	XC2_CONFIG
Error Description:	The network clock 2 mate cross couple hardware state does not agree with the network clock 2 mate cross couple software state.
Possible Error Effect:	The clock signals from the two sides may not be phase aligned properly.
Corrective Action Taken:	The audit requests a restore or remove of the mate cross couple so that the hardware is updated to agree with the software.
Manual Action Required:	Try to remove and restore the cross couple. If the audit continues rolling in this error, diagnose the ONTC side which the network clock 2 cross couple is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else fffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 89 Error Code: XC2_FE

Error Code:	XC2_FE
Error Description:	A network clock 2 mate cross couple has been found to be in an invalid state.
Possible Error Effect:	System timing may be bad.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	Try to remove and restore the cross couple. If the audit rolls in this error, then diagnose the ONTC side which the network clock cross couple is on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else fffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 90 Error Code: XC2_HDW_INH

Error Code:	XC2_HDW_INH
Error Description:	The network 2 clock cross couple software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited. Errors could be masked or allowed to report errors when unexpected.
Corrective Action Taken:	The audit requests that the hardware interrupt inhibit status be set to what the software interrupt inhibit status is. If both the interrupt statuses are inhibit man, the audit changes them to inhibit auto.
Manual Action Required:	If the audit rolls in this error, diagnose the ONTC side which the cross couple is

	on, to see if there is a hardware problem.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 91 Error Code: XC2_IIA

Error Code:	XC2_IIA
Error Description:	The network clock 2 cross couple software interrupt inhibit state is invalid.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited. Errors could be masked or allowed to report errors when unexpected.
Corrective Action Taken:	The interrupt inhibit state is written to a state which is consistent with the state the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 92 Error Code: XC2_PCD

Error Code:	XC2_PCD
Error Description:	A network clock 2 cross couple has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in the SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	If there is more then one unit of this type on the side, for example the MMPs and TMS NCT links, then the units number is printed, or else ffffff is printed if there is only one unit of this type on the side.
<i>Logical Index =</i>	The side the error was found on (that is, 0 or 1).
<i>Dump =</i>	Audit's copy of the data read.

Table 93 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets, and or asserts.
Corrective Action Taken:	Return audit complete; wait for next cycle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	AUpcdindex, the PCD audit cursor.
<i>Dump =</i>	NA.

PCI7LAB

Software Release: 5E10 and later

1. CCS7 POINT CODE INDEX (PCI7LAB) AUDIT

The PCI7LAB audit is responsible for verifying static to dynamic existence of individual static rIPCILST7 tuples to the corresponding dynamic rIPCILST7 tuples. This audit runs in both the switching module (SM) and the administrative module (AM).

2. DATA STRUCTURES

RLPCI7_LAB is a static relation which translates a point code index to a dpc/opc combination for CCS7 trunks. Tuples are inserted or deleted whenever the corresponding tuple in RLLAB7_PCI are inserted or deleted. The RLPCI7_LAB relation is keyed by the point code index (PCI). Corresponding rIPCILST7 tuples are created based on the RLPCI7_LAB relation.

3. AUDIT CHECKS

The PCI7LAB audit invokes the static driver (DRSTATIC) to step through the tuples in RLPCI7_LAB and perform the following checks.

LOSTTUP = Verify that the corresponding rIPCILST7 tuple exists for the PCI7_LAB.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The corresponding rIPCILST7 tuple does not exist for this PCI7_LAB.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rIPCILST7 tuple will be inserted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Relation ID for PCILST7.
<i>Good Data =</i>	DBP7_IDLE_C7 or p ci field value from PCI7_LAB.
<i>Logical Index =</i>	DBP7_IDLE_C7 or p ci field value from PCILST7.
<i>Dump =</i>	NA for "IDLE" PCILST7 but contents of PCI7_LAB for other PCILST7's.

PCILST7

Software Release: 5E10 and later

1. PCI LIST HEAD CELL (PCILST7) AUDIT

The PCILST7 audit is responsible for verifying the linkage data and other semantic data which is stored in the RLPCILST7 relation. This audit runs in both the switching module (SM) and the administrative module (AM). Refer to Figure 1 for a pictorial view of the linkages related to the rIPCILST7 tuple.

2. DATA STRUCTURES

PCILST7 is a dynamic relation which contains the CCS7 message consolidation and acknowledgement handler (MCAH) point code index (PCI) list head cells. Each PCI list contains the MCAH table entries for the PCI maintained in a indexed sorted list (indexing provided by the RLCINDX7 relation).

A special "IDLE" PCI list is used to maintain the idle index (rLCINDX7 tuples) nodes.

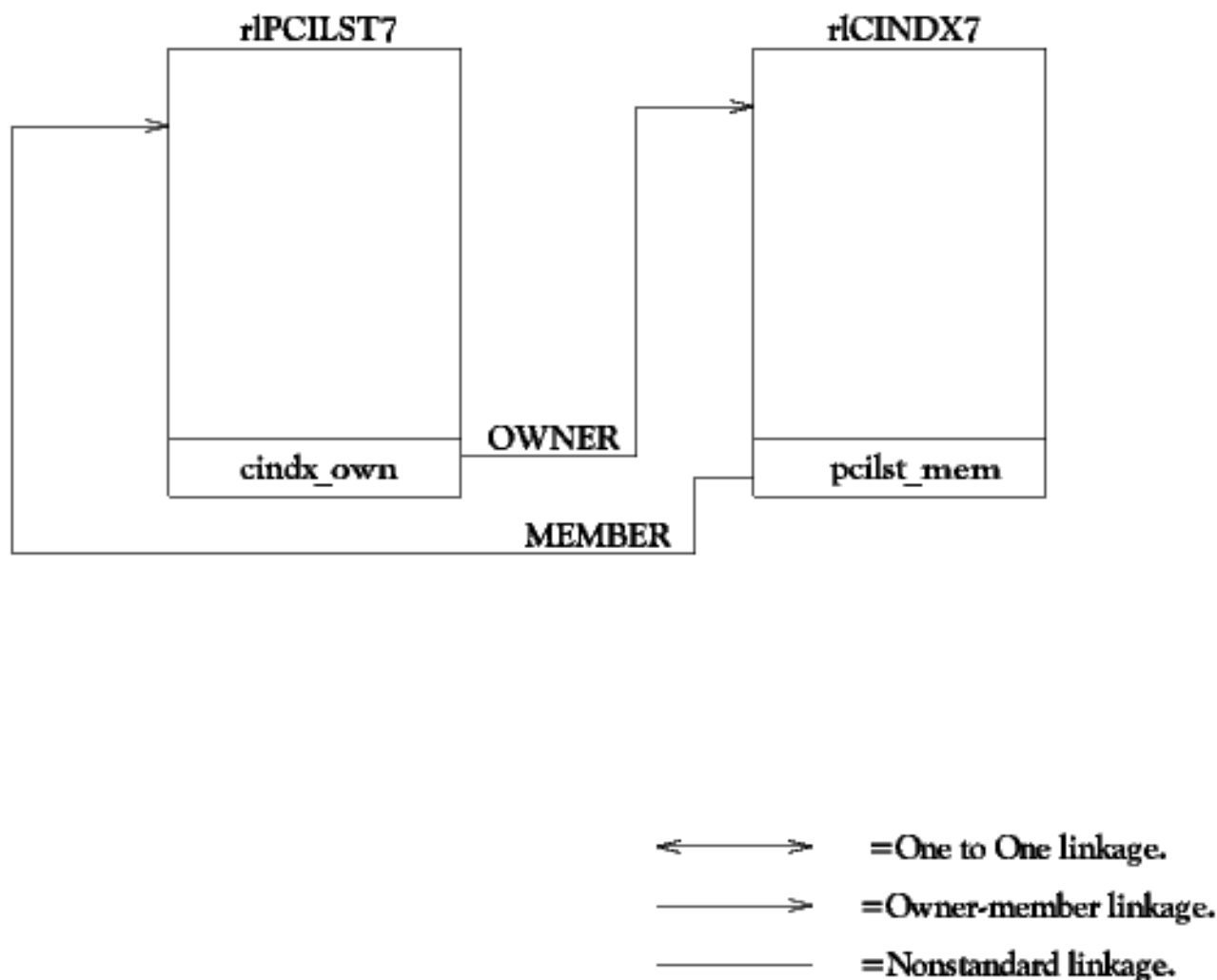


Figure 1 (PCILST7-1) RLPCILST7 Linkages

3. AUDIT CHECKS

The PCILST7 audit uses the DRINDEX audit driver to step through each of the rIPCILST7 tuples and perform the following checks.

- BADDATAR** = For the "IDLE" rIPCILST7 tuple, verify that the *length* field is set to zero. Otherwise, verify the *length* field matches the number of CINDX7's linked to this PCILST7.
- MEMBAD** = Verify that there is at least one CINDX7 linked this PCILST7.
- TWOLINKS** = Verify that the linkages between the CINDX7 and PCILST7 are correct.
- BAD_COUNTS** = After we have cycled through the whole list of member CINDX7's to this PCILST7, verify that the *length* field value in the rIPCILST7 tuple is correct.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	After the audit has traversed through all the member rCINDX7 tuples to this PCILST7, the <i>length</i> field value does not match the number of CINDX7's linked to this PCILST7.
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	The length field will be reset to match the number of CINDX7's linked to this PCILST7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rIPCILST7 tuple.
<i>Bad Data</i> =	Value of <i>length</i> in rIPCILST7 tuple.
<i>Good Data</i> =	Local count of CINDX7's linked to this PCILST7.
<i>Logical Index</i> =	The key of the rIPCILST7 tuple (<i>pci</i>).
<i>Dump</i> =	Contents of the rIPCILST7 tuple being audited.

Table 2 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	If this is the "IDLE" PCILST7, the value of <i>length</i> was found to be not zero. Otherwise, while the audit was traversing through the member rCINDX7 tuples to this PCILST7, the <i>length</i> field value was found to be less than the current number of linked CINDX7's.
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	If this is the "IDLE" PCILST7, then <i>length</i> will be reset to zero. Otherwise, the <i>length</i> field will be reset to match the current number of CINDX7's linked to this PCILST7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rIPCILST7 tuple.
<i>Bad Data</i> =	Current value of <i>length</i> in rIPCILST7 tuple.
<i>Good Data</i> =	Value that <i>length</i> field should be.
<i>Logical Index</i> =	The key of the rIPCILST7 tuple (<i>pci</i>).
<i>Dump</i> =	Contents of the rIPCILST7 tuple being audited.

Table 3 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	There was no rCINDX7 tuple found to be linked to this rIPCILST7 tuple. Every

	rIPCILST7 tuple should have a CINDX7 tuple linked to it.
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	An "IDLE" rCINDX7 tuple will be linked to this PCILST7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rIPCILST7 tuple.
<i>Bad Data</i> =	Linkage field of CINDX7 to PCILST7.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIPCILST7 tuple (<i>pci</i>).
<i>Dump</i> =	Contents of the rIPCILST7 tuple being audited.

Table 4 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	The set member linkages between the PCILST7 and the associated CINDX7 were found to be incorrect.
Possible Error Effect:	Errors in traversal of data.
Corrective Action Taken:	Related audits will run to clean up any other linkage problems which could exist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rIPCILST7 tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIPCILST7 tuple (<i>pci</i>).
<i>Dump</i> =	Contents of the rIPCILST7 tuple being audited.

PCIRTE

Software Release: 5E14 and later

1. POINT CODE ROUTING DATA FOR ISUP (PCIRTE) AUDIT

This audit steps through the point code route relation (RLPCI_RTE) and checks the data in it against the ISUP point code status relation (RLIPCS_PCI), the destination point code status relation (RLDPC_STAT), cluster status relation (RLCLU_STAT) and the network status relation (RLNET_STAT).

2. DATA STRUCTURES

For details on the fields contained in the RLPCI_RTE, RLIPCS_PCI, RLDPC_STAT, RLCLU_STAT and RLNET_STAT relations refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_DPC

Error Code:	BAD_DPC
Error Description:	If Error Address is: 1 = The audit detects that rIPCI_RTE.isup_dpc does not match the rIDPC_STAT.dpc of the owning rIDPC_STAT tuple. 2 = The rIPCI_RTE.isup_dpc does not belong to the rICLU_STAT.cluster of the rICLU_STAT tuple which owns the rIPCI_RTE tuple. 3 = The rIPCI_RTE.isup_dpc does not belong to the rINET_STAT.network of the rINET_STAT tuple which owns the rIPCI_RTE tuple.
Possible Error Effect:	In all cases, the message transfer part (MTP) fails to inform integrated services digital network user part (ISUP) Level 4 about the congestion/accessibility of the destination point code (DPC) associated with the rIPCI_RTE tuple.
Corrective Action Taken:	In all cases, the audit unlinks the rIPCI_RTE tuple from any still valid linkage, sets non-0 fields to the correct values and notifies application code about the recovered tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1 = Corresponds to Error Description. 2 = Corresponds to Error Description. 3 = Corresponds to Error Description.

<i>Bad Data =</i>	rIPCI_RTE.isup_dpc.
<i>Good Data =</i>	If the Error Address is: 1 = rIDPC_STAT.dpc. 2 = Cluster to which the rIPCI_RTE.isup_dpc should belong. 3 = Network to which the rIPCI_RTE.isup_dpc should belong.
<i>Logical Index =</i>	rIPCI_RTE.pci.
<i>Dump 1 =</i>	Contents of rIPCI_RTE tuple.
<i>Dump 2 =</i>	1 = Contents of rIDPC_STAT tuple. 2 = Contents of rICLU_STAT tuple. 3 = Contents of rINET_STAT tuple.

Table 2 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	If Error Address is: 1 = The audit attempts to read the owning rIDPC_STAT tuple of the current rIPCI_RTE tuple but fails. 2 = The audit attempts to read the owning rICLU_STAT tuple of the current rIPCI_RTE tuple but fails. 3 = The audit attempts to read the owning rINET_STAT tuple of the current rIPCI_RTE tuple but fails.
Possible Error Effect:	In all cases, the MTP fails to inform ISUP Level 4 about the <u>congestion/accessibility of the DPC associated with the rIPCI_RTE tuple.</u>
Corrective Action Taken:	In all cases, the audit unlinks the rIPCI_RTE tuple from any still valid linkage, sets non-0 fields to the correct values and notifies application code about the recovered tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1 = Corresponds to Error Description. 2 = Corresponds to Error Description. 3 = Corresponds to Error Description.
<i>Bad Data =</i>	If the Error Address is: 1 = rIDPC_STAT.dpci. 2 = rICLU_STAT.clix. 3 = rINET_STAT.netid.
<i>Good Data =</i>	If the Error Address is: 1 = rIPCI_RTE.routeset.st_ownk - DBSFDPC_STAT. 2 = rIPCI_RTE.routeset.st_ownk - DBSFCLU_STAT.

	3 = rIPCI_RTE.routeset.st_ownk - DBSFNET_STAT.
<i>Logical Index</i> =	rIPCI_RTE.pci.
<i>Dump</i> =	Contents of rIPCI_RTE tuple.

Table 3 Error Code: INV

Error Code:	INV
Error Description:	The audit detects an invalid value in the rIPCI_RTE.rteset_type.
Possible Error Effect:	The MTP fails to inform ISUP Level 4 about the congestion/accessibility of the DPC associated with the rIPCI_RTE tuple.
Corrective Action Taken:	The audit reads the next rIPCI_RTE tuple. If this error rolls, notify Customer Technical Support. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIPCI_RTE.isup_opc.
<i>Bad Data</i> =	rIPCI_RTE.rteset_type.
<i>Good Data</i> =	rIPCI_RTE.isup_dpc.
<i>Logical Index</i> =	rIPCI_RTE.pci.
<i>Dump</i> =	Contents of rIPCI_RTE tuple.

Table 4 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	If Error Address is: 1 = The value of rIPCI_RTE.rteset_type (CCDPC_RTSET) indicates that the rIPCI_RTE tuple should be owned by a rIDPC_STAT tuple but it is not. 2 = The value of rIPCI_RTE.rteset_type (CCCLU_RTSET) indicates that the rIPCI_RTE tuple should be owned by a rICLU_STAT tuple but it is not. 3 = The value of rIPCI_RTE.rteset_type (CCNET_RTSET) indicates that the rIPCI_RTE tuple should be owned by a rINET_STAT tuple but it is not.
Possible Error Effect:	In all cases, the MTP fails to inform ISUP Level 4 about the congestion/accessibility of the DPC associated with the rIPCI_RTE tuple.
Corrective Action Taken:	In all cases, the audit unlinks the rIPCI_RTE tuple from any still valid linkage, sets non-0 fields to the correct values and notifies application code about the recovered tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1 = Corresponds to Error Description. 2 = Corresponds to Error Description. 3 = Corresponds to Error Description.
<i>Bad Data</i> =	rIPCI_RTE.rteset_type.
<i>Good Data</i> =	rIPCI_RTE.isup_dpc.
<i>Logical Index</i> =	rIPCI_RTE.pci.
<i>Dump</i> =	Contents of rIPCI_RTE tuple.

PCITAB

Software Release: 5E11 and later

1. POINT CODE INDEX TO TABLE (PCITAB) AUDIT

PCITAB (point code index to table) audit step through the static relation RLDPC_DPCI (destination point code to destination point code index). For each tuple in the RLDPC_DPCI relation, the audit checks the corresponding CCS array CCpcistat[] to make sure the contents in the array are accurate by comparing with the RLDPC_DPCI and RL GSM_OPC (global SM to origination point code) static relations.

2. DATA STRUCTURES

For details on the fields contained in the RLDPC_DPCI relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DPCI_BAD

Error Code:	DPCI_BAD
Error Description:	The rIDPC_DPCI.pci for a destination point code (DPC) is not null. The destination point code index (DPCI) in CCpci2dpci[] for this PCI should be equal to rIDPC_DPCI.dpcindx but it is not.
Possible Error Effect:	Common channel signaling 7 (CCS7) trunks fail to come into service, and CCS7 calls fail.
Corrective Action Taken:	The audit set CCpci2dpci[rIDPC_DPCI.pci].dpcindx equal to rIDPC_DPCI.dpcindx for the DPC.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIDPC_DPCI.dpc.
<i>Bad Data =</i>	CCpci2dpci[index].
<i>Good Data =</i>	rIDPC_DPCI.dpcindx.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

Table 2 Error Code: GSM_INV

Error Code:	GSM_INV
Error Description:	The error is due to one of the following: 0 On a global switching module (GSM), the gsm at CCpcistat[pci] does not equal the global switching module (SM) number. 1 = The pci in rIDPC_DPCI is null. On a GSM, the gsm at CCpcistat[pci] equals the global SM number. 2 = The pci in rIDPC_DPCI is not null. On a GSM, the gsm at CCpcistat[pci] is

	not null, but cannot be found in the static relation rLGSM_OPC. 3 = On a non-GSM, the gsm at CCpcistat[pci] is not null, but cannot be found in the static relation rLGSM_OPC.
Possible Error Effect:	Corresponds to Error Description as follows: 0 = CCS7 trunks fails to come into service, and CCS7 calls fail. 1 = CCS7 trunks fail to come into service, and CCS7 calls fail. 2 = CCS7 operation may be affected. 3 = CCS7 operation may be affected.
Corrective Action Taken:	Corresponds to Error Description as follows: 0 = The gsm for CCpcistat[pci] is set to current SM. 1 = All fields for CCpcistat[pci] are set to 0 except for the inac_time field which is set to the current time. 2 = All fields for CCpcistat[pci] are set to 0 except for the inac_time field which is set to the current time. 3 = All fields for CCpcistat[pci] are set to 0 except for the inac_time field which is set to the current time.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCpcistat[] array.
<i>Bad Data =</i>	gsm value in CCpcistat[] for all error cases.
<i>Good Data =</i>	Corresponds to Error Description as follows: 0 = Corresponds to the error in error description. 1 = Corresponds to the error in error description. 2 = Corresponds to the error in error description. 3 = Corresponds to the error in error description.
<i>Logical Index =</i>	Index (pci) used to read CCpcistat[] for all error cases.
<i>Dump =</i>	NA.

Table 3 Error Code: GSMI_BAD

Error Code:	GSMI_BAD
Error Description:	The global switching module index in the dynamic point code index status array does not match the corresponding field in the static relation RL GSM_OPC for this global switching module.
Possible Error Effect:	CCS7 trunks fail to come into service, and CCS7 calls fail.
Corrective Action Taken:	The audit corrects the array entry to rLGSM_OPC.gsml.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCpcistat[rIDPC DPCL.pci].
<i>Bad Data =</i>	CCpcistat[rIDPC DPCL.pci].gsml.
<i>Good Data =</i>	rLGSM_OPC.gsml.
<i>Logical Index =</i>	rIDPC DPCL.pci used as the index into the array.
<i>Dump =</i>	NA.

Table 4 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	The stat, gsmi and gsm fields for pci of 0 in CCpcistat[] are not all set to zero. NOTE: Values printed for bad data, good data, and logical key are used for reporting contents of the stat, gsmi and gsm fields for pci 0.
Possible Error Effect:	Bad data may be accessed.
Corrective Action Taken:	The audit sets all the fields for pci of 0 to zero except for the inac_time field which is set to the current time.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCpcistat[].
<i>Bad Data =</i>	Value of CCpcistat[0].stat.
<i>Good Data =</i>	Value of CCpcistat[0].gsmi.
<i>Logical Index =</i>	Value of CCpcistat[0].gsm.
<i>Dump =</i>	NA.

Table 5 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The indices in CCpci2dpci[] correspond to a point code index (PCI). For an index into CCpci2dpci[] for which there is no PCI in relation RLDPC_DPCI (rlDPC_DPCI.pci == 0), CCpci2dpci[] should be 0 but it is not.
Possible Error Effect:	The CCpci2dpci[] translates a PCI to a corresponding DPCI. Application code may try to route to the DPC of the DPCI for which there is no PCI.
Corrective Action Taken:	The audit sets CCpci2dpci[index] equal to 0.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CCpci2dpci[index] entry.
<i>Bad Data =</i>	CCpci2dpci[index].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index.
<i>Dump =</i>	NA.

PCNAM

Software Release: 5E11 and later

1. CALLING NAME FOR PRI (PCNAM) AUDIT

Calling name information delivery-primary rate interface (CNAM-PRI) is one of three components of the calling name for ISDN PRI feature on custom and non-custom ISDN PRI. The CNAM-PRI component provides name information; either name characters, a private indication, or an unavailable indication, to a subscribed PRI interface at the time of call setup. The CNAM-PRI audit verifies the dynamic data relation RLPCNAM relative to the static data relation RLEDLTRKG. For each tuple in RLPCNAM, RLEDLTRKG is searched for an entry with the same "group number". If an entry does not exist, then the entry in RLPCNAM is removed. If an entry does exist, then the audit verifies that the calling name is "allowed" and the automatic message accounting is "Usage Sensitive" in RLEDLTRKG. The RLPCNAM entry is removed if these are not set.

2. DATA STRUCTURES

For details on the fields contained in the RLPCNAM relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

PERIIA-A

Software Release: 5E10 only

1. SWITCHING MODULE INTERRUPT INHIBIT ADMINISTRATOR PERIPHERAL CIRCUIT (PERIIA) AUDIT

The switching module interrupt inhibit administrator (SMIIA) database contains information about all the peripheral circuits in the switching modules (SMs). State information about SM circuits is kept in the switching module equipment status relation (RLSMEST). The SMIIA peripheral circuit audit (PERIIA) ensures that the SMIIA state information in RLSMEST is consistent with the hardware mask registers. The audit also guarantees consistency between the interrupt inhibit administrator state and the in-service/out-of-service state of the peripheral circuits.

The audit consists of a core module and modules for each peripheral unit. The core module works with the peripheral modules to ensure consistent states between circuits on most peripheral units such as the integrated services line unit (ISLU), packet switch unit (PSU), and so forth.

The PERIIA audit is part of the switch maintenance audits subgroup. The core module is part of the BASE software package; the peripheral modules reside in the loadable packages associated with the peripheral units. The audit runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The PERIIA audit uses the RLSMEST relation and the circuit data relation (RLCKTDATA). It also checks the hardware masks for various peripheral units. The data structures are shown in Figure 1.

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) to retrieve tuples from RLSMEST. It makes the following checks:

- (a) If the state attribute is equal to SMALLOW and the bas_stat attribute is SMOOS, the audit reports an error.
- (b) If the state attribute is SMINHIBIT and the qual attribute is SMITEMP, the audit marks sets the touched (iatouch) attribute to DBYES. If the iatouch attribute is DBYES on the next audit cycle, the audit reports an error.
- (c) The audit ensures that the values of the state, qual, and the peripheral circuit inhibit type (pinhtype) attributes are in-range and valid.
- (d) The audit checks the rLSMEST information with that contained in rLCKTDATA.

Once the consistency of the software is verified, the audit ensures that the hardware masks of the peripheral units agree with the SMIIA database. The audit invokes the peripheral modules for each circuit audited regardless of the service state at the time.

4. ERROR HANDLING

The audit corrects errors by making the hardware mask register agree with the software state.

The PERIIA audit schedules no related audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The interrupt inhibit administrator (IIA) current state is allow (SMALLOW), while the circuit is out-of-service (OOS). This combination is illegal because interrupts should never be allowed on an OOS circuit.
Possible Error Effect:	Unwanted interrupts on an out-of-service circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	The current state (SMALLOW).
<i>Good Data =</i>	The correct normal state (SMINHIBIT).
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra equipment status table tuple exists.
Possible Error Effect:	Possible loss of call processing resources.
Corrective Action Taken:	Delete the extra equipment status table tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Equipment status table tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The internal name of the circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 3 Error Code: IL_PTYPE

Error Code:	IL_PTYPE
Error Description:	The IIA peripheral circuit inhibit type (pinhtype) has an unexpected value (not SMINH_ALL or SMINHMAINT). This error code only warns of a possible software problem.
Possible Error Effect:	No system effect.
Corrective Action Taken:	The IIA peripheral circuit inhibit type is corrected based on the service state of the circuit. The peripheral hardware masks are refreshed to match the current IIA state which is stored in rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	Wrong peripheral inhibit type.
<i>Good Data =</i>	Correct peripheral inhibit type.
<i>Logical Index =</i>	Name of the affected circuit.
<i>Dump =</i>	rSMEST tuple.

Table 4 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	This type of error indicates that the IIA current state qualifier is of an unknown type (not SMIA_AUTO, SMIA_MAN, or SMIA_TEMP).
Possible Error Effect:	All IIA requests for the circuit will be rejected. An assert will occur when an IIA action is requested for the circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state information which is stored in the equipment status table.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the Equipment Status Table tuple.
<i>Bad Data</i> =	Faulty current state qualifier.
<i>Good Data</i> =	The new qualifier (SMIA_AUTO).
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

Table 5 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The IIA current state qualifier is temporary (SMIATEMP) for a circuit that is out-of-service. Temporary inhibits are permitted only on in-service circuits. A software problem may exist with regard to the use of temporary inhibits on this circuit.
Possible Error Effect:	No direct system affect.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	Bad current state qualifier (SMIATEMP).
<i>Good Data</i> =	Correct new qualifier (SMIA_AUTO).
<i>Logical Index</i> =	Name of the affected circuit.
<i>Dump</i> =	rSMEST.

Table 6 Error Code: STKTEMP

Error Code:	STKTEMP
Error Description:	A circuit was found stuck in a temporarily inhibited state. This indicates that an inhibit request intended to be temporary has now become permanent. A program which requested a temporary inhibit may have been purged.
Possible Error Effect:	Interrupts are blocked on a circuit when they should be allowed.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the equipment status table tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

PERIIA-B

Software Release: 5E11 only

1. SWITCHING MODULE INTERRUPT INHIBIT ADMINISTRATOR PERIPHERAL CIRCUIT (PERIIA) AUDIT

The switching module interrupt inhibit administrator (SMIIA) database contains information about all the peripheral circuits in the switching modules (SMs). State information about SM circuits is kept in the switching module equipment status relation (RLSMEST). The SMIIA peripheral circuit audit (PERIIA) ensures that the SMIIA state information in RLSMEST is consistent with the hardware mask registers. The audit also guarantees consistency between the interrupt inhibit administrator state and the in-service/out-of-service state of the peripheral circuits.

The audit consists of a core module and modules for each peripheral unit. The core module works with the peripheral modules to ensure consistent states between circuits on most peripheral units such as the integrated services line unit (ISLU), packet switch unit (PSU), and so forth.

The PERIIA audit is part of the Switch Maintenance audits subgroup. The core module is part of the BASE software package; the peripheral modules reside in the loadable packages associated with the peripheral units. The audit runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The PERIIA audit uses the RLSMEST relation and the circuit data relation (RLCKTDATA). It also checks the hardware masks for various peripheral units. The data structures are shown in Figure 1.

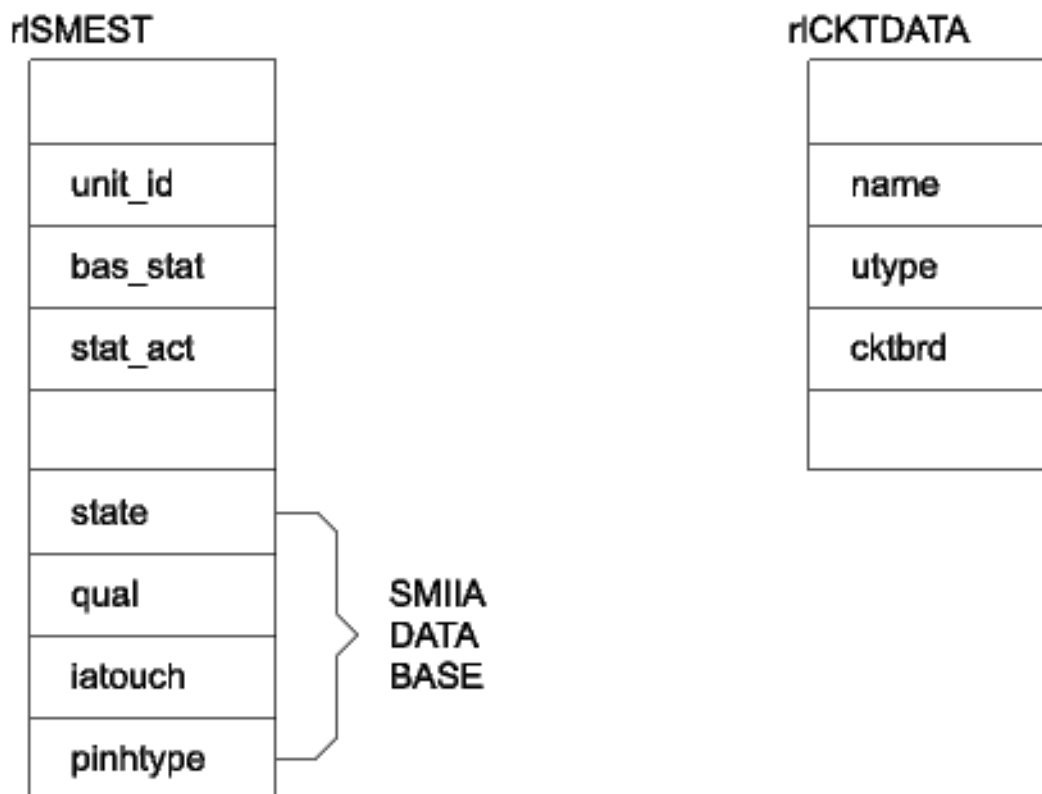


Figure 1 (PERIIA-1) Data Structures of the PERIIA Audit

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) to retrieve tuples from RLSMEST. It makes the following checks:

- (a) If the state attribute is equal to SMALLOW and the bas_stat attribute is SMOOS, the audit reports an error.
- (b) If the state attribute is SMINHIBIT and the qual attribute is SMIA TEMP, the audit marks sets the touched (iatouch) attribute to DBYES. If the iatouch attribute is DBYES on the next audit cycle, the audit reports an error.
- (c) The audit ensures that the values of the state, qual, and the peripheral circuit inhibit type (pinh type) attributes are in-range and valid.
- (d) The audit checks the rLSMEST information with that contained in rICKTDATA.

Once the consistency of the software is verified, the audit ensures that the hardware masks of the peripheral units agree with the SMIIA database. The audit invokes the peripheral modules for each circuit audited regardless of the service state at the time.

4. ERROR HANDLING

The audit corrects errors by making the hardware mask register agree with the software state.

The PERIIA audit schedules no related audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The interrupt inhibit administrator (IIA) current state is allow (SMALLOW), while the circuit is out-of-service (OOS). This combination is illegal because interrupts should never be allowed on an OOS circuit.
Possible Error Effect:	Unwanted interrupts on an out-of-service circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	The current state (SMALLOW).
<i>Good Data =</i>	The correct normal state (SMINHIBIT).
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra equipment status table tuple exists.
Possible Error Effect:	Possible loss of call processing resources.
Corrective Action Taken:	Delete the extra equipment status table tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Equipment status table tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The internal name of the circuit.

<i>Dump</i> =	Equipment status table tuple.

Table 3 Error Code: IL_PTYPE

Error Code:	IL_PTYPE
Error Description:	The IIA peripheral circuit inhibit type (pinhype) has an unexpected value (not SMINH_ALL or SMINHMAINT). This error code only warns of a possible software problem.
Possible Error Effect:	No system effect.
Corrective Action Taken:	The IIA peripheral circuit inhibit type is corrected based on the service state of the circuit. The peripheral hardware masks are refreshed to match the current IIA state which is stored in rISMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	Wrong peripheral inhibit type.
<i>Good Data</i> =	Correct peripheral inhibit type.
<i>Logical Index</i> =	Name of the affected circuit.
<i>Dump</i> =	rISMEST tuple.

Table 4 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	This type of error indicates that the IIA current state qualifier is of an unknown type (not SMIA_AUTO, SMIA_MAN, or SMIA_TEMP).
Possible Error Effect:	All IIA requests for the circuit will be rejected. An assert will occur when an IIA action is requested for the circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state information which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the equipment status table tuple.
<i>Bad Data</i> =	Faulty current state qualifier.
<i>Good Data</i> =	The new qualifier (SMIA_AUTO).
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

Table 5 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The IIA current state qualifier is temporary (SMIA_TEMP) for a circuit that is out-of-service. Temporary inhibits are permitted only on in-service circuits. A software problem may exist with regard to the use of temporary inhibits on this circuit.
Possible Error Effect:	No direct system affect.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in rISMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	Bad current state qualifier (SMIA_TEMP).
<i>Good Data</i> =	Correct new qualifier (SMIA_AUTO).
<i>Logical Index</i> =	Name of the affected circuit.
<i>Dump</i> =	rISMEST.

Table 6 Error Code: STKTEMP

Error Code:	STKTEMP

Error Description:	A circuit was found stuck in a temporarily inhibited state. This indicates that an inhibit request intended to be temporary has now become permanent. A program which requested a temporary inhibit may have been purged.
Possible Error Effect:	Interrupts are blocked on a circuit when they should be allowed.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the equipment status table tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

PERIIA-C

Software Release: 5E12 only

1. SWITCHING MODULE INTERRUPT INHIBIT ADMINISTRATOR PERIPHERAL CIRCUIT (PERIIA) AUDIT

The switching module interrupt inhibit administrator (SMIIA) database contains information about all the peripheral circuits in the switching modules (SMs). State information about SM circuits is kept in the switching module equipment status relation (RLSMEST). The SMIIA peripheral circuit audit (PERIIA) ensures that the SMIIA state information in RLSMEST is consistent with the hardware mask registers. The audit also guarantees consistency between the interrupt inhibit administrator state and the in-service/out-of-service state of the peripheral circuits.

The audit consists of a core module and modules for each peripheral unit. The core module works with the peripheral modules to ensure consistent states between circuits on most peripheral units such as the integrated services line unit (ISLU), packet switch unit (PSU), and so forth.

The PERIIA audit is part of the Switch Maintenance audits subgroup. The core module is part of the BASE software package; the peripheral modules reside in the loadable packages associated with the peripheral units. The audit runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

For details on the fields contained in the rLSMEST and rICKTDATA structures, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below. The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The interrupt inhibit administrator (IIA) current state is allow (SMALLOW), while the circuit is out-of-service (OOS). This combination is illegal because interrupts should never be allowed on an OOS circuit.
Possible Error Effect:	Unwanted interrupts on an out-of-service circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	The current state (SMALLOW).
<i>Good Data =</i>	The correct normal state (SMINHIBIT).
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP

Error Description:	An extra equipment status table tuple exists.
Possible Error Effect:	Possible loss of call processing resources.
Corrective Action Taken:	Delete the extra equipment status table tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Equipment status table tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The internal name of the circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 3 Error Code: IL_PTYPE

Error Code:	IL_PTYPE
Error Description:	The IIA peripheral circuit inhibit type (pinhtype) has an unexpected value (not SMINH_ALL or SMINHMAINT). This error code only warns of a possible software problem.
Possible Error Effect:	No system effect.
Corrective Action Taken:	The IIA peripheral circuit inhibit type is corrected based on the service state of the circuit. The peripheral hardware masks are refreshed to match the current IIA state which is stored in rISMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rISMEST tuple.
<i>Bad Data =</i>	Wrong peripheral inhibit type.
<i>Good Data =</i>	Correct peripheral inhibit type.
<i>Logical Index =</i>	Name of the affected circuit.
<i>Dump =</i>	rISMEST tuple.

Table 4 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	A circuit is in an allowed state when the unit is inhibited.
Possible Error Effect:	Interrupts are allowed on a circuit when they should be inhibited.
Corrective Action Taken:	The IIA (interrupt inhibit administrator) current state is updated to the normal state of inhibit. The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	Value of the unit's IIA state.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 5 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	This type of error indicates that the IIA current state qualifier is of an unknown type (not SMIA_AUTO, SMIA_MAN, or SMIATEMP).
Possible Error Effect:	All IIA requests for the circuit will be rejected. An assert will occur when an IIA action is requested for the circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state information which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	Faulty current state qualifier.
<i>Good Data =</i>	The new qualifier (SMIA_AUTO).
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 6 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The IIA current state qualifier is temporary (SMIATEMP) for a circuit that is out-of-service. Temporary inhibits are permitted only on in-service circuits. A software problem may exist with regard to the use of temporary inhibits on this circuit.
Possible Error Effect:	No direct system affect.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	Bad current state qualifier (SMIATEMP).
<i>Good Data</i> =	Correct new qualifier (SMIA_AUTO).
<i>Logical Index</i> =	Name of the affected circuit.
<i>Dump</i> =	rSMEST.

Table 7 Error Code: SALW

Error Code:	SALW
Error Description:	A circuit is in an inhibited state when the unit is allowed.
Possible Error Effect:	Interrupts are blocked on a circuit when they should be allowed.
Corrective Action Taken:	The IIA (interrupt inhibit administrator) current state is updated to the normal state of allow (SMALLOW). The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the equipment status table tuple.
<i>Bad Data</i> =	Value of the unit's IIA state.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

Table 8 Error Code: STKTEMP

Error Code:	STKTEMP
Error Description:	A circuit was found stuck in a temporarily inhibited state. This indicates that an inhibit request intended to be temporary has now become permanent. A program which requested a temporary inhibit may have been purged.
Possible Error Effect:	Interrupts are blocked on a circuit when they should be allowed.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the equipment status table tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

PERIIA-D

Software Release: 5E13 and later

1. SWITCHING MODULE INTERRUPT INHIBIT ADMINISTRATOR PERIPHERAL CIRCUIT (PERIIA) AUDIT

The switching module interrupt inhibit administrator (SMIIA) database contains information about all the peripheral circuits in the switching modules (SMs). State information about SM circuits is kept in the switching module equipment status relation (RLSMEST). The SMIIA peripheral circuit audit (PERIIA) ensures that the SMIIA state information in RLSMEST is consistent with the hardware mask registers. The audit also guarantees consistency between the interrupt inhibit administrator state and the in-service/out-of-service state of the peripheral circuits.

The audit consists of a core module and modules for each peripheral unit. The core module works with the peripheral modules to ensure consistent states between circuits on most peripheral units such as the integrated services line unit (ISLU), packet switch unit (PSU), and so forth.

The PERIIA audit is part of the Switch Maintenance audits subgroup. The core module is part of the BASE software package; the peripheral modules reside in the loadable packages associated with the peripheral units. The audit runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

For details on the fields contained in the rLSMEST and rICKTDATA structures, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below. The audit produces the following error code(s).

Table 1 Error Code: ALWOOS

Error Code:	ALWOOS
Error Description:	The interrupt inhibit administrator (IIA) current state is allow (SMALLOW), while the circuit is out-of-service (OOS). This combination is illegal because interrupts should never be allowed on an OOS circuit.
Possible Error Effect:	Unwanted interrupts on an out-of-service circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the equipment status table tuple.
<i>Bad Data</i> =	The current state (SMALLOW).
<i>Good Data</i> =	The correct normal state (SMINHIBIT).
<i>Logical Index</i> =	The name of the affected circuit.
<i>Dump</i> =	Equipment status table tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP

Error Description:	An extra equipment status table tuple exists.
Possible Error Effect:	Possible loss of call processing resources.
Corrective Action Taken:	Delete the extra equipment status table tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Equipment status table tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The internal name of the circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 3 Error Code: IL_PTYPE

Error Code:	IL_PTYPE
Error Description:	The IIA peripheral circuit inhibit type (pinhntype) has an unexpected value (not SMINH_ALL or SMINHMAINT). This error code only warns of a possible software problem.
Possible Error Effect:	No system effect.
Corrective Action Taken:	The IIA peripheral circuit inhibit type is corrected based on the service state of the circuit. The peripheral hardware masks are refreshed to match the current IIA state which is stored in rISMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rISMEST tuple.
<i>Bad Data =</i>	Wrong peripheral inhibit type.
<i>Good Data =</i>	Correct peripheral inhibit type.
<i>Logical Index =</i>	Name of the affected circuit.
<i>Dump =</i>	rISMEST tuple.

Table 4 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	A circuit is in an allowed state when the unit is inhibited.
Possible Error Effect:	Interrupts are allowed on a circuit when they should be inhibited.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit. The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	Value of the unit's IIA state.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 5 Error Code: M_UNKQUAL

Error Code:	M_UNKQUAL
Error Description:	This type of error indicates that the IIA current state qualifier is of an unknown type (not SMIA_AUTO, SMIA_MAN, or SMIATEMP).
Possible Error Effect:	All IIA requests for the circuit will be rejected. An assert will occur when an IIA action is requested for the circuit.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state information which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	Faulty current state qualifier.
<i>Good Data =</i>	The new qualifier (SMIA_AUTO).
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 6 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The IIA current state qualifier is temporary (SMIATEMP) for a circuit that is out-of-service. Temporary inhibits are permitted only on in-service circuits. A software problem may exist with regard to the use of temporary inhibits on this circuit.
Possible Error Effect:	No direct system affect.
Corrective Action Taken:	The IIA current state is updated to the normal state of inhibit (SMINHIBIT). The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in rISMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	Bad current state qualifier (SMIATEMP).
<i>Good Data</i> =	Correct new qualifier (SMIA_AUTO).
<i>Logical Index</i> =	Name of the affected circuit.
<i>Dump</i> =	rISMEST.

Table 7 Error Code: PHERRL

Error Code:	PHERRL
Error Description:	Software detected a problem with the PSU PH error lead mask bits. There is an inconsistency between hardware (the bitmap returned from PCpsiucmd()) and software (data transport and SMph_dslgrp[]). This indicates that PH(s) error leads are erroneously masked or unmasked. The <i>Error Address</i> in the Dump Description indicates the following: 0 = PSUCOM 0 is act and PSUCOM 0 is being audited. 1 = PSUCOM 0 is act and PSUCOM 1 is being audited. 2 = PSUCOM 1 is act and PSUCOM 0 is being audited. 3 = PSUCOM 1 is act and PSUCOM 1 is being audited. 4 = neither PSUCOM is act - Illegal state.
Possible Error Effect:	PH(s) could be in service but have their error lead masked. This could prohibit real errors from being reported. Also PH(s) could be OOS with its error lead unmasked, which could cause false error reports.
Corrective Action Taken:	The PSUPH error lead masks are refreshed to match the current software state which is stored in SMph_dslgrp[] and DT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The active PSUCOM and the PSUCOM being audited.
<i>Bad Data</i> =	Incorrect PH error lead mask bits.
<i>Good Data</i> =	Correct PH error lead mask bits.
<i>Logical Index</i> =	The PSU shelf number.
<i>Dump</i> =	None.

Table 8 Error Code: PHPBE

Error Code:	PHPBE
Error Description:	Software has detected a problem with the PSU PH packet bus enables. There is an inconsistency between hardware (the bitmap returned from PCpsiucmd()) and software (data transport and SMph_dslgrp[]). This indicates that PH(s) have transitioned from being enabled on the packet bus to being disabled or vice versa. The <i>Error Address</i> in the Dump Description indicates the following:

	0 = PSUCOM 0 is act and PSUCOM 0 is being audited. 1 = PSUCOM 0 is act and PSUCOM 1 is being audited. 2 = PSUCOM 1 is act and PSUCOM 0 is being audited. 3 = PSUCOM 1 is act and PSUCOM 1 is being audited. 4 = neither PSUCOM is act - Illegal state.
Possible Error Effect:	PH(s) could be in service and not allowed onto the packet bus. This could cause PH LOSS OF LAYER 2 errors. Also PH(s) could be OOS and allowed onto the packet bus which could cause false error reports.
Corrective Action Taken:	The PSUPH PB enable masks are refreshed to match the current software state which is stored in SMph_dslgrp[] and DT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The active PSUCOM and the PSUCOM being audited.
<i>Bad Data</i> =	Incorrect packet bus enable mask bits.
<i>Good Data</i> =	Correct packet bus enable mask bits.
<i>Logical Index</i> =	The PSU shelf number.
<i>Dump</i> =	None.

Table 9 Error Code: PSUPBE

Error Code:	PSUPBE
Error Description:	Software detected a problem with the PSU packet bus enables. There is an inconsistency between hardware (the bitmap returned from PCpsiucmd()) and software (SMmcustat[] and stshlf[] - shelf status). This indicates that PF(s) or PI(s) have transitioned from being enabled on the packet bus to being disabled or vice versa. The <i>Error Address</i> in the Dump Description indicates the following: 0 = PSUCOM 0 is act and PSUCOM 0 is being audited. 1 = PSUCOM 0 is act and PSUCOM 1 is being audited. 2 = PSUCOM 1 is act and PSUCOM 0 is being audited. 3 = PSUCOM 1 is act and PSUCOM 1 is being audited. 4 = neither PSUCOM is act - Illegal state.
Possible Error Effect:	PF(s) or PI(s) could be in service and not allowed onto the packet bus. This could cause PI or PSU LOSS OF LAYER 2 errors. Also PF(s) or PI(s) could be OOS and allowed onto the packet bus which could cause false error reports.
Corrective Action Taken:	The PSUCOM PB enable masks are refreshed to match the current software state which is stored in SMmcustat[] and stshlf[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The active PSUCOM and the PSUCOM being audited.
<i>Bad Data</i> =	Incorrect packet bus enable mask bits.
<i>Good Data</i> =	Correct packet bus enable mask bits.
<i>Logical Index</i> =	The packet bus enable bits that have changed.
<i>Dump</i> =	None.

Table 10 Error Code: SALW

Error Code:	SALW
Error Description:	A circuit is in an inhibited state when the unit is allowed.
Possible Error Effect:	Interrupts are blocked on a circuit when they should be allowed.
Corrective Action Taken:	The IIA (interrupt inhibit administrator) current state is updated to the normal

	state of allow (SMALLOW). The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	Value of the unit's IIA state.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

Table 11 Error Code: STKTEMP

Error Code:	STKTEMP
Error Description:	A circuit was found stuck in a temporarily inhibited state. This indicates that an inhibit request intended to be temporary has now become permanent. A program which requested a temporary inhibit may have been purged.
Possible Error Effect:	Interrupts are blocked on a circuit when they should be allowed.
Corrective Action Taken:	The IIA current state is updated to the normal state of either allow (SMALLOW) or inhibit (SMINHIBIT), where the normal state depends upon the service state of the circuit. The IIA current state qualifier is updated to auto (SMIA_AUTO). The peripheral hardware masks are refreshed to match the current IIA state which is stored in the equipment status table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the equipment status table tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The name of the affected circuit.
<i>Dump =</i>	Equipment status table tuple.

25. PF - PO

PHDATA-A

Software Release: 5E10 - 5E11

1. PATH HUNT DATA (PHDATA) AUDIT

The purpose of the PHDATA audit is to verify the integrity of the Glphdata array. Glphdata array is a dynamic data structure and is made up of rIPHDATA structures.

The rIPHDATA data structure contains information about time multiplexed switch (TMS) network control and timing (NCT) link numbers and link status for a switching module (SM). When a SM is grown, a rIPHDATA tuple is created from static relation RLMODATT through a DB trigger and stored in Glphdata array. The audit verifies that the information contained in the rIPHDATA tuple for the SM is consistent with the information in the rIMODATT tuple.

The PHDATA audit runs in the communication module processor (CMP) environment in the segmented mode.

2. DATA STRUCTURES

This audit uses three data structures: Glphdata, rIPHDATA and rIMODATT. Glphdata array is an array of rIPHDATA structures, one entry per SM. The rIPHDATA structure is used to store TMS NCT information about the SM, copied from the static RLMODATT relation. The rIPHDATA structure has a number of fields, however, PHDATA audit checks only four of them. The audited fields are, *hostnum* (for RSM only), *clid* (cluster ID - for SM in MMRS), *tmssevenlnk* (one per SM - contains even NCT link number) and *tmsoddlnk* (one per SM - contains odd NCT link number). The rest of the information is periodically refreshed by operational code [peripheral control (PC)]. The following data structures are used for SM-2000 processors only:

RLNLI_LNK	= The SM and (E)NLI to TMSLNK mapping relation.
<i>sm</i>	= A field in the RLNLI_LNK relation representing the SM number.
<i>nli</i>	= A field in the RLNLI_LNK relation representing the (E)NLI number.
<i>tmslnk_0</i>	= A field in the RLNLI_LNK relation representing the first TMSLNK connected to the NLI or the only TMSLNK for an ENLI.
<i>tmslnk_1</i>	= A field in the RLNLI_LNK relation representing the second TMSLNK connected to the NLI or PCNULLTMS for an ENLI.
RLPHDATA	= A data structure containing network link and time slot information related to an SM.
<i>eq_lis</i>	= A field in the RLMODATT relation representing the number of equipped LIs for a particular SM DLI, NLI, or ENLI.

3. AUDIT CHECKS

PHDATA audit uses the following error codes. For details about each error code see the comment blocks given. If the audit is just starting:

ZAP	= The audit will take this error as a result of a write protect, out of range, or invalid operation by any audit system process.

For all the SMs that are equipped, get rIMODATT tuple for that SM and do following checks:

NUMELEM	= Check <i>hostnum</i> field.

CLID = Check *clid* field.

EVEN_NCT_LINK = Check *tmsevenlnk* field.

ODD_NCT_LINK = Check *tmsoddlnk* field.

4. ERROR HANDLING

PHDATA audit does not use any audit drivers.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The audit found that <i>rhgmax</i> field of the current PCphdata[] array entry doesn't match the number of network control and timing (NCT) links to the switching module (SM) as determined from the <i>enli_flag</i> and <i>sm2000</i> fields.
Possible Error Effect:	The call associated with this entry has may trying to use a NCT link that doesn't exist.
Corrective Action Taken:	The audit reports the error and writes the correct value of the <i>rhgmax</i> field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PCphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>rhgmax</i> filed that did not agree with the value determined from the <i>enli_flag</i> , <i>sm2000</i> and <i>eq_lis</i> fields of the PCphdata array entry.
<i>Good Data</i> =	The correct value as determined.
<i>Logical Index</i> =	The value of loop variable <i>sm_num</i> . This variable indeed represents the number of the PCphdata array entry currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple.
<i>Dump 1</i> =	The PCphdata array entry, of type rIPHDATA structure, currently being audited.
<i>Dump 2</i> =	The rIMODATT tuple associated with <i>sm_num</i> .

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The audit found that the <i>sm2000</i> field of the current PCphdata[] array entry has a bad value.
Possible Error Effect:	The call associated with this entry indicates the switching module (SM) is or is not an SM-2000 when the static data in rIMODATT indicates otherwise.
Corrective Action Taken:	The audit reports the error and writes the correct value of the <i>sm2000</i> field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PCphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>sm2000</i> field that did not agree with the same value in the <i>unitttype</i> field of the rIMODATT tuple.
<i>Good Data</i> =	The value of the <i>sm2000</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The value of loop variable <i>sm_num</i> . This variable represents the number of the PCphdata array entry currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple.
<i>Dump 1</i> =	The PCphdata array entry, of type rIPHDATA structure, currently being audited.
<i>Dump 2</i> =	The rIMODATT tuple associated with <i>sm_num</i> .

Table 3 Error Code: CLID

Error Code:	CLID
Error Description:	The audit found that the <i>clid</i> field of the current Giphdata entry representing a SM has a bad value. This means that the current entry should be that of a SM in a MMRSM and its <i>clid</i> field has a different value than found in rIMODATT tuple. It is also possible that this entry is not for a SM in MMRSM and the value of the

	<i>clid</i> is corrupted.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This audit will report the error and correct the value of <i>clid</i> field taking it from the rIMODATT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the Glnphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>clid</i> field that did not match the same field in the rIMODATT tuple.
<i>Good Data</i> =	The value of <i>clid</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The number of the Glnphdata array currently being audited. It also represents the module field (key) of the associated rIMODATT tuple or the SM number.
<i>Dump</i> =	The Glnphdata array entry, of type rIPHDATA structure, currently being audited.

Table 4 Error Code: EVEN_NCT_LINK

Error Code:	EVEN_NCT_LINK
Error Description:	The audit found that the <i>tmsevenlnk</i> field of the current Glnphdata entry representing a SM has a bad value. This means that the current entry has a different value than found in <i>tmsevennct</i> field of rIMODATT tuple.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This audit will report the error and correct the value of <i>tmsevenlnk</i> field taking it from the <i>tmsevennct</i> field of the rIMODATT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the Glnphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>tmsevenlnk</i> field that did not match the value in <i>tmsevennct</i> field of the rIMODATT tuple.
<i>Good Data</i> =	The value of <i>tmsevennct</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The number of the Glnphdata[] array currently being audited. It also represents the module field (key) of the associated rIMODATT tuple or the SM number.
<i>Dump</i> =	The Glnphdata array entry, of type rIPHDATA structure, currently being audited.

Table 5 Error Code: ODD_NCT_LINK

Error Code:	ODD_NCT_LINK
Error Description:	The audit found that the <i>tmsoddlnk</i> field of the current Glnphdata entry representing a SM has a bad value. This means that the current entry has a different value than found in <i>tmsoddnct</i> field of rIMODATT.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This audit will report the error and correct the value of <i>tmsoddlnk</i> field taking it from the <i>tmsoddnct</i> field of the rIMODATT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the Glnphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>tmsoddlnk</i> field that did not match the value in <i>tmsoddnct</i> field of the rIMODATT tuple.
<i>Good Data</i> =	The value of <i>tmsoddnct</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The number of the Glnphdata array currently being audited. It also represents the module field (key) of the associated rIMODATT tuple or the SM number.
<i>Dump</i> =	The Glnphdata array entry, of type rIPHDATA structure, currently being audited.

Table 6 Error Code: NLI_LNK

Error Code:	NLI_LNK
Error Description:	The audit found that the <i>tmslnk_0</i> or <i>tmslnk_1</i> field of the current PCphdata[] array entry doesn't match the appropriate oddlinks[] or evenlinks[] array entry of the current PCphdata[] array entry. This error can only occur on a SM-2000.
Possible Error Effect:	The call associated with this entry may be trying to use the wrong link.
Corrective Action Taken:	The audit reports the error and writes the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PCphdata array entry currently being audited.

<i>Bad Data</i> =	The value of the loop variable <i>nli</i> . This variable is used with the <i>enli_flag</i> to determine which entry in the <i>oddlinks</i> or <i>evenlinks</i> array in the current PCphdata array entry should be compared against.
<i>Good Data</i> =	A 1 or 2 indicates the <i>enli_flag</i> wasn't set. A 3 or 4 indicates the <i>enli_flag</i> was set. A 1 or 3 indicates that the <i>tmslnk_0</i> and <i>tmslnk_1</i> were compared against the <i>evenlinks</i> array. A 2 or 4 indicates that they were compared against <i>oddlinks</i> .
<i>Logical Index</i> =	The value of loop variable <i>sm_num</i> . This variable represents the number of the PCphdata array entry currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple.
<i>Dump 1</i> =	The PCphdata array entry, of type rIPHDATA structure, currently being audited.
<i>Dump 2</i> =	The rINLI_LNK tuple associated with <i>nli</i> .
<i>Dump 3</i> =	The rIMODATT tuple associated with <i>sm_num</i> .

Table 7 Error Code: NUMELEM

Error Code:	NUMELEM
Error Description:	The audit found that the <i>hostnum</i> field of the current Gphdata entry representing a SM has a bad value. This means that the current entry should be that of an RSM and its <i>hostnum</i> field has a different value than found in rIMODATT tuple. It is also possible that this entry is not for an RSM and the value of the <i>hostnum</i> is corrupted.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This audit will report the error and correct the value of <i>hostnum</i> field taking it from the rIMODATT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the Gphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>hostnum</i> field of rIPHDATA tuple that did not match the same field in the rIMODATT tuple.
<i>Good Data</i> =	The value of <i>hostnum</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The number of the Gphdata[] array currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple or the SM number.
<i>Dump</i> =	The Gphdata array entry, of type rIPHDATA structure, currently being audited.

Table 8 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The audit failed to read a static rIMODATT tuple or a static rINLI_LNK tuple for the switching module (SM) which is equipped (in service).
Possible Error Effect:	There is an office-dependent data (ODD) problem for this SM.
Corrective Action Taken:	The audit reports the error and schedules the DD audit to recover any DD key errors if there are any, because by reading the DD bits the audit decides whether the SM is in operation or not.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0 or the <i>nli</i> number used to read the rINLI_LNK tuple.
<i>Bad Data</i> =	The return value of the DB read for tuple.
<i>Good Data</i> =	0 or 1 for MODATT or NLI_LNK respectively.
<i>Logical Index</i> =	The value of loop variable <i>sm_num</i> . This variable represents the number of the PCphdata array entry currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple.
<i>Dump</i> =	The rIMODATT tuple used to read the rINLI_LNK tuple, in the NLI_LNK case (GOOD DATA = 1).

Table 9 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit found that <i>rhgsel</i> field of the current PCphdata[] array entry has exceeded the number of network control and timing (NCT) links to the SM as determined from the <i>enli_flag</i> and <i>sm2000</i> fields.
Possible Error Effect:	The call associated with this entry may be trying to use a NCT link that doesn't

	exist.
Corrective Action Taken:	The audit reports the error and writes the correct value of the <i>rhgsel</i> field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PCphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>rhgsel</i> field that did not agree with the value determined from the <i>enli_flag</i> , <i>sm2000</i> and <i>eq_lis</i> fields of the PCphdata array entry.
<i>Good Data</i> =	The correct value as determined.
<i>Logical Index</i> =	The value of loop variable <i>sm_num</i> . This variable represents the number of the PCphdata array entry currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple.
<i>Dump 1</i> =	The PCphdata array entry, of type rIPHDATA structure, currently being audited.
<i>Dump 2</i> =	The rIMODATT tuple associated with <i>sm_num</i> .

Table 10 Error Code: RMT_FLAG

Error Code:	RMT_FLAG
Error Description:	The audit found that <i>enli_flag</i> field of the current PCphdata[] array entry has a bad value.
Possible Error Effect:	The call associated with this entry indicates an enli on the SM and/or non optically integrated remote switching module (ORM) has an enli (or lack of an enli on SM-2000 ORM) when the rIMODATT tuple indicates otherwise.
Corrective Action Taken:	The audit report the error and will write the correct value of the <i>enli_flag</i> field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PCphdata array entry currently being audited.
<i>Bad Data</i> =	The value of the <i>enli_flag</i> field that did agree with the values in <i>imtype</i> and <i>unitttype</i> fields of the rIMODATT tuple.
<i>Good Data</i> =	The value of <i>imtype</i> field of the rIMODATT tuple.
<i>Logical Index</i> =	The value of loop variable <i>sm_num</i> . This variable indeed represents the number of the PCphdata array entry currently being audited. It also represents the <i>module</i> field (key) of the associated rIMODATT tuple.
<i>Dump 1</i> =	The PCphdata array entry, of type rIPHDATA structure, currently being audited.
<i>Dump 2</i> =	The rIMODATT tuple associated with <i>sm_num</i> .

Table 11 Error Code: ZAP

Error Code:	ZAP
Error Description:	When the PHDATA audit was running, the audit system process took an interrupt and it was purged.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This error will be printed on the receive-only printer (ROP) and the audit will return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0 (Unknown here).
<i>Bad Data</i> =	0 (Unknown here).
<i>Good Data</i> =	0 (Unknown here).
<i>Logical Index</i> =	0 (Unknown here).
<i>Dump</i> =	None.

PHDATA-B

Software Release: 5E12 and later

1. PATH HUNT DATA (PHDATA) AUDIT

The purpose of the PHDATA audit is to verify the integrity of the PCphdata[] array in the CMP and SMphdata[] array in the SMKP. PCphdata[] array is a dynamic data structure and is made up of rIPHDATA structures.

The rIPHDATA data structure contains information about time multiplexed switch (TMS) network control and timing (NCT) link numbers and link status for a switching module (SM).

2. DATA STRUCTURES

For details on the fields contained in the rIPHDATA structure refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

PHDATA audit does not use any audit drivers.

The audit produces the following error code(s).

Table 1 Error Code: BAD_BIT_SET

Error Code:	BAD_BIT_SET
Error Description:	The audit found that litype_map or/and rhgmap bits in a PHDATA entry greater than or equal to rIMODATT.eq_lis (that is, number of equipped LIs) were not set to clear.
Possible Error Effect:	The call associated with this entry may be trying to use an unequipped (E)NLI or/and RHG.
Corrective Action Taken:	The audit reports the error and clears the bits in this PHDATA entry greater than or equal to rIMODATT.eq_lis (that is, number of equipped LIs).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The value of the rIMODATT.eq_lis field.
<i>Bad Data =</i>	The value of the rIPHDATA.litype_map field.
<i>Good Data =</i>	0
<i>Logical Index =</i>	The index into the PHDATA array.
<i>Dump 1 =</i>	The PHDATA array entry, of type rIPHDATA structure, currently being audited.
<i>Dump 2 =</i>	The rIMODATT tuple.

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The audit found that the sm2000 field of the current PHDATA array entry was inconsistent with the static rIMODATT tuple.
Possible Error Effect:	The call associated with this entry indicates the switching module (SM) is or is not an SM-2000 when the static data in rIMODATT indicates otherwise.
Corrective Action Taken:	The audit reports the error and writes the correct value of the sm2000 field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The beginning address of the PHDATA array entry currently being audited.
<i>Bad Data =</i>	PHDATA.sm2000

<i>Good Data</i> =	The value of the unitttype field of the rIMODATT tuple.
<i>Logical Index</i> =	The index into the PHDATA index.
<i>Dump 1</i> =	The PHDATA array entry.
<i>Dump 2</i> =	The rIMODATT tuple.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit found that enli_flag field of the current PHDATA[] array entry has a bad value. If litype_map=0 then enli_flag is DBYES otherwise it is DBNO except for VCDX office enli_flag is always DBNO and and litype_map is always 0.
Possible Error Effect:	Varies according to condition. Maintenance or diagnostics failure.
Corrective Action Taken:	The audit reports the error and writes the correct value based on condition given in error description.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0
<i>Bad Data</i> =	The value of the rIPHDATA.enli_flag field.
<i>Good Data</i> =	The value of the rIPHDATA.litype_map field.
<i>Logical Index</i> =	The index into the PHDATA array.
<i>Dump 1</i> =	The PHDATA array entry.
<i>Dump 2</i> =	The rIMODATT tuple.

Table 4 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	When one of the following is happen for non-SM-2000: - The audit found that an entry in rIPHDATA, oddlinks[] or evenlinks[] didn't match the appropriate tmsevennct or tmsoddnct field of the rIMODATT. - The PHDATA.rhgmap or PHDATA.litype_map value is invalid.
Possible Error Effect:	The call associated with this entry may be trying to use the wrong link.
Corrective Action Taken:	The audit reports the error and writes the correct value of the fields based on the rIMODATT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of the rIMODATT.unitttype field.
<i>Bad Data</i> =	The return value of the DB read for rITRCUPATH tuple.
<i>Good Data</i> =	The value of the local variable litype_map.
<i>Logical Index</i> =	The index into the PHDATA array.
<i>Dump 1</i> =	The PHDATA array entry.
<i>Dump 2</i> =	The rIMODATT tuple.

Table 5 Error Code: COUNTS_BAD

Error Code:	COUNTS_BAD
Error Description:	Either a new error in dynamic data was introduced after the audit began execution or there is a static ODD inconsistency between the MODATT relation and the NLI_LNK relation. In particular, PHDATA.rhgmap has been verified against relation NLI_LNK and PHDATA.rhgmax has been verified against rIMODATT.eq_tmslnks but, the value of PHDATA.rhgmax != the number of bits set in PHDATA.rhgmap as it should be.
Possible Error Effect:	Call processing for this SM may not hunt all available relative hunt groups.
Corrective Action Taken:	None.
Manual Action Required:	Site personnel should verify the rIMODATT, rINLI_LNK and rISLNTMS tuples for the affected SM against their respective population rules.
Dump Description:	
<i>Error Address</i> =	PHDATA.rhgmap.

<i>Bad Data</i> =	PHDATA.rhgmax.
<i>Good Data</i> =	The number of bits set in PHDATA.rhgmap.
<i>Logical Index</i> =	SM number of the affected SM (index into the PHDATA array).
<i>Dump 1</i> =	The PHDATA array entry for this SM.
<i>Dump 2</i> =	The rIMODATT tuple for this SM.

Table 6 Error Code: NLI_LNK

Error Code:	NLI_LNK
Error Description:	When one of the following is happen for SM-2000: - The audit found that the oddlinks[] or evenlinks[] array entry of the current PHDATA[] array entry didn't match the appropriate tmslnk_0 or tmslnk_1 field of the rNLI_LNK. - The rhgmap or litype_map bit value didn't set correctly based on nli value.
Possible Error Effect:	The call associated with this entry may be trying to use the wrong link.
Corrective Action Taken:	The audit reports the error and writes the correct value of the fields based on the rNLI_LNK tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of the rIMODATT.unittyp field.
<i>Bad Data</i> =	The NLI whose associated data is being checked. Otherwise, the value of the rPHDATA.rhgmap.
<i>Good Data</i> =	The value of the rNLI_LNK.tmslnk_1. If this value is PCNULLTMS then NLI is an ENLI or an NLI1.
<i>Logical Index</i> =	The index into the PHDATA array.
<i>Dump 1</i> =	The PHDATA array entry.
<i>Dump 2</i> =	The rNLI_LNK tuple associated with nli.
<i>Dump 3</i> =	The rIMODATT tuple associated with sm_num.

Table 7 Error Code: NUMELEM

Error Code:	NUMELEM
Error Description:	The audit found that one or more of the hostnum , rhgmax , clid , tmsevenlnk , tmsoddlnk , eq_lis , and orm field of the current PHDATA entry being audited. This means that the current entry has a different value than found in rIMODATT tuple.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This audit will report the error and correct the value of the fields based on the rIMODATT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PHDATA array entry currently being audited.
<i>Bad Data</i> =	The value of the rPHDATA.hostnum field.
<i>Good Data</i> =	The value of hostnum field of the rIMODATT tuple.
<i>Logical Index</i> =	The index into the PHDATA array.
<i>Dump 1</i> =	The PHDATA array entry.
<i>Dump 2</i> =	The rIMODATT tuple.

Table 8 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The audit failed to read a static rIMODATT tuple or a static rNLI_LNK tuple for an equipped (in service) switching module (SM).
Possible Error Effect:	There is an office-dependent data (ODD) problem for this SM.
Corrective Action Taken:	The audit reports the error and schedules the DD audit to recover any DD key errors if there are any, since the audit decides whether the SM is in operation or not by reading the DD data.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA or the nli number used to read the rINLI_LNK tuple.
<i>Bad Data</i> =	The return value of the DB read for tuple.
<i>Good Data</i> =	0 or 1 for MODATT or NLI_LNK respectively.
<i>Logical Index</i> =	The value of loop variable sm_num in the PHDATA array.
<i>Dump</i> =	The rIMODATT tuple used to read the rINLI_LNK tuple, in the NLI_LNK case (<i>Good Data</i> = 1).

Table 9 Error Code: OUTFRANGE

Error Code:	OUTFRANGE
Error Description:	The audit found that rhgsel field of the current PHDATA[] array entry has bad value for this SM. That is, the currently selected RHG is not a valid RHG. It is determined based on rhgmap value. For VCDX office the value of rhgsel is always PC1ST_RHG.
Possible Error Effect:	The call associated with this entry may be trying to use a NCT link that doesn't exist.
Corrective Action Taken:	The audit reports the error and correct value of the rhgsel field to PC1ST_RHG which is always valid.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The beginning address of the PHDATA array entry currently being audited.
<i>Bad Data</i> =	The value of the rIPHDATA.rhgsel field.
<i>Good Data</i> =	The value of the rIPHDATA.rhgmap field.
<i>Logical Index</i> =	The value of loop variable sm_num into the PHDATA array.
<i>Dump 1</i> =	The PHDATA array entry.
<i>Dump 2</i> =	The rIMODATT tuple.

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, an out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to condition.
Corrective Action Taken:	This error will be printed on the receive-only printer (ROP) and the audit will return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	None.

PHDB

Software Release: 5E10 and later

1. PATH DATA BLOCK (PHDB) AUDIT

The path data block relation (RLPHDB) contains information about the two time slots that make up the network path of a call and about the state of the path. The switching module (SM) requests the path, and the administrative module (AM) allocates it. The path data block audit (PHDB) verifies the consistency of the path data block relation. An error in the path data block data can cause lost resources or asserts.

This audit is part of the peripheral control audits subgroup. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

The PHDB audit verifies the data consistency of the path data block relation, which contains information about the network path and the state of the path, the B-channel will be moved to another process but the path is still in the stable state.

2. DATA STRUCTURES

The data structures checked by the PHDB audit are the path data block relation (RLPHDB), the process control block linkage area relation (RLPCBLA), the channel control block common area relation (RLCCBCOM), and the time slot data block relation (RLTSDB). The RLPHDB relation has member linkages with RLPCBLA and RLCCBCOM. It also has both a one-to-one linkage and an owner linkage with RLTSDB. Figure 1 illustrates the layout of a path data block, and Figure 2 illustrates the rIPHDB linkages. Figure 3 illustrates the layout of rIPHDB and rIPCBLA tuples as used for a wide-band call.

LOGICAL KEY (phdbnum)
PATH STATE (pathstt)
. . .
CALL PROCESS SUSPENSION (suspend)
REQUEST FOR ACKNOWLEDGMENT FROM CALL PROCESS (acknowledge)

Figure 1 (PHDB-1) Path Data Block

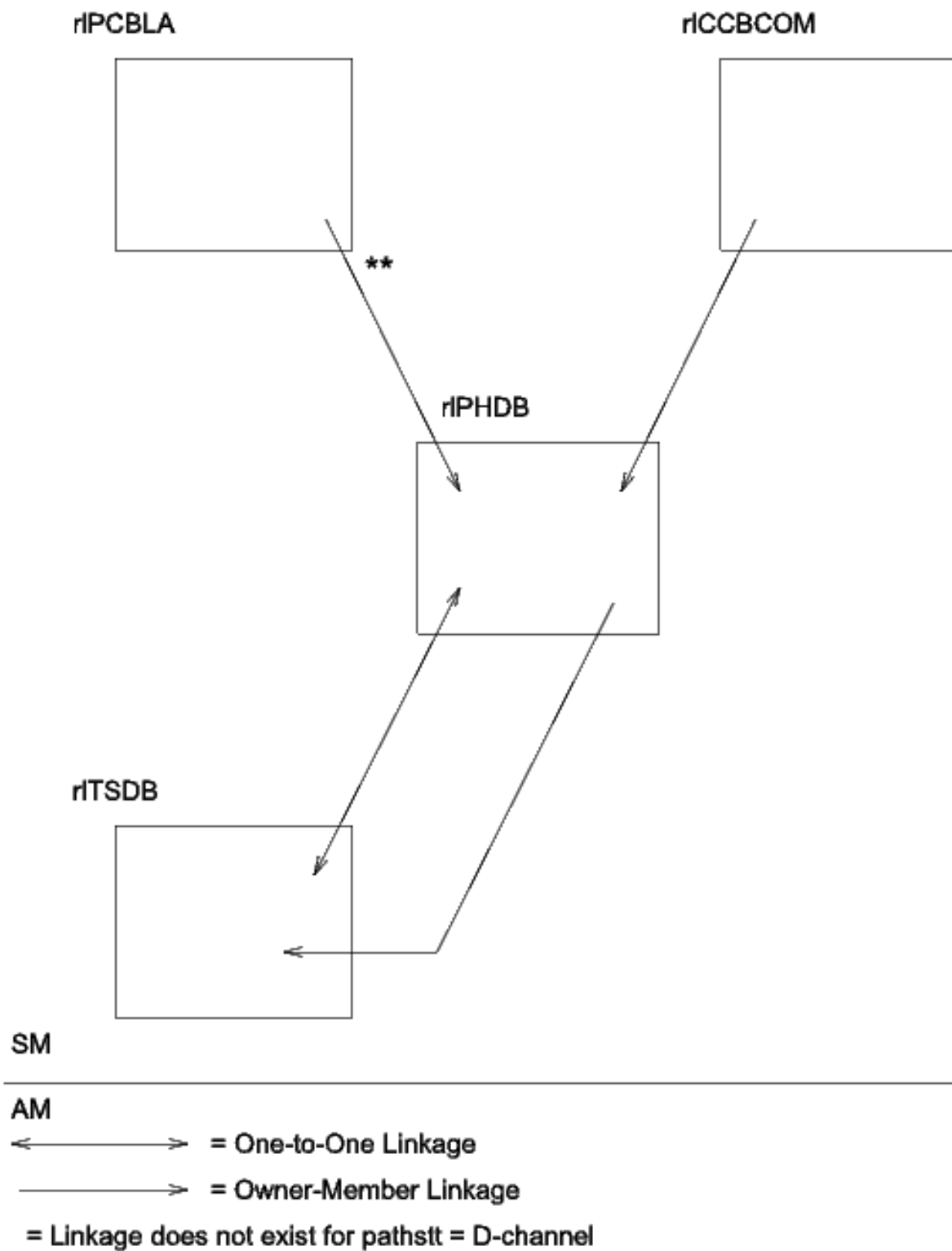


Figure 2 (PHDB-2) rPHDB Linkages

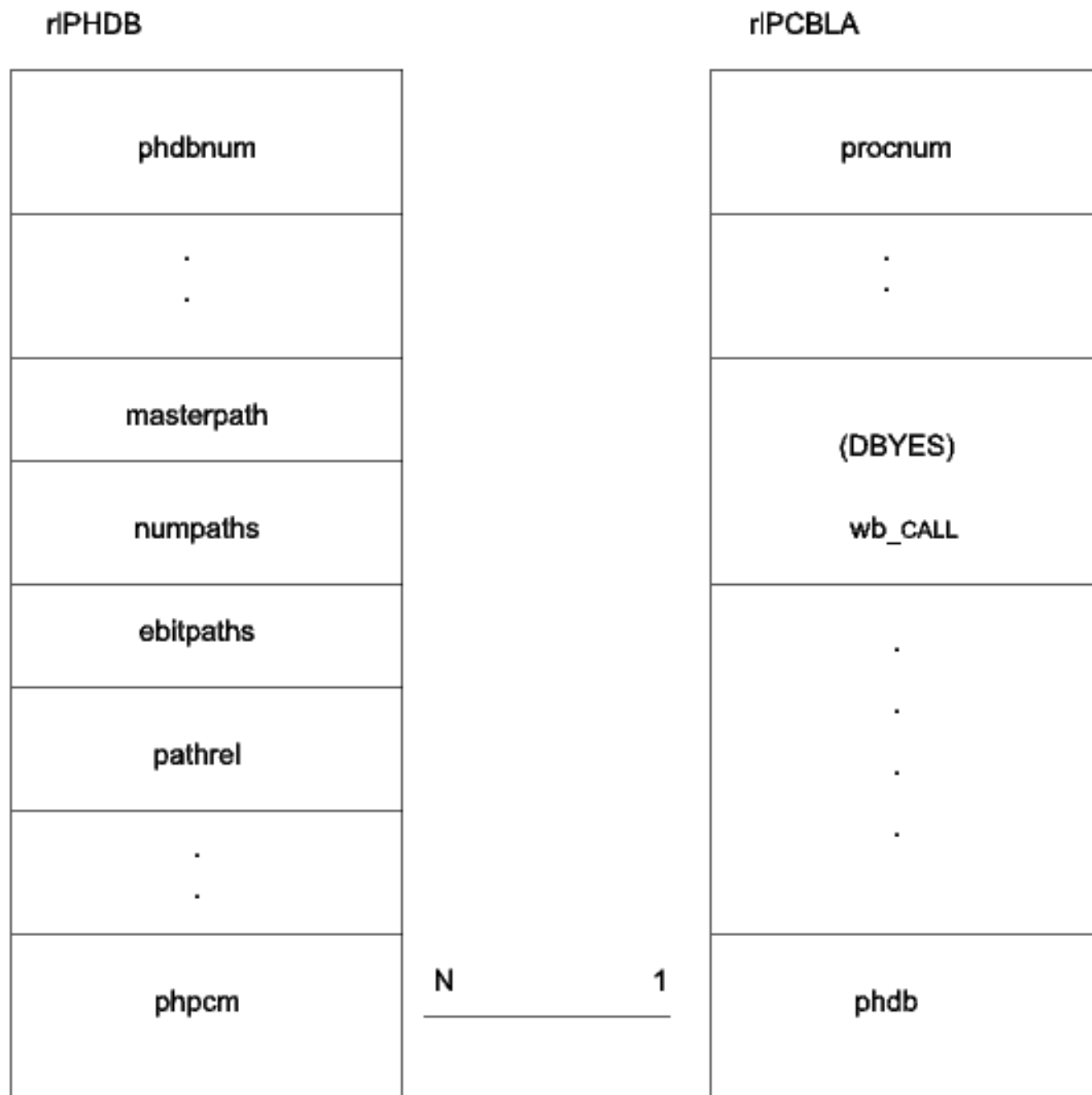


Figure 3 (PHDB-3) Attributes Used by Wide-Band Processing

3. AUDIT CHECKS

To verify the linkages associated with the path data blocks, the PHDB audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the DRLINK driver executes a semantic check function that performs the following checks:

- (a) The audit checks the path state (pathstt) field to ensure that it is a valid state variable.
- (b) The audit checks the path state variable against the path data block linkages. Idle path data blocks should be

owned by the path data block idle list head cell or the path data blocks cannot be used by call processing. An idle path data block should have a zero linkage (except idle list linkage); if it does not, a resource is linked to a path data block on the idle link list and cannot be used by call processing. The linkages to rITSDB, rIPCBLA, and rCCBCOM tuples for a path data block in a stable state should be active; if not, a bad linkage may cause an assert.

When the **pathstt** field indicates a D-channel, the linkage to rIPCBLA will not exist.

If the linkage to the rIPCBLA tuple is active and the type of path (**pathtyp**) is PHWIDEBAND, additional checks are made.

- The **wb_call** attribute in the rIPCBLA tuple must be set.
 - If **masterpath** in the rIPHDB tuple is set, **numpaths** must be between 2 and 24 or the call will fail, and **ebitpaths** must be between 0 and **numpaths**.
 - If **masterpath** in the rIPHDB tuple is not set, the 3 attributes **numpaths**, **ebitpaths**, and **path_rel** must be 0 or the error will be reported and the incorrect attribute will be reset.
- (c) The audit checks the suspend or acknowledge field of a stable path data block against the path state field. These fields should be set to a correct value (DBNO).
- (d) The rIPHDB tuple should not be in a transient state for more than five minutes. The audit times the path state field for a change from the transient state.

If the user places an active call (referred to as process A) on hold, the call appearance will indicate the call is on hold using a fluttering of the lamp. The PC primitive PHopen() is used to open the hardware connection inside TSI and indicate that the path is open in the relation PHDB. The linkages associated with the PCBLA tuple are intact at this moment. Assume the user originates another call (referred to as process B) by pressing another call appearance button. If these two calls want to use the same B-channel, process B will call LPmvport() to get the logical port from process A. If LPmvport() returns SUCCESS, the RTA subsystem will move the physical port (PORTLA) from process A to process B and indicate the state field in the PCBLA tuple of process A as DBPSHOLD. It should be noted here that all of the operations for moving the port around different terminal processes should be done without taking any time break. At this point, the PCBLA tuple associated with process A will not own the B-channel. Also the path (PHDB), even though still in a stable state, is not owned by any B-channel (CCBCOM)[7].

The PHDB audit verifies that:

- If the path is in stable state and the associated PCBLA also indicates the B-Channel has been moved to another process (by checking the state field), then the logical port (CCBCOM) should not own this path (PHDB).
- The path must be marked as OPEN.

4. ERROR HANDLING

Any tuples that fail these checks are reinitialized and placed on the idle list. If any process owns a path data block, that process is purged. Audit control schedules the related audits to run in the elevated mode.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 1 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The path release in the rIPHDB tuple is set (rIPHDB.pathrel = DBYES) even though the rIPHDB tuple is not the master tuple (rIPHDB.masterpath = DBNO).
Possible Error Effect:	This could bring down the entire call.
Corrective Action Taken:	The path release is reset (rIPHDB.pathrel = DBNO) and the call is allowed to continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPHDB tuple found in error.
<i>Bad Data =</i>	The previous path release status (rIPHDB.pathrel).
<i>Good Data =</i>	The bit identifying this rIPHDB tuple as the master tuple for the wide-band call (rIPHDB.masterpath).
<i>Logical Index =</i>	The logical key of the rIPHDB tuple (rIPHDB.phdbnum).
<i>Dump =</i>	The contents of the rIPHDB tuple.

Table 2 Error Code: CALLFLAG_BAD

Error Code:	CALLFLAG_BAD
Error Description:	The path type in the rIPHDB tuple appears to be for a wide-band call (rIPHDB.pathtyp = PHWIDEBAND); however, the associated rIPCBLA tuple is not wide-band (rIPCBLA.wb_call = DBNO).
Possible Error Effect:	Possible failures or asserts when Call Processing attempts to use this rIPHDB tuple.
Corrective Action Taken:	The rIPHDB tuple is recovered to an idle state along with any associated rIPCBLA tuples. Any process associated with the rIPHDB tuple will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPHDB tuple found in error.
<i>Bad Data =</i>	The path type for the rIPHDB tuple (rIPHDB.pathtyp).
<i>Good Data =</i>	The wide-band bit from the rIPCBLA tuple (rIPCBLA.wb_call).
<i>Logical Index =</i>	The logical key of the rIPHDB tuple (rIPHDB.phdbnum).
<i>Dump =</i>	The contents of the rIPHDB tuple.

Table 3 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	The number of paths that have received ebit continuity messages in the rIPHDB tuple is non-zero (rIPHDB.ebitpaths > 0) even though the rIPHDB tuple is not the master tuple (rIPHDB.masterpath = DBNO).
Possible Error Effect:	Possible failures or asserts when call processing attempts to use this rIPHDB tuple.
Corrective Action Taken:	The number of paths having received ebit continuity is reset (rIPHDB.ebitpaths = 0) and the call is allowed to continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPHDB tuple found in error.
<i>Bad Data =</i>	The previous number of paths having received ebit continuity messages (rIPHDB.ebitpaths).
<i>Good Data =</i>	The bit identifying this rIPHDB tuple as the master tuple for the wide-band call (rIPHDB.masterpath).
<i>Logical Index =</i>	The logical key of the rIPHDB tuple (rIPHDB.phdbnum).
<i>Dump =</i>	The contents of the rIPHDB tuple.

Table 4 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	An rIPHDB tuple whose state field indicates that it is idle does not have as its owner the rIPHDB tuple idle list head cell.
Possible Error Effect:	The rIPHDB tuple is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit places the rIPHDB tuple on the idle list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rIPHDB tuple.
<i>Bad Data</i> =	Actual owner's physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	Logical key of rIPHDB tuple.
<i>Dump</i> =	rIPHDB tuple.

Table 5 Error Code: INV

Error Code:	INV
Error Description:	The number of paths that have received ebit continuity messages in the rIPHDB tuple is larger than the number of paths the call is using (rIPHDB.ebitpaths > rIPHDB.numpaths).
Possible Error Effect:	Possible failures or asserts when call processing attempts to use this rIPHDB tuple.
Corrective Action Taken:	The number of paths having received ebit continuity is reset to be no larger than the number of paths (rIPHDB.ebitpaths = rIPHDB.numpaths) and the call is allowed to continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple found in error.
<i>Bad Data</i> =	The previous number of paths having received ebit continuity messages (rIPHDB.ebitpaths).
<i>Good Data</i> =	The number of paths in this rIPHDB tuple (rIPHDB.numpaths).
<i>Logical Index</i> =	The logical key of the rIPHDB tuple (rIPHDB.phdbnum).
<i>Dump</i> =	The contents of the rIPHDB tuple.

Table 6 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The number of paths used for a wide-band call in the rIPHDB tuple is non-zero (rIPHDB.numpaths > 0) even though the rIPHDB tuple is not the master tuple (rIPHDB.masterpath = DBNO).
Possible Error Effect:	Possible failures or asserts when Call Processing attempts to use this rIPHDB tuple.
Corrective Action Taken:	The number of paths is reset (rIPHDB.numpaths = 0) and the call is allowed to continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple found in error.
<i>Bad Data</i> =	The previous number of paths (rIPHDB.numpaths).
<i>Good Data</i> =	The bit identifying this rIPHDB tuple as the master tuple for the wide-band call (rIPHDB.masterpath).
<i>Logical Index</i> =	The logical key of the rIPHDB tuple (rIPHDB.phdbnum).
<i>Dump</i> =	The contents of the rIPHDB tuple.

Table 7 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The number of paths used by a wide-band call in the rIPHDB tuple (rIPHDB.numpaths) is out of range. This field should be between 2 and 24.
Possible Error Effect:	Possible failures or asserts when call processing attempts to use this rIPHDB tuple.
Corrective Action Taken:	The rIPHDB tuple is recovered to an idle state along with any associated rPCBLA tuples. Any process associated with the rIPHDB tuple will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple found in error.
<i>Bad Data</i> =	The number of paths being used for a wide-band call (rIPHDB.numpaths).
<i>Good Data</i> =	The bit identifying this rIPHDB tuple as the master tuple for the wide-band call (rIPHDB.masterpath).
<i>Logical Index</i> =	The logical key of the rIPHDB tuple (rIPHDB.phdbnum).
<i>Dump</i> =	The contents of the rIPHDB tuple.

Table 8 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	A bad linkage has been found in a stable rIPHDB tuple. One of the following errors is present: • The rIPHDB tuple is linked to a head cell (logical key < 0). • The rIPHDB tuple is linked to a dummy tuple (logical key 0). • The linkage is not being used (physical key = 0). All of the links in the rIPHDB tuple should be active in a stable state.
Possible Error Effect:	The bad linkage may cause an ASSERT before the audit finds the error.
Corrective Action Taken:	The audit idles the rIPHDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple.
<i>Bad Data</i> =	Linkage contents.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIPHDB tuple.
<i>Dump</i> =	rIPHDB tuple.

Table 9 Error Code: STATE

Error Code:	STATE
Error Description:	An rIPHDB tuple has its suspend or acknowledge field set to a wrong value. They both should be set to DBNO when the rIPHDB is in a stable state.
Possible Error Effect:	The rIPHDB tuple is in an invalid state.
Corrective Action Taken:	The audit idles the rIPHDB tuple, and zeros the linkages involved. The audit attempts to find the owner of the rIPHDB tuple and purges the corresponding process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple.
<i>Bad Data</i> =	rIPHDB tuple suspend field (bit 1) acknowledge field (bit 0).
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIPHDB tuple.
<i>Dump</i> =	rIPHDB tuple.

Table 10 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	An rIPHDB tuple has an invalid state variable.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit idles the rIPHDB tuple, and zeros the linkages involved. The audit attempts to find the owner of the rIPHDB tuple and purges the corresponding process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple.
<i>Bad Data</i> =	The state variable.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIPHDB tuple.
<i>Dump</i> =	rIPHDB tuple.

Table 11 Error Code: STUCK

Error Code:	STUCK
Error Description:	The tuple has been in the transient state for more than 5 minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rIPHDB tuple is idled. The stuck call is terminated.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIPHDB tuple.
<i>Dump</i> =	rIPHDB tuple.

Table 12 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rIPHDB tuple has a nonzero linkage in one of its link blocks.
Possible Error Effect:	Some resource is linked to an rICCB COM on the idle link list and may be unusable by call processing.
Corrective Action Taken:	The audit initializes the rIPHDB tuple and places it on the idle list. As part of initialization, the audit zeros the linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPHDB tuple.
<i>Bad Data</i> =	Linkage contents.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rIPHDB tuple.
<i>Dump</i> =	rIPHDB tuple.

PHDBBT

Software Release: 5E10 and later

1. PATH DATA BLOCK BACKGROUND TIMER (PHDBBT) AUDIT

The PHDBBT audit checks the background timer head cells (PCbthead) and the background timer blocks (PCbtblk) looking for blocks lost to the system.

The PHDBBT audit runs in segmented mode only. The audit is not allowed to be run in directed mode or as part of a selective initialization. An assert is generated if the audit is entered using a mode other than segmented mode.

2. DATA STRUCTURES

This audit uses three array structures: PCbthead[], PCbtblk[] and AUphbtmap[]. Each PCbthead[] array entry may have multiple PCbtblk[] array entries linked to them. The AUphbtmap[] is used by the audit to store a "touch" flag for comparison against what the PCbthead[] array has linked to it and what the PCbtblk[] array actually has in use.

3. AUDIT CHECKS

The PHDBBT audit does not use any audit drivers and uses the following error codes. For details about each error see the comment blocks given.

If the audit is just starting:

ZAP = The audit will take this error as a result of a write protect, out of range, or invalid operation by any audit system process.

The audit map (AUphbtmap[]) is zeroed. The appropriate audit map bit is set (touched) for each valid PCbtblk[] array entry linked to PCbthead[] entries:

IDLE_LINK = Check that the background timer block index is in range or the block has already been tagged.

Step through each PCbtblk[] array entry and check:

BIT1MAP = Check that the corresponding AUphbtmap[] touch bit is set.

4. ERROR HANDLING

When a lost PCbtblk[] entry is found it is inserted as first in the idle list and the audit continues.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	A peripheral control (PC) path data block background timer block (PCbtblk[]) is lost to the system.
Possible Error Effect:	This can block calls from being originated on the switching module (SM).
Corrective Action Taken:	The PC path data block background timer block (PCbtblk[]) is placed on the idle list (PCbthead[]).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Audit touch flag map (AUphbtmap[]) entry.
<i>Bad Data</i> =	Bit position in AUphbtmap[] found in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into PCbtblk[].
<i>Dump</i> =	PCbtblk[] array entry.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A PC path data block background timer head cell (PCbthead[]) points to an out-of-range path data block background timer block (PCbtblk[]).
Possible Error Effect:	This may cause the network control system process (NCSP) to assert when requesting or canceling path data block background timing.
Corrective Action Taken:	The PC path data block background timer head cell (PCbthead[]) is tied off at the point of corruption. The PC path data block background timer blocks left in limbo are then placed onto the reconstructed head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Audit touch flag map (AUphbtmap[]).
<i>Bad Data</i> =	The current index into PCbtblk[].
<i>Good Data</i> =	Bit position in AUphbtmap[].
<i>Logical Index</i> =	The current index into PCbthead[].
<i>Dump 1</i> =	Current PCbtblk[] array entry.
<i>Dump 2</i> =	Previous PCbtblk[] array entry in the linked list.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Reinitialize all data that could have made the audit cause a system reset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into PCbthead[].
<i>Dump</i> =	NA.

PHDSLGP

Software Release: 5E10 and later

1. PROTOCOL HANDLER DIGITAL SUBSCRIBER LINE GROUP (PHDSLGP) AUDIT

The protocol handler (PH) is used with the packet switching unit (PSU) to ensure that packets are transmitted and received in the proper formats. The digital subscriber lines (DSLs) are the lines over which the packets flow. The switch contains one DSL group for each PH. Each group contains 16 members. The PHDSLGP audit verifies the mapping of DSL groups to PHs.

The PHDSLGP audit is part of the integrated services digital network (ISDN) base audits subgroup; it is part of the ISDNBASE loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

The data structures used by this audit are the DSL global data relation (RLDSLGDATA), the circuit data relation (RLCKTDATA) and the switch maintenance equipment status relation (RLSMEST), the protocol handler DSL group array (SMph_dslgp), the DSL group to protocol handler array (SMdslgp_ph[]), the PH hardware/image type array SMph_type[], and the DT keys DTLOGIPP and DTPHIMAGE. Refer to Figure 1 .

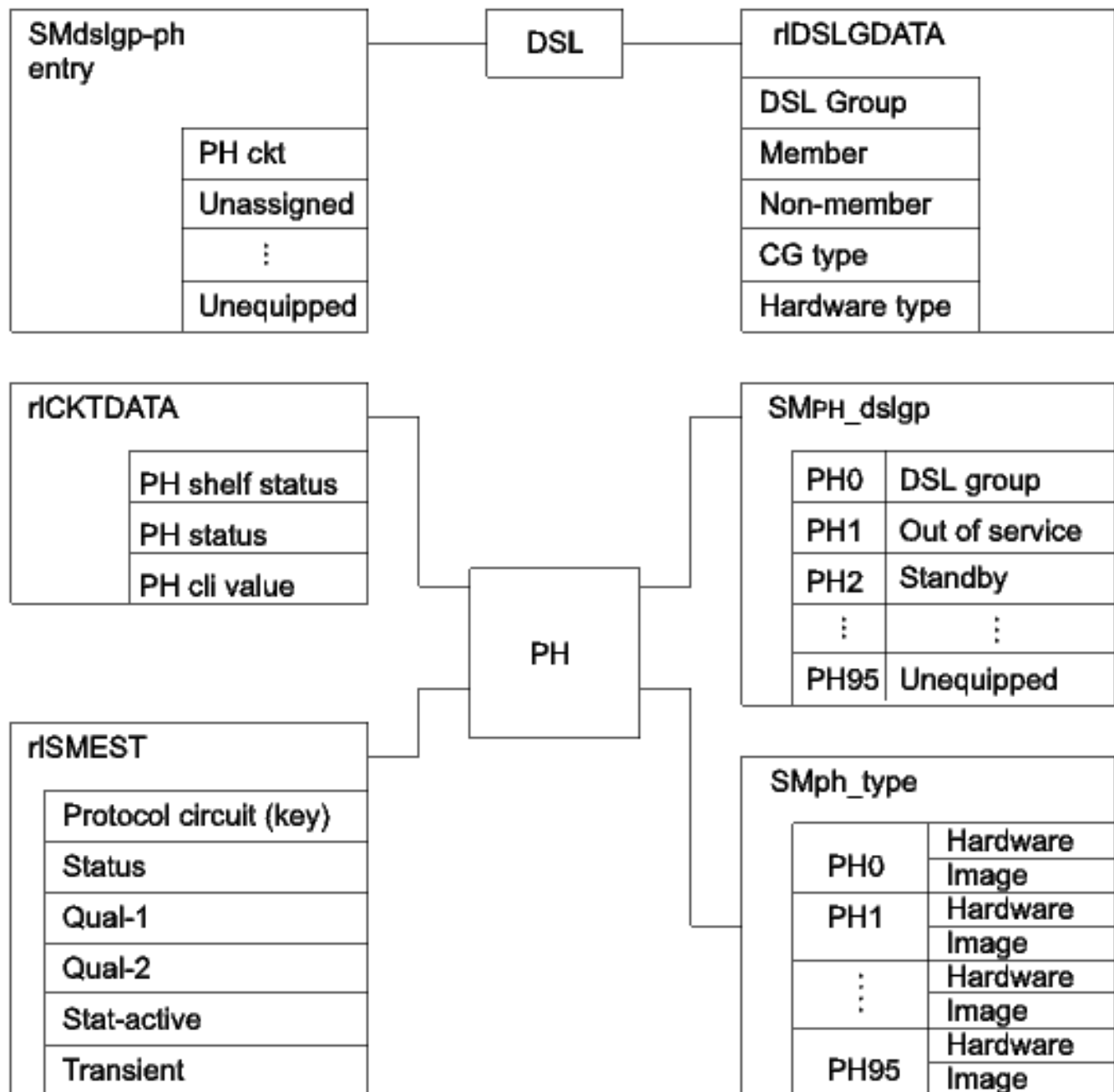


Figure 1 (PHDSLGP-1) Data Structures of the PHDSLGP Audit

3. AUDIT CHECKS

For each PH the audit performs semantic checks based on its entry in `SMph_dslgp[ ]`:

- The circuit data relation (`rCKTDATA`) is read for the PH shelf (16 PHs on the shelf). If the shelf is unequipped, all PHs on the shelf (entries in `SMph_dslgp[ ]`) must be unequipped.
- If the PH is unequipped, no further checks are performed.
- The PH's hardware type in `SMph_type[ ]` is checked against `rCKTDATA` for the PH.

- (d) The PH image type in SMph_type[] is checked against data transport key DTPHIMAGE.
- (e) If the PH is equipped, its associated rSMEST tuple is read:
 - If rSMEST indicates that it is out-of-service (OOS), the corresponding entry in SMph_dslgp[] and data transport data should be OOS.
 - If rSMEST indicates standby (STBY) status, the corresponding entry in SMph_dslgp[] should be STBY. Also, the PH image in SMph_type[] should be null.
 - If rSMEST indicates active (ACT), a point-to/point-back check is performed against SMdslgp_ph[].
 - Data transport key DTLOGIPP is checked against SMph_dslgp[].

For each entry in SMdslgp_ph[], the audit reads the static relation rDSLGDATA to determine if the DSL group has ports (members) assigned. Semantic checks are performed based on the entry in SMdslgp_ph[]:

- (a) If the DSL group is unequipped according to its entry in SMdslgp_ph[], no ports should be assigned to the DSL group.
- (b) If the DSL group is unassigned according to its entry in SMdslgp_ph[], entries in SMph_dslgp[] and SMph_type[] are checked to ensure there are no spare PHs of the same hardware type available.
- (c) If the DSL group is assigned to a PH, a point-to/point-back check is performed with the corresponding entry in SMph_dslgp[]. Also, if the point-to/point-back check passes, the PH hardware and image types in SMph_type[] are checked against data from rPSLGDATA.

4. ERROR HANDLING

If an error is detected in SMdslgp_ph[], the following recovery is performed:

- (a) If the DSL group has ports assigned to it and the audit is not running during a selective initialization, an Integrated Services Port Administrator (ISPA) job is queued which will unassign ports from this DSL.
- (b) The array entry is recovered to unassigned.

If an error is detected in SMph_dslgp[], the following recovery is performed:

- (a) If the PH was assigned to a DSL group, the corresponding entry in SMdslgp_ph[] is written to UNASSIGNED.
- (b) The array entry in SMph_dslgp[] is recovered to OOS.
- (c) The data transport bit "DTLOGIPP" is updated to OOS.
- (d) The PH image in SMph_type[] and data transport key DTPHIMAGE are nulled out.
- (e) The rSMEST tuple for the PH is updated to OOS/AUOOS.
- (f) The PH circuit is placed on the Maintenance Request Administrator (MRA) recovery queue.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADPHIMG

Error Code:	BADPHIMG
Error Description:	A PH does not have the image required by the DSL group to which it is assigned.
Possible Error Effect:	Subscriber service may not be administered through this DSL group.
Corrective Action Taken:	The PH image value is rewritten, the DTPHIMAGE value for the PH is checked and, if necessary, rewritten which will force the PH to be repumped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PH entry in SMph_dslgp[].
<i>Bad Data</i> =	The PH image type.
<i>Good Data</i> =	The image the DSL group requires.
<i>Logical Index</i> =	Physical PH number.
<i>Dump</i> =	NA.

Table 2 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	A protocol handler's rSMEST tuple has an unexpected basic status.
Possible Error Effect:	The PH is in an unknown state.
Corrective Action Taken:	The PH is taken out-of-service and placed on the MRA circuit recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Contents of basic status field of rSMEST.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	PH circuit name.
<i>Dump</i> =	PH rSMEST tuple.

Table 3 Error Code: CKTDATA

Error Code:	CKTDATA
Error Description:	An entry in SMdslgp_ph[] contains an invalid protocol handler (PH) circuit name.
Possible Error Effect:	The PH is in an unknown state.
Corrective Action Taken:	None.
Manual Action Required:	A tuple in rCKTDATA needs to be added for the circuit name in the SMdslgp_ph[] since the PH circuit name contained in the SMdslgp_ph[] has no corresponding entry in rCKTDATA, or change rDSLGDATA.assigned to 0 (DBNO).
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The contents of array entry in SMdslgp_ph[].
<i>Good Data</i> =	The address of array entry in SMdslgp_ph[].
<i>Logical Index</i> =	The key of the rCKTDATA tuple (rCKTDATA.name).
<i>Dump</i> =	NA.

Table 4 Error Code: DSLNOMEMBERS

Error Code:	DSLNOMEMBERS
Error Description:	A DSL group has no ports assigned to it, but is not marked unequipped in SMdslgp_ph[].
Possible Error Effect:	This DSL may be used by call processing when it should not be.
Corrective Action Taken:	The DSL group is marked unequipped in SMdslgp_ph[] array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of array entry in SMdslgp_ph[].
<i>Bad Data</i> =	Contents of array entry in SMdslgp_ph[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DSL group.
<i>Dump</i> =	Associated rDSLGDATA tuple.

Table 5 Error Code: DSLONE2ONE

Error Code:	DSLONE2ONE
Error Description:	A DSL group points to a protocol handler, but the associated protocol handler does not point back.
Possible Error Effect:	Subscriber service can not be given for this DSL.
Corrective Action Taken:	All ports on this DSL group are unassigned.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of array entry in SMdslgp_ph[].
<i>Bad Data</i> =	Contents of array entry in SMph_dslgp[].
<i>Good Data</i> =	DSL Group number.
<i>Logical Index</i> =	DSL Group number.
<i>Dump</i> =	rSMEST for PH.

Table 6 Error Code: DTLOGIPP

Error Code:	DTLOGIPP
Error Description:	Data transport data for a PH was found to be inconsistent with the global array SMph_dslgp[].
Possible Error Effect:	Data transport does not have valid status for a PH.
Corrective Action Taken:	Data transport bit in error is recovered to be consistent with data in SMph_dslgp[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PH entry in SMph_dslgp[].
<i>Bad Data</i> =	DT data for this PH.
<i>Good Data</i> =	Contents of array entry in SMph_dslgp[].
<i>Logical Index</i> =	PH number.
<i>Dump</i> =	rSMEST for PH.

Table 7 Error Code: DTPHIMG

Error Code:	DTPHIMG
Error Description:	The DT key DTPHIMAGE and SMph_type[] do not have the same image listed for a PH.
Possible Error Effect:	Data Transport has a different PH image listed than SMph_type[].
Corrective Action Taken:	All incorrect values are corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The PH image type from DTPHIMAGE.
<i>Good Data</i> =	The PH image from SMph_type[].
<i>Logical Index</i> =	Physical PH number.
<i>Dump</i> =	NA.

Table 8 Error Code: GRBG

Error Code:	GRBG
Error Description:	A DSL group is assigned to a PH that exists on a different shelf, or there is out-of-range data in the SMdslgp_ph[] global array.
Possible Error Effect:	Call processing may be affected for this DSL group.
Corrective Action Taken:	All ports assigned to this DSL group are unassigned.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMdslgp_ph[] entry.
<i>Bad Data</i> =	Contents of SMdslgp_ph[] entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	DSL group number.
<i>Dump</i> =	NA.

Table 9 Error Code: HWCMPAT

Error Code:	HWCMPAT
Error Description:	A PH and DSL group are assigned to each other but have different hardware

	types.
Possible Error Effect:	Service cannot be provided using this assignment.
Corrective Action Taken:	The DSL group is unassigned and the PH is written to audit-out-of-service and put on the MRA recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMph_dslgp[] entry.
<i>Bad Data</i> =	PH hardware type.
<i>Good Data</i> =	DSL group hardware type.
<i>Logical Index</i> =	Physical PH number.
<i>Dump</i> =	NA.

Table 10 Error Code: LOST_DSLGDATA

Error Code:	LOST_DSLGDATA
Error Description:	A tuple in the static relation RLDSLGDATA can not be read for the DSL group.
Possible Error Effect:	The PHDSLGP audit can not validate PH to DSL group mappings.
Corrective Action Taken:	None.
Manual Action Required:	The static relation RLDSLGDATA must be populated for each possible DSL group in this SM.
Dump Description:	
<i>Error Address</i> =	Address of rIDLSDGDATA buffer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	DSL group number.
<i>Logical Index</i> =	DSL group number.
<i>Dump</i> =	NA.

Table 11 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	A protocol handler points to a DSL group but the DSL group does not point back.
Possible Error Effect:	The PH cannot serve the DSL group.
Corrective Action Taken:	The PH is taken out-of-service, and placed on the MRA circuit recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in SMdslgp_ph[].
<i>Bad Data</i> =	Contents of entry in SMdslgp_ph[].
<i>Good Data</i> =	PH circuit name.
<i>Logical Index</i> =	DSL group number.
<i>Dump</i> =	PH rSMEST tuple.

Table 12 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	A protocol handler is assigned to a DSL group on another shelf, or out-of-range data has been detected in the global array SMph_dslgp[].
Possible Error Effect:	MRA can not access the PH's DSL group correctly.
Corrective Action Taken:	The PH is taken out-of-service and placed on the MRA circuit recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMph_dslgp[] entry.
<i>Bad Data</i> =	Contents of SMph_dslgp[] entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	PH number.
<i>Dump</i> =	PH rSMEST tuple.

Table 13 Error Code: PHHWTYPE

Error Code:	PHHWTYPE
Error Description:	A PH does not have the correct hardware type listed in SMph_type[].
Possible Error Effect:	None.
Corrective Action Taken:	The hardware type value is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.

<i>Bad Data</i> =	The PH hardware type from SMph_type[].
<i>Good Data</i> =	The PH hardware type from the rICKTDATA.cli value.
<i>Logical Index</i> =	Physical PH number.
<i>Dump</i> =	NA.

Table 14 Error Code: PHNOOS

Error Code:	PHNOOS
Error Description:	A protocol handler's associated rSMEST tuple is in the out-of-service state, but the corresponding entry in SMph_dslgp[] is not out-of-service.
Possible Error Effect:	MRA can not access the PH correctly.
Corrective Action Taken:	The PH circuit is placed on the MRA circuit recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of entry in SMph_dslgp[].
<i>Bad Data</i> =	Contents of SMph_dslgp[] entry.
<i>Good Data</i> =	Correct contents of SMph_dslgp entry (SMPH_OOS).
<i>Logical Index</i> =	PH number.
<i>Dump</i> =	PH rSMEST tuple.

Table 15 Error Code: PHNSPARE

Error Code:	PHNSPARE
Error Description:	A protocol handler's associated rSMEST tuple is in the standby state, but the array entry in SMph_dslgp[] does not indicate spare status.
Possible Error Effect:	MRA can not access the PH correctly.
Corrective Action Taken:	The PH is taken out-of-service and placed on the MRA circuit recovery queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Contents of SMph_dslgp[] entry for the PH.
<i>Good Data</i> =	Correct contents of SMph_dslgp[] entry (SMPH_SPARE).
<i>Logical Index</i> =	PH number.
<i>Dump</i> =	PH rSMEST tuple.

Table 16 Error Code: PHNULLIMG

Error Code:	PHNULLIMG
Error Description:	The image value for an out of service PH is not null.
Possible Error Effect:	None.
Corrective Action Taken:	The PH image value is rewritten.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The PH image type from SMph_type[].
<i>Good Data</i> =	SMIMAGENULL.
<i>Logical Index</i> =	Physical PH number.
<i>Dump</i> =	NA.

Table 17 Error Code: PHNUNEQUIP

Error Code:	PHNUNEQUIP
Error Description:	A protocol handler is unequipped according to rICKTDATA (shelf) but is not according to SMph_dslgp[].
Possible Error Effect:	DSL groups may be assigned to this PH, but should not be.
Corrective Action Taken:	The PH entry in the global array SMph_dslgp is updated to unequipped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PH entry in SMph_dslgp[].
<i>Bad Data</i> =	Contents of PH entry in SMph_dslgp[].
<i>Good Data</i> =	The correct contents of SMph_dslgp[] (SMPH_UNEQUIPPED).
<i>Logical Index</i> =	PH number.
<i>Dump</i> =	NA.

Table 18 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	An rSMEST tuple was not found for either a PH or the PSUCOM.
Possible Error Effect:	The circuit's status is unknown.
Corrective Action Taken:	The CKTDATA audit will find and correct this.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Circuit name.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	NA.

Table 19 Error Code: UNASSIGNED

Error Code:	UNASSIGNED
Error Description:	A standby PH was found when an unassigned DSL group of the same hardware type exists on the same shelf.
Possible Error Effect:	Subscriber service is degraded.
Corrective Action Taken:	The PH is written to audit-out-of-service and placed on the MRA recovery queue for assignment to the DSL group.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rDSLGDATA buffer.
<i>Bad Data</i> =	Contents of SMdslgp_ph[] array.
<i>Good Data</i> =	PH circuit name.
<i>Logical Index</i> =	DSL group number.
<i>Dump</i> =	NA.

Table 20 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	A PH should be unequipped according to rCKTDATA but is not.
Possible Error Effect:	MRA cannot access this PH correctly.
Corrective Action Taken:	The global array SMph_dslgp[] is recovered to unequipped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rCKTDATA buffer.
<i>Bad Data</i> =	Contents of array entry in SMph_dslgp[].
<i>Good Data</i> =	SMPH_UNEQUIPPED.
<i>Logical Index</i> =	PH number.
<i>Dump</i> =	NA.

Table 21 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	Return to audit control.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

PHISTORY

Software Release: 5E10 and later

1. PENDING HISTORY (PHISTORY) AUDIT

Every port has a current status that represents its maintenance condition. Status values include in-service, in-service hunted, out-of-service family of equipment, and out-of-service power cross. The current status of a given port is stored in the port linkage area relation (RLPORTLA).

Ports with out-of-service maintenance conditions may have more than one status because they may be out of service for more than one reason. A port's highest priority status is stored in an rIORTLA tuple as its current status. Any lower priority statuses (pending statuses) are stored in tuples of the pending history relation (RLPHISTORY). The system keeps up to three pending statuses for a given out-of-service port in a single rIPHISTORY tuple, which is linked to the rIORTLA tuple. Each rIPHISTORY tuple is a push-down stack in which the higher priority status is on top.

If an out-of-service status is applied to a port that is already out-of-service, the latest status replaces the current status if it is a higher priority. The previous status then becomes a pending status in an rIPHISTORY tuple. If an out-of-service status is cleared for a given port, the highest priority pending status becomes the new current status. If no pending status exists, the status becomes in-service or in-service hunted depending on the port hunt type.

The pending history audit (PHISTORY) verifies that rIPHISTORY tuples are properly linked. Idle rIPHISTORY tuples must be linked to an idle head cell in the RLDB_HEADC relation. Idle rIPHISTORY tuples must not be linked to rIORTLA tuples. Busy rIPHISTORY tuples must be linked to a single rIORTLA tuple and also belong to a list of busy rIPHISTORY tuples. The busy rIPHISTORY tuple list is identified by another head cell tuple in the rIORTLA relation. The audit also verifies that the number of pending statuses for a given rIPHISTORY tuple is within range.

This audit is part of the terminal maintenance audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The pending history audit operates on a member relation of the database head cells relation (RLDB_HEADC). The RLDB_HEADC relation has head cells for busy and available pending history (RLPHISTORY) tuples. Figure 1 shows how the tuples of the member relation, RLPHISTORY, are linked to the RLDB_HEADC relation and the port linkage area relation (RLPORTLA). Each tuple of RLPHISTORY has a count field and up to three statuses.

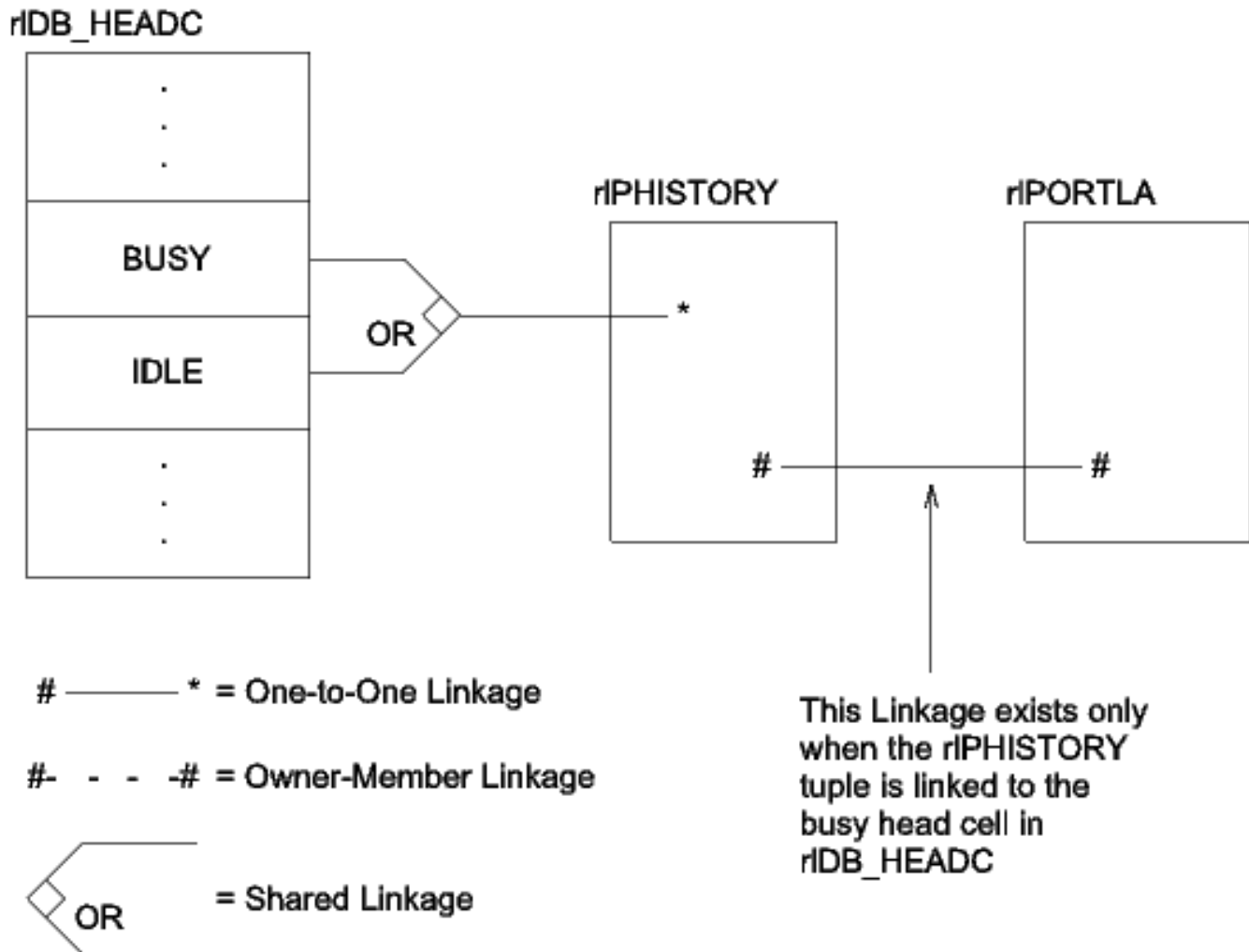


Figure 1 (PHISTORY-1) rIPHISTORY Linkages

3. AUDIT CHECKS

To verify the linkages between rIPHISTORY and rIDB_HEADC, and the linkages between rIPHISTORY and rIORTLA, the PHISTORY audit calls the DRINASEK audit driver.

After the driver verifies the owner-member linkages, the audit performs the following semantic checks:

- The audit ensures that the level number (that is, the number of pending statuses) in rIPHISTORY (on the busy list) is valid by checking the count field. Valid values are the integers one through three, where three is the maximum number for which space has been allocated.
- The audit verifies that an rIPHISTORY tuple on the busy list is linked to an rIORTLA tuple.
- The audit verifies that an rIPHISTORY tuple on the idle list is not linked to an rIORTLA tuple.
- The audit verifies that every rIPHISTORY tuple is linked to either the busy or idle list head cells in rIDB_HEADC.

4. ERROR HANDLING

Any tuples that fail these checks are zeroed and placed on the idle list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	A tuple linked to idle list was linked to an rPORTLA tuple.
Possible Error Effect:	The rPHISTORY tuple may incorrectly be reallocated to another port. Pending history of linked rPORTLA may be lost.
Corrective Action Taken:	The linked port was idled, any associated process was purged, and the rPHISTORY tuple was idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Link field to rPORTLA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rPHISTORY tuple.
<i>Dump</i> =	rPHISTORY tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A tuple linked to busy list was not linked to an rPORTLA tuple.
Possible Error Effect:	Reduced number of available rPHISTORY tuples.
Corrective Action Taken:	The tuple was returned to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of tuple.
<i>Bad Data</i> =	Link field to rPORTLA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rPHISTORY tuple.
<i>Dump</i> =	rPHISTORY tuple.

Table 3 Error Code: LVLNUM

Error Code:	LVLNUM
Error Description:	The number of pending statuses is zero or greater than the number for which space is allocated.
Possible Error Effect:	All currently pending statuses will be lost.
Corrective Action Taken:	The rPHISTORY tuple is placed on the available list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	lvnum field in rPHISTORY tuple.
<i>Bad Data</i> =	lvnum field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rPHISTORY key.
<i>Dump</i> =	rPHISTORY tuple.

Table 4 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rPHISTORY tuple is validly linked to an illegal rIDB_HEADC tuple.
Possible Error Effect:	May not be able to create a pending status entry for a port since rPHISTORY tuple(s) are lost.
Corrective Action Taken:	The rPHISTORY tuple is placed on the available list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPHISTORY tuple.
<i>Bad Data</i> =	Owner link field.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	riPHISTORY key.
<i>Dump</i> =	riPHISTORY.

PHJBUF

Software Release: 5E11 and later

1. PROTOCOL HANDLER DATA ADMINISTRATOR JOB BUFFER (PHJBUF) AUDIT

PHJUBUF audit steps through the RLJBUF (PHDA job buff) relation. For each tuple, the audit checks the linkage between rJBUF tuple and its owner RLDB_HEADC (database head cell) relation.

2. DATA STRUCTURES

For details on the fields contained in the RLJBUF relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

PICB-A

Software Release: 5E10 - 5E11

1. PERIPHERAL INTERFACE CONTROL BUS (PICB) AUDIT

Peripheral interface control buses pass control information from the peripheral units to the module controller/time-slot interchanger (MC/TSI). The information passed over the bus includes control and timing configurations. The PICB audit ensures that the global peripheral interface control bus array (Gpicb) is identical to the static peripheral interface control bus relation (RLPICB).

The PICB audit is part of the switch maintenance audits subgroup. It runs in the segmented, nonsegmented, and initialization modes.

2. DATA STRUCTURES

As Figure 1 shows, the PICB audit checks the information contained in the dynamic peripheral interface control bus global array (Gpicb) with the information in the static peripheral interface control bus relation (RLPICB). The fields checked include the peripheral interface control bus (PICB) number, the peripheral interface data bus (PIDB) number, the unit type, the unit number, and the board type. It also verifies the global dynamic array holding the mask for circuits in line unit service groups. This mask contains the PICB number.

3. AUDIT CHECKS

The PICB audit invokes the static driver (DRSTATIC) to read the tuples of rLPICB in the ODD. The audit compares each tuple in rLPICB for an identical one in Gpicb. Differences between the two data structures are errors. The PICB number from rLPICB is checked with the PClusgname[] array entries corresponding to the PICB's servicing line units.

4. ERROR HANDLING

If the audit finds a tuple in RLPICB that does not exist in Gpicb, it creates an entry in Gpicb. If the information in a Gpicb tuple differs from the information in the corresponding tuple of rLPICB, the audit considers the information in Gpicb in error and overwrites it with the contents of the tuple of RLPICB. If the dynamic global array PClusgname is incorrect, it will be corrected.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The static and dynamic copies of rLPICB are not identical. The dynamic copy in Gpicb[] global array is wrong.
Possible Error Effect:	May not be able to properly set up calls. Depending on the type of PICB that is wrong, it might also cause hardware problem.
Corrective Action Taken:	The static version of rLPICB is copied into the dynamic version.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Beginning address of the dynamic tuple that is found to be in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	picb number.
<i>Dump 1 =</i>	The dynamic rLPICB tuple under audit.
<i>Dump 2 =</i>	The static rLPICB tuple that is considered correct.

Table 2 Error Code: SGROUP

Error Code:	SGROUP
Error Description:	The PClusgname[] array entry associated with a particular line unit does not contain the correct mask (0xPP00, where PP = PICB + 1).
Possible Error Effect:	If the PClusgname[] entry is wrong, circuit names for this line unit will be miscalculated. This would cause serious problems for growth, maintenance, and operational software. Other audits such as HLSC, LINE, and CHAN depend on the correctness of PClusgname[].
Corrective Action Taken:	The PClusgname[] entry is corrected using the PICB number in static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PClusgname[] entry in error.
<i>Bad Data</i> =	The PICB + 1 value in PClusgname[].
<i>Good Data</i> =	The PICB value in rPICB (plus 1).
<i>Logical Index</i> =	The PICB number.
<i>Dump</i> =	The static rPICB tuple that is considered correct.

PICB-B

Software Release: 5E12 and later

1. PERIPHERAL INTERFACE CONTROL BUS (PICB) AUDIT

Peripheral interface control buses pass control information from the peripheral units to the module controller/time-slot interchanger (MC/TSI). The information passed over the bus includes control and timing configurations. The PICB audit ensures that the global peripheral interface control bus array (Glpicb) is identical to the static peripheral interface control bus relation (RLPICB).

The PICB audit is part of the switch maintenance audits subgroup. It runs in the segmented, nonsegmented, and initialization modes.

2. DATA STRUCTURES

As Figure 1 shows, the PICB audit checks the information contained in the dynamic peripheral interface control bus global array (Glpicb) with the information in the static peripheral interface control bus relation (RLPICB). The fields checked include the peripheral interface control bus number (PICB #), the peripheral interface data bus number (PIDB #), the unit type, the unit number, and the board type. It also verifies the global dynamic array holding the mask for circuits in line unit service groups. This mask contains the PICB number.

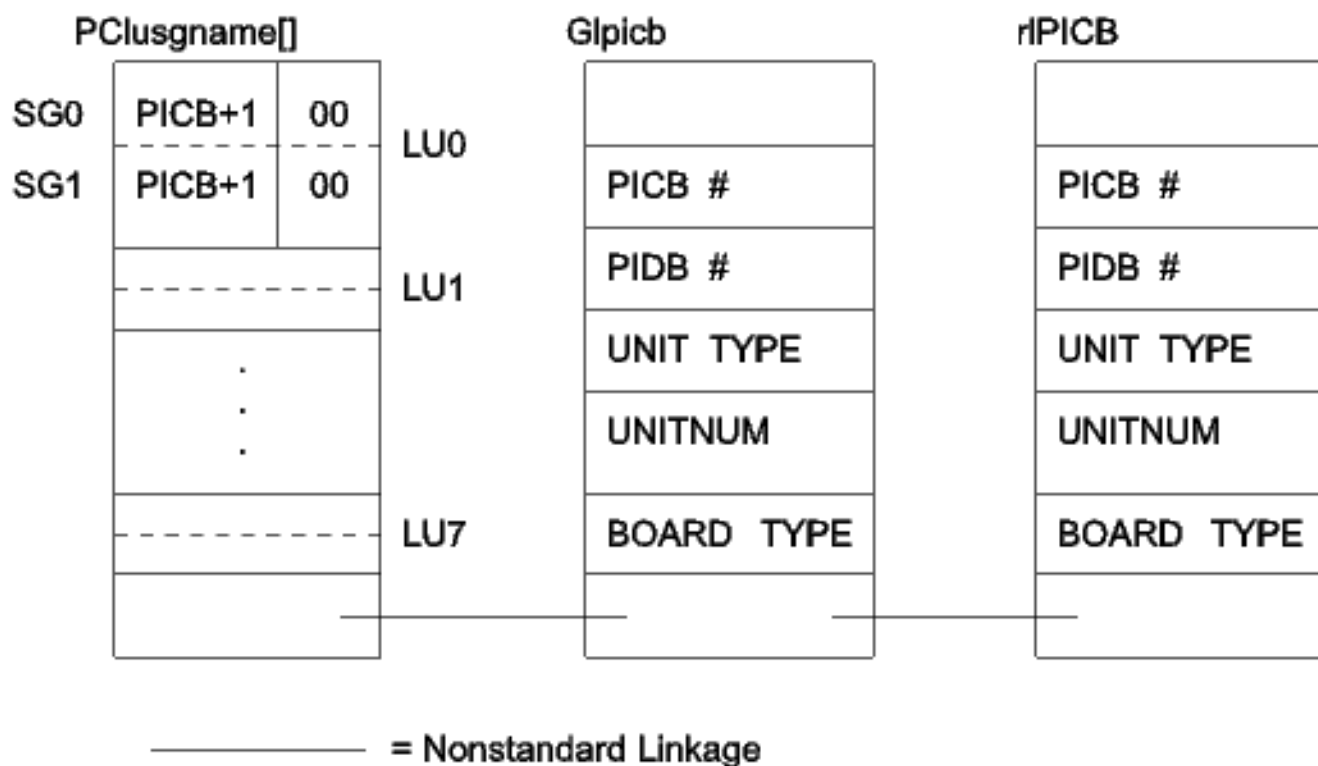


Figure 1 (PICB-1) Data Structures of the PICB Audit

3. AUDIT CHECKS

The PICB audit invokes the static driver (DRSTATIC) to read the tuples of rLPICB in the ODD. The audit compares each tuple in RLPICB for an identical one in Glpicb. Differences between the two data structures are errors. The PICB number from rLPICB is checked with the PClusname[] array entries corresponding to the PICB's servicing line units.

4. ERROR HANDLING

If the audit finds a tuple in rPICB that does not exist in Gpicb, it creates an entry in Gpicb. If the information in a Gpicb tuple differs from the information in the corresponding tuple of rPICB, the audit considers the information in Gpicb in error and overwrites it with the contents of the tuple of rPICB. If the dynamic global array PClusgname is incorrect, it will be corrected.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The static and dynamic copies of rPICB are not identical. The dynamic copy in Gpicb[] global array is wrong.
Possible Error Effect:	May not be able to properly set up calls. Depending on the type of PICB that is wrong, it might also cause hardware problem.
Corrective Action Taken:	The static version of rPICB is copied into the dynamic version.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Beginning address of the dynamic tuple that is found to be in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	PICB number.
<i>Dump 1</i> =	The dynamic rPICB tuple under audit.
<i>Dump 2</i> =	The static rPICB tuple that is considered correct.

Table 2 Error Code: SGROUP

Error Code:	SGROUP
Error Description:	The PClusgname[] array entry associated with a particular line unit does not contain the correct mask (0xPP00, where PP = PICB + 1).
Possible Error Effect:	If the PClusgname[] entry is wrong, circuit names for this line unit will be miscalculated. This would cause serious problems for growth, maintenance, and operational software. Other audits such as HLSC, LINE, and CHAN depend on the correctness of PClusgname[].
Corrective Action Taken:	The PClusgname[] entry is corrected using the PICB number in static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PClusgname[] entry in error.
<i>Bad Data</i> =	The PICB + 1 value in PClusgname[].
<i>Good Data</i> =	The PICB value in rPICB (plus 1).
<i>Logical Index</i> =	The PICB number.
<i>Dump</i> =	The static rPICB tuple that is considered correct.

PIPESTAT

Software Release: 5E10 and later

1. PIPE STATUS (PIPESTAT) AUDIT

Quad linked packet switched (QLPS) network "pipes" are the communication paths between switching modules (SMs) and the communication module (CM). Information regarding these "pipes" is stored in static ODD and data delivery keys (such as, RLMHPIPE and DDMSGs). Other dynamic data structures (such as SMqlpipedata[] and SMqlstat) summarize this information to represent the current status of the "pipe". This data is also used by switch maintenance in maintaining the SM's copy of QLPS logical link status.

2. DATA STRUCTURES

The PIPESTAT audit makes reference to the following data structures:

- SMdqlstate[]
- SMqlstate[]
- SMqlpipedata[]
- SMqlstat
- SMqpipestate[]
- RLMHPIPE relation (stores MH PIPE static information).

3. AUDIT CHECKS

The following checks are performed by the PIPESTAT audit:

- ZAP = A reset was taken during the audit.
- ODD_DATA = Verify Pipe data with corresponding static ODD in RLMHPIPE.
- RANGE = Verify **nli** or **qtlmkpos** in ODD to be in range.
- STATUS = Verify QLPS network link status data against DD data.
- COUNTS = Verify the counts of ACT (D)QLNKs.
- STATE = Verify Pipe state against static data in RLMHPIPE and RLQENDPTS.
- NOMATCH = Verify **nli** field for an unused entry is set properly.

4. ERROR HANDLING

Different recovery is taken depending upon the error reported. Refer to the **Corrective Action Taken** section of the individual error.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS

Error Description:	One of the following is true (refer to the error address field of the audit report to determine which case applies): - The count of active d-qlinks associated with an endpoint (SMno_dqlnks[endpoint]) is not equal to the number of entries in SMdqlstate[][endpoint] associated with the endpoint which have a state value of SMQLSIS (q-link in service). - The count of active qlinks (SMno_qlnks) is not equal to the number of entries in SMdqlstate[network][endpoint] where "network" ranges from 0 to SMQLPSNUMNETS minus 1, "endpoint" ranges from SMQGP0ENDPT to SMQGP1ENDPT, and the state field has a value of SMQLSIS.
Possible Error Effect:	Maintenance code may take inappropriate action because the count of active (d-)qlinks (that is, the value of SMno_dqlnk[endpoint] or SMbno_qlnks) is incorrect.
Corrective Action Taken:	The audit sets the value of the counter (SMno_dqlnk[endpoint] or SMno_qlnk) equal to the calculated value. The new value appears in the good data field of the audit report.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	SMIMCLMAX implies that SMno_dqlnks[] was being checked. (SMIMCLMAX + SMQGPMAX) implies that SMno_qlnks was being checked.
<i>Bad Data =</i>	The value of the active link counter (SMno_dqlnk[endpoint] or SMno_qlnk).
<i>Good Data =</i>	The count of active (d-)qlinks as calculated by the audit.
<i>Logical Index =</i>	The endpoint (index into SMno_dqlnks[] or SMQGP1ENDPT+1).
<i>Dump =</i>	NA.

Table 2 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The nli field in an unused entry in a pipe data array was not set to the appropriate default value. Specifically: - qph_index but one or more of the following is true: - SMqphpipedata[qph_index-1][network].nli != SMNO_QPH. - SMqphnstat[qph_index].network[network].qpipe0 != 0. - SMqphnstat[qph_index].network[network].qpipe1 != 0. - If good data is RLQPHPIPE, no rIQPHPIPE tuple exists with keys rIQPHPIPE.qnetwork = network (given in the logical key field of the audit report) and rIQPHPIPE.qendpt = rIQPH.qendpt where the rIQPH tuple is keyed by rIQPH.qphindx = QPH index (given in the error address field of the audit report). However, one or more of the following is also true: - SMqphpipedata[qph_index-1][network].nli != SMNO_QPH. - SMqphnstat[qph_index].network[network].qpipe0 != 0. - SMqphnstat[qph_index].network[network].qpipe1 != 0.

	If good data is RLMHPIPE, SMqlpipedata[network].nli != SMNO_MHQP. The entry is known to be unused because no rIMHPIPE tuple exists with a key rIMHPIPE.network = index into SMqlpipedata[].
Possible Error Effect:	The SM on which the audit is running may attempt to use a non-existent connection to the QLPS network. Frames routed over the non-existent connection are lost. The QLPS network may be degraded.
Corrective Action Taken:	If the relation identifier is RLQPH, the audit sets SMqphdpipedata[qph_index][network].nli to SMNO_QPH. If the relation identifier is RLMHPIPE, the audit sets SMqlpipedata[network].nli to SMNO_MHQP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	RLQPH or RLQPHPIPE imply that this field contains the qph index. RLMHPIPE implies that this field is NA.
<i>Bad Data</i> =	RLQPH or RLQPHPIPE imply SMqphdpipedata[qph_idx][network].nli. RLMHPIPE implies SMqlpipedata[network].nli.
<i>Good Data</i> =	Relation identifier.
<i>Logical Index</i> =	QLPS network.
<i>Dump 1</i> =	RLMHPIPE implies SMqlpipedata[network]. RLQPH or RLQPHPIPE imply SMqphnstat[qph_index].network[network].
<i>Dump 2</i> =	RLMHPIPE implies NA. RLQPH or RLQPHPIPE imply SMqphdpipedata[qph_index-1][network].

Table 3 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	Pipe data is inconsistent with static office-dependent data (ODD). The good data field indicates which relation is involved: RLMHPIPE = The following rIMHPIPE tuple is keyed by qnetwork (that is, the QLPS network number) given in the logical key. One or more of the following is true: - SMqlpipedata[qnetwork].nli != rIMHPIPE.nli - SMqlpipedata[qnetwork].nlisublnk != SMQLSUBLNK - SMqlpipedata[qnetwork].qtlmkpos != rIMHPIPE.qtlmkpos RLQPHPIPE = The following rIQPHPIPE tuple is keyed by qnetwork and qendpt [that is, the QLPS network number and a QLPS protocol handler (QPH) endpoint index]. The qendpt value was read from an rIQPH tuple keyed by the QPH index associated with the QPH being checked. The logical key provides the qnetwork and qph_idx values. One or more of the following is true: - SMqphdpipedata[qnetwork].nli != rIQPHPIPE.nli - SMqphdpipedata[qnetwork].nlisublnk != CCQPHSUBLNK - SMqphdpipedata[qnetwork].qtlmkpos != rIQPHPIPE.qtlmkpos

Possible Error Effect:	The SM on which the audit is running may lose connectivity to the QLPS network. The network may be degraded.
Corrective Action Taken:	The audit sets the pipe data fields to the values described in the error description section.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	nli value from the pipe data.
<i>Bad Data =</i>	nli value from the pipe data.
<i>Good Data =</i>	Relation identifier (see error description).
<i>Logical Index =</i>	This field contains the index(s) into the arrays being audited. If the relation identifier is RLMHPIPE, the value is just the network number. If the relation identifier is RLQPHPIPE, the bottom 16 bits contain the network number and the top 16 bits contain the QPH index.
<i>Dump 1 =</i>	Audit's version of the pipe data (from the relation indicated in good data).
<i>Dump 2 =</i>	The pipe data. If good data is RLMHPIPE, this is the contents of SMqlpipedata[network]. If good data is RLQPHPIPE, this is the contents of SMqphnstat[qph_index][network].

Table 4 Error Code: RANGE

Error Code:	RANGE
Error Description:	The audit found an invalid (out-of-range) value in either rLMHPIPE or rLQPHPIPE. The good data field of the error report contains the relation identifier. The error address indicates which field which is dumped in the bad data was implicated: (1) Either the rLMHPIPE.nli or rLQPHPIPE.nli field. (2) Either the rLMHPIPE.qtlkpos or rLQPHPIPE.qlnkpos field. Either the static ODD is corrupted, or the audit code is not supporting all valid values for the implicated field.
Possible Error Effect:	If the ODD is corrupted, the SM on which the audit is running may lose connectivity to the QLPS network. The network may be degraded. Operational code may also assert. If the audit code is wrong, this error appears repeatedly until the code is fixed.
Corrective Action Taken:	None.
Manual Action Required:	A technician should determine whether the implicated field's value is valid (see population rules for the affected relation). The field's value is provided in bad data. The relation identifier is provided in good data. If the field's value is not valid, the technician must update it in the appropriate relation. If the field's value is valid, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Dump Description:	
<i>Error Address =</i>	Error type indicator (1 or 2, see error description).
<i>Bad Data =</i>	Field's value (see error description).
<i>Good Data =</i>	relation identifier (see error description).
<i>Logical Index =</i>	This field contains the index(s) into the arrays being audited. If the relation identifier is RLMHPIPE, the value is just the network number. If the relation identifier is RLQPHPIPE, the bottom 16 bits contain the network number and the top 16 bits contain the QPH index.
<i>Dump =</i>	Pipe data. If good data is RLMHPIPE, this is the contents of SMqlpipedata[network]. If good data is RLQPHPIPE, this is the contents of SMqphnstat[qph_index][network].

Table 5 Error Code: STATE

Error Code:	STATE
Error Description:	The values in one or more of SMqpipestate[network], SMqllstate[network][endpoint], or SMDqlstate[network][endpoint].state are inconsistent with static relations RLMHPIPE and RLQENDPTS. The exact nature of the inconsistency can be determined by looking at the data provided in the audit error report.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit invokes maintenance code to correct the array entry values to agree with the ODD relations RLMHPIPE and RLQENDPTS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	State of the QLPS pipe on this network (SMqpipestate[network]).
<i>Bad Data =</i>	The top 16 bits contain the QLPS logical link state for this network and endpoint combination (SMqllstate[network][endpoint]). The bottom 16 bits contain the state of the d-qlink associated with this network and endpoint combination (SMDqlstate[network][endpoint].state).
<i>Good Data =</i>	The top 16 bits contain the return value from the audit's attempt to read the tuple of RLMHPIPE keyed by "network" (rlMHPIPE.qnetwork = network). The bottom 16 bits contain the return value from the audit's attempt to read the tuple of RLQENDPTS where rlQENDPTS.qnetwork = network and rlQENDPTS.qservice = QLOSDS.
<i>Logical Index =</i>	The top 16 bits contain the endpoint being checked. The bottom 16 bits contain the network being checked.
<i>Dump 1 =</i>	The rlMHPIPE buffer. If the return value given in good data is not GLSUCCESS, this data is NA.
<i>Dump 2 =</i>	The rlQENDPTS buffer.

Table 6 Error Code: STATUS

Error Code:	STATUS
Error Description:	QLPS network link status data is inconsistent with DD data. The data being checked depends on whether MH links or QPH links are being checked. If good data is RLMHPIPE, then MH links were being checked. If good data is RLQPHPIPE, then QPH links were being checked. In the MH case, the audit is checking fields in SMqllstat.network[network]. In the QPH case, the audit is checking fields in SMqphnstat[qph_endpoint].network[network]. The rlMHPIPE tuple is keyed by the value of network (rlMHPIPE.qnetwork = network). The rlQPHPIPE tuple is keyed by network and qph_endpt (rlQPHPIPE.qnetwork = network and rlQPHPIPE.qendpt = qph_endpoint). These values are given in the logical key. In the MH case, one or more of the following was true: • SMQLSTAT.qpipe0 != value determined from rlMHPIPE.nli, rlMHPIPE.qtlkpos, DDRDLI, DDMSGs and DDRNLI0 and DDRNLI1. • SMQLSTAT.qpipe1 != value determined from rlMHPIPE.nli, rlMHPIPE.qtlkpos, DDRDLI, DDMSGs and DDRNLI0 and DDRNLI1. • SMQLSTAT.act_qlps != value determined from DDMSGs and the network value. • SMQLSTAT.qgl0_ms0 != value determined from DDMSGs and the

	network value. - SMQLSTAT.qgl0_ms1 != value determined from DDMSGs and the network value. - SMQLSTAT.qgl1_ms0 != value determined from DDMSGs and the network value. - SMQLSTAT.qgl1_ms1 != value determined from DDMSGs and the network value. In the QPH case, one or more of the following was true: - SMQLSTAT.qpipe0 != value determined from rIQPHPIPE.nli, rIQPHPIPE.qlnkpos, DDRDLI and DDMSGs. - SMQLSTAT.qpipe1 != value determined from rIQPHPIPE.nli, rIQPHPIPE.qlnkpos, DDRDLI and DDMSGs. - SMQLSTAT.act_qlps != value determined from DDMSGs and the network value.
Possible Error Effect:	The SM on which the audit is running may lose connectivity to the QLPS network. The network may be degraded.
Corrective Action Taken:	The audit sets the SMQLSTAT fields to the values determined from the data as described in the error description.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	DDMSGs key.
<i>Bad Data</i> =	DDRDLI key.
<i>Good Data</i> =	Relation identifier (see error description).
<i>Logical Index</i> =	This field contains the index(s) into the arrays being audited. If the relation identifier is RLMHPIPE, the value is just the network number. If the relation identifier is RLQPHPIPE, the bottom 16 bits contain the network number and the top 16 bits contain the QPH index.
<i>Dump 1</i> =	Pipe data. If good data is RLMHPIPE, this is the contents of SMqlpipedata[network]. If good data is RLQPHPIPE, this is the contents of SMqphnstat[qph_index][network].
<i>Dump 2</i> =	Correct SMQLSTAT values as determined by the audit.
<i>Dump 3</i> =	SMQLSTAT values from the data being audited (see error description).

Table 7 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write-protect violation, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None. The audit has no record of what data it was checking prior to the interrupt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

PKB

Software Release: 5E10 and later

1. PACKET BUFFERS (PKB) AUDIT

The Operator Services Position System (OSPS) message transport feature provides reliable message communication between OSPS directory assistance (DA) feature-level software in the integrated services digital network (ISDN) switching module processor (SMP) and OSPS terminal equipment. Two audits are needed to maintain the integrity of the dynamic data in the switching modules (SMs) for the network side, the packet buffers audit (PKB) and the supplementary protocol block (SPB) audit.

The PKB audit is part of the OSPS audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The audit uses two data structures, the dynamic packet buffers relation (RLPKB) and the dynamic supplementary protocol block relation (RLSPB). The RLSPB relation owns the rIPKB tuples, as Figure 1 shows.

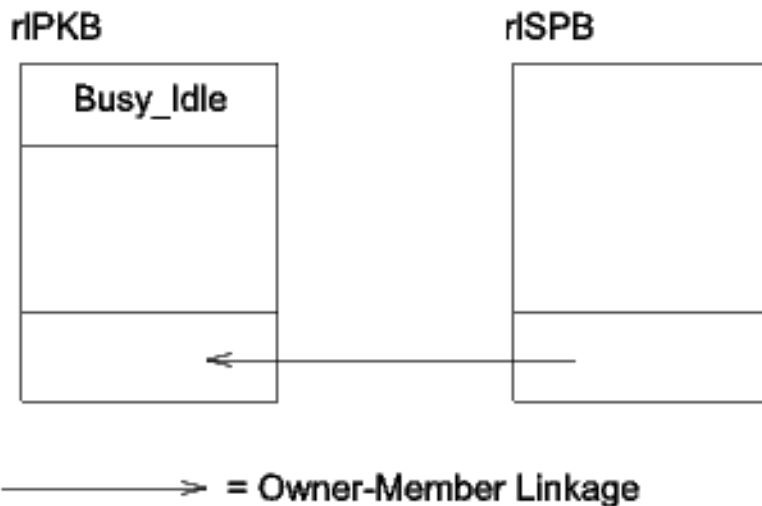


Figure 1 (PKB-1) Data Structures of the PKB Audit

3. AUDIT CHECKS

The PKB audit calls the indirect access sequential key driver (DRINASEK) to step through the tuples of RLPKB. The driver checks the linkages between RLPKB and RLSPB. If the linkage checks pass, the audit calls its semantic check function to make the following checks.

- (a) When the busy/idle state is idle, the tuple must be owned by the idle head cell in rISPB.
- (b) When the tuple is busy, it must be linked to a tuple of rISPB.
- (c) If the busy_idle field is set to something other than busy or idle, the audit will report an error.

4. ERROR HANDLING

When the audit detects errors, it prints the contents of RLPKB and initializes the bad tuple.

The PKB audits schedules the SPB audit as a related audit when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The busy_idle field is set to an invalid state.
Possible Error Effect:	The rIPKB tuple will be lost to the system.
Corrective Action Taken:	The rIPKB tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPKB.
<i>Bad Data =</i>	pkb_ptr->busy_idle.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPKB number.
<i>Dump =</i>	rIPKB.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIPKB is not idle when it should be.
Possible Error Effect:	The rIPKB will be lost to the system.
Corrective Action Taken:	The associated calls are torn down and the corresponding rIPKB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPKB.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	Expected owner's physical key.
<i>Logical Index =</i>	Logical key of rIPKB.
<i>Dump =</i>	rIPKB tuple.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIPKB is not busy when it should be.
Possible Error Effect:	The rIPKB will be lost to the system.
Corrective Action Taken:	The associated calls are torn down and the corresponding rIPKB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPKB.
<i>Bad Data =</i>	Actual owner's invalid physical key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIPKB.
<i>Dump =</i>	rIPKB tuple.

Table 4 Error Code: NOTBUFTYPE

Error Code:	NOTBUFTYPE
Error Description:	The rIPKB tuple buffer type is incorrect. The two buffer types are transmit and receive.
Possible Error Effect:	The rIPKB tuple will be lost to the system.
Corrective Action Taken:	The associated calls are torn down and the corresponding rIPKB tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPKB tuple.
<i>Bad Data =</i>	rIPKB tuple attribute buftype.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	Logical key of rIPKB tuple (pkbkey).
<i>Dump</i> =	rIPKB tuple.

PMDB

Software Release: 5E10 and later

1. PROCESS MESSAGE DATA BLOCK (PMDB) AUDIT

The process message data block (PMDB) is used to store incoming and outgoing messages for some terminal processes. The PMDB is available to any operating system for distributed systems (OSDS) terminal process on the switching module (SM) only.

The number of PMDBs is engineered on a per configuration basis. The number of PMDBs is the number of terminal process stacks plus a few more. A PMDB's index is the same as the index of the SCB it is associated with. The PMDB resource is not write protected. This resource is not available to system processes.

The process message data block audit runs in the SM environment only.

2. DATA STRUCTURES

In addition to auditing the PMDB (OSbpmdb[]), during the audit recovery the following data is also referenced:

- The array of SCBs (OSscb[]), which corresponds one-to-one with the PMDB.
- The array of process control blocks (OSbpcb[]), any of which may own a single SCB.

Figure 1 shows how a PMDB is associated with a stack.

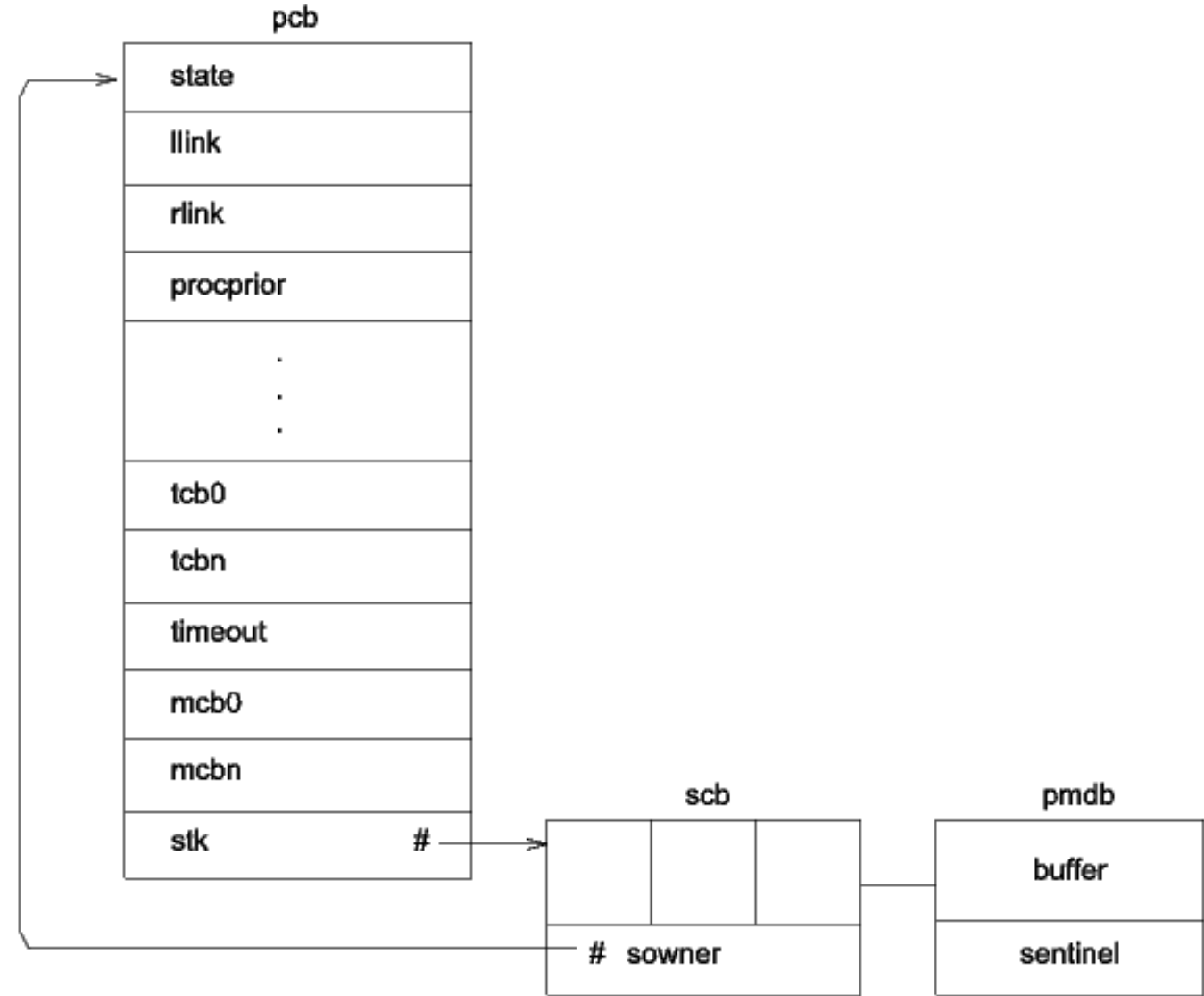


Figure 1 (PMDB-1) Stack and PMDB Association

3. AUDIT CHECKS

The PMDB audit does not use any audit driver. If the audit has just been entered:

ZAP = Check for system interrupt handling.

When each PMDB is being entered:

PMDBSNTL = Check the sentinel field of the PMDB.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: PMDBSNTL

Error Code:	PMDBSNTL
Error Description:	The sentinel field (field which detects overwrites of PMDB) of a PMDB has been corrupted.
Possible Error Effect:	The data in the PMDB may not be correct. The messages could get corrupted or lost.
Corrective Action Taken:	The associated process (if any) will be purged and the sentinel field will be reset to the correct value (OSPMDBSNTL).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the sentinel field of the PMDB.
<i>Bad Data</i> =	Contents of the sentinel field of PMDB.
<i>Good Data</i> =	The correct value of the sentinel field (OSPMDBSNTL).
<i>Logical Index</i> =	The PMDB's Index.
<i>Dump 1</i> =	The PMDB.
<i>Dump 2</i> =	The SCB (key is the same as the PMDB's).
<i>Dump 3</i> =	The PCB (keyed by the SCB's "sowner" field).

Table 2 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as a result of a write protect, out of range read, or invalid operation, by foreground or by any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will idle the last PMDB it was checking (if any), abort and be restarted in directed mode. If there is process, it will purge. Otherwise, it will just be restarted in directed mode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PMDB (if any).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The PMDB's Index (if any).
<i>Dump</i> =	The PMDB (if any).

PMFACGRP

Software Release: 5E10 and later

1. PERFORMANCE MONITORING FACILITY TO GROUP (PMFACGRP) AUDIT

Each DS1/DFI-2 facility is associated with a performance monitoring group ID. The PMFACGRP audit verifies that for each protected rIPMFACGRP tuple there exists an unprotected rIFACPM tuple.

2. DATA STRUCTURES

The RLPMFACGRP relation associates a DS1 or DFI-2 facility with a performance monitoring group. Each facility has one rIPMFACGRP tuple and one rIFACPM tuple.

The IDCU performance monitoring data structures are shown in Figure 1. The DFI-2 performance monitoring data structures are shown in Figure 2.

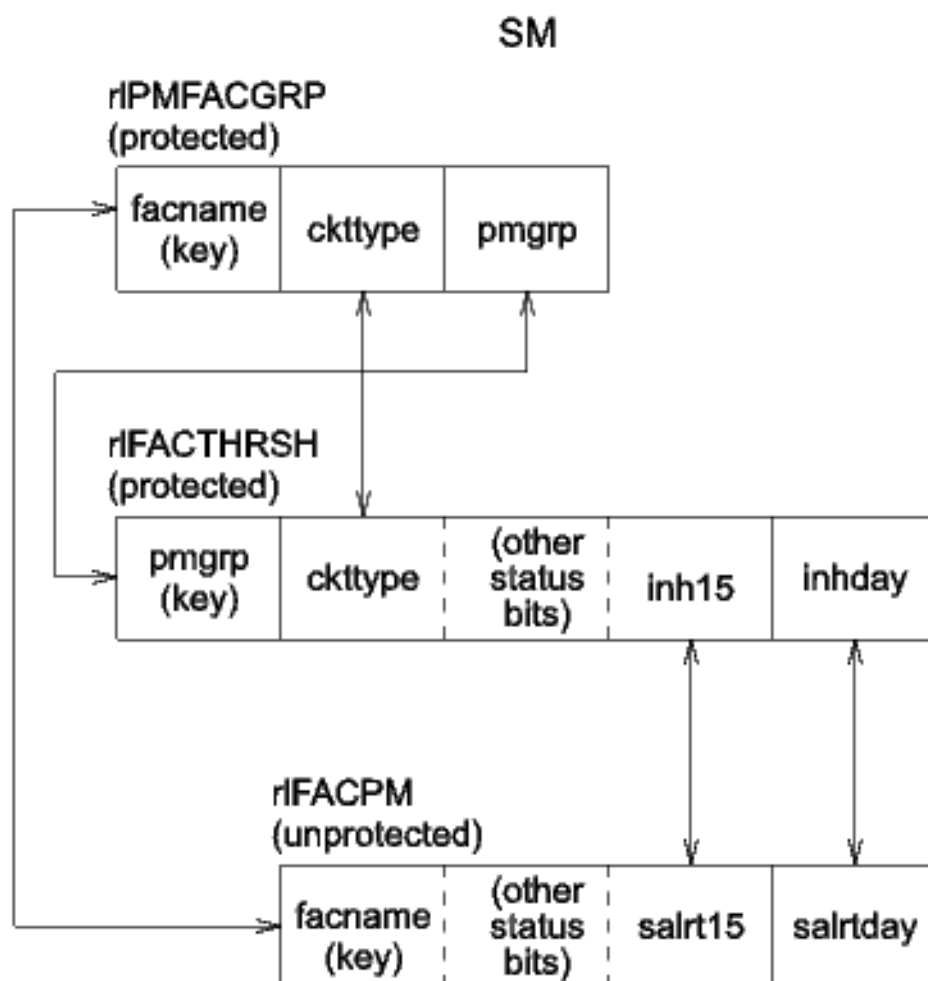


Figure 1 (PMFACGRP-1) IDCU Performance Monitoring Data Structures

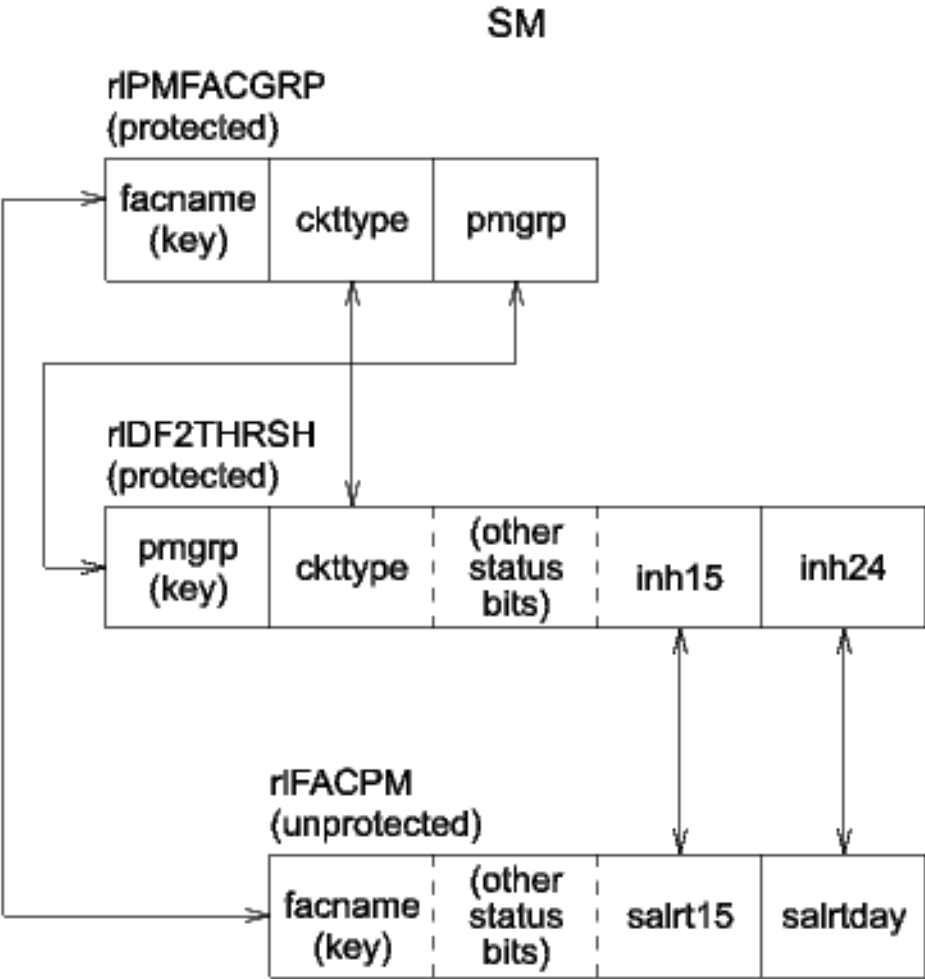


Figure 2 (PMFACGRP-2) DFI-2 Performance Monitoring Data Structures

3. AUDIT CHECKS

The PMFACGRP audit uses the DRSTATIC audit driver to step through the RLPMFACGRP relation. The PMFACGRP audit uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given.

For each tuple, the PMFACGRP audit makes the following checks:

LOSTTUP = Check for missing tuples.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP

Error Description:	An unprotected tuple in RLFACPM relation with the facility name as key does not exist but a corresponding protected tuple in RLPMFACGRP does exist.
Possible Error Effect:	Performance monitoring status is missing for this facility.
Corrective Action Taken:	The missing rIFACPM tuple is inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPMFACGRP tuple.
<i>Bad Data</i> =	Facility name (rIPMFACGRP.facname).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Facility name (rIPMFACGRP.facname).
<i>Dump</i> =	NA.

PMS

Software Release: 5E10 and later

1. PLANT MEASUREMENTS SYSTEM (PMS) AUDIT

The plant measurements system (PMS) audit verifies the integrity of the incore database plant measurements database (PMDB). In addition, the minimum number and types of sets in the PMDB and the minimum number and required records of selected sets are checked. The audits include a low-level access (LLA) audit and a record integrity audit.

If errors are found in the incore PMDB, the records that were found to be mutilated are retrieved from one of two copies in disk. If both PMDB copies are found to be mutilated, a new copy is created and copied to incore and disk.

NOTE: Of the two PMS audits (PMS 1 and PMS 2), only PMS 1 can be requested manually. If the PMS 1 audit cannot correct errors that occur, PMS 2 will run automatically.

1.1 Requesting PMS Audits

To request an audit of the PMDB, use the following input message:

```
AUD:PMS=1[:a|:b|:a,b];
```

where:

a = Indicates whether an audit should correct errors. Valid values are:

CORR = Audit should correct errors (default).

NCG = Audit should not correct errors.

b = Indicates what type of report generated. Valid values are:

DETL = Generates a raw data report for each error found plus an audit completion message (default). A limited number of raw data reports are generated.

SUM = Generates only an audit completion message when the audit terminates. Raw data reports are not generated for any errors that are found.

To interpret system responses to the AUD:PMS input message, refer to the APP:AUD Appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD PMS Output Messages and Error Reports

The AUD:PMS output message reports the termination status of the PMS audit and/or any errors found while executing the audit.

When the manual audit finds serious errors which it cannot correct, the system integrity monitor (SIM) demands the running of this audit to perform the major recovery, which causes the database to be locked, and the output message REPT SIMCHK 161 to be printed (refer to the Output Messages manual [235-600-750] for detailed information on REPT:SIMCHK). Major recovery actions may include retrieval of a backup copy of the database from disk or creation of a new database if no valid backup copy exists.

For a detailed explanation of the AUD:PMS output message, see the Output Messages manual (235-600-750).

Table 1 lists the PMS audit error messages and their associated categories.

Table 1 PMS AUDIT ERROR MESSAGES

Category	Error Number	Meaning
0	0	lcall failure.
0	1	Fatal, cannot create new database.
0	2	Fatal, cannot copy new disk database to incore.
0	3	New database created; reboot the system.
1	1000	setid_rec mutilated.
1	1001	Requesting demand audit to recover.
2	2000	setid_rec mutilated.
2	2001	Fatal, incore and disk databases mutilated, new database will be created.
3	3000	Invalid return value from lla_audit.
3	3001	Invalid dispatched message.
3	3002	Cannot release the incore database.
3	3003	Cannot release the disk database.
3	3004	Cannot unlock incore database.
3	3005	setid_rec corrected.
3	3006	lla_gtrid failure.
3	3007	Invalid set type.
3	3008	Cannot lock the incore database.
3	3009	Disk database written to incore.
4	4000	Noncorrectable internal structure.
5	5001	Cannot attach to incore database.
5	5002	lla_ainit failure.
5	5004	Cannot unlink /etc/pmdb.aud.
5	5005	Invalid execution mode.
5	5006	Invalid audit member number.
5	5007	Cannot attach to disk database.
5	5008	lla_opensq failure.
5	5009	Cannot get setid_rec.
5	5011	lla_rdget failure.
5	5012	lla_uprid failure.
5	5013	lla_opnset failure.
5	5014	lla_seqset failure.
5	5016	Cannot open /etc/pmdb.
5	5017	Cannot open /etc/pmdb.aud.
5	5018	Invalid disk name.
5	5019	Invalid audit type.
5	5020	Fail to write incore database to disk.
5	5021	fstat failure.
5	5022	Fail to write disk database to incore.
5	5023	plib failure.
5	5024	makeseg failure.
5	5025	msgenab failure.
5	5026	Cannot receive the message from SIM.
5	5027	Internal structure corrected, refer to /etc/log/pmsaud.
5	5028	Cannot attach to secondary disk database - /etc/pmdb.aud.
5	5029	Cannot falloc disk database.
5	5030	Restored incore database from disk.
5	5031	Primary disk database is bad.
5	5032	Secondary disk database is bad.
5	5033	Restored primary disk database.
5	5034	Restored secondary disk database.

The DATA1 through DATA4 fields are always zero in AUD:PMS error messages. Additional information is provided in the supplementary data portion of each message. Complete lists of supplementary data messages may be found on the AUD:PMS output message page and the APP:LLA appendix in the Appendixes section of the Output Messages manual (235-600-750).

If the audits are unable to interface with the audit control subsystem, they output one of the following error messages:

1.3 Responding to AUD PMS Error Reports

If the AUD:PMS error code indicates that the primary and/or secondary disk database(s) is/are bad and the

condition is not corrected when the audit is run in correcting mode, check to make sure enough contiguous space exists on the filesystem to create the database(s). refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual, and correct the problem. Once this problem is corrected, rerun the audit in correcting mode.

If the audit was run in the noncorrecting mode, rerun the audit in the correcting mode. If errors still occur, save the output and refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual for contacting the next level of assistance.

PORTCLSS

Software Release: 5E10 and later

1. PORT CLASS (PORTCLSS) AUDIT

The PORTCLSS audit will perform the static rIORTCLSS tuple to dynamic rIQPRTCLSS tuple checks.

2. DATA STRUCTURES

The static data structure that is being audited is the rIORTCLSS tuple. A brief description of structure fields is as follows:

portcode	= Code representing a block of ports.
cktname	= Circuit name of hardware owning the portcode.
unitnum	= Unit number of the IDCU.
unittype	= Hardware unit type.
rtnum	= Number of the RT relative to the IDCU.
codenum	= Sequence number of portcode relative to owning hardware.
code_type	= Type of portcode (Analog/D, B1, or B2).

3. AUDIT CHECKS

The audit sequence is as follows:

LOSTODD	= Verify that an rIICKT2CODE tuple exists.
DYN_STAT	= The rIQPORTCLSS tuple is audited against rIICKT2CODER and rIORTCLSS tuple.
DLOSTTUP	= The rIQPRTCLSS tuple existence check.

4. ERROR HANDLING

The general error handling mechanism of the PORTCLSS audit is to report the error and correct it. If there is a static ODD inconsistency, then the audit will also report the error which will require the intervention of service personnel to fix the static ODD.

The audit produces the following error code(s).

Table 1 Error Code: DLOSTTUP

Error Code:	DLOSTTUP
Error Description:	The audit found a missing rIQPRTCLSS tuple for a given rIORTCLSS tuple.
Possible Error Effect:	The call processing will not be able to use the rIQPRTCLSS tuple.
Corrective Action Taken:	The audit will insert an rIQPRTCLSS tuple in the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIORTCLSS pointer.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIORTCLSS tuple.
<i>Dump 1</i> =	The dump of the rIORTCLSS tuple.

<i>Dump 2 =</i>	The dump of the rICKT2CODE tuple.

Table 2 Error Code: CYN_STAT

Error Code:	CYN_STAT
Error Description:	The PORTCLSS audit detected an inconsistency between the dynamic rQPRTCLSS tuple and the two static tuples that it is being audited against (rICKT2CODER and rIPORTCLSS).
Possible Error Effect:	The call processing will not be able to use the rQPRTCLSS tuple.
Corrective Action Taken:	The audit will correct the rQPRTCLSS to match the information that is derived from the static rICKT2CODE and rIPORTCLSS tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	The dump of the rIPORTCLSS tuple.
<i>Dump 2 =</i>	The dump of the rQPRTCLSS tuple.
<i>Dump 3 =</i>	The dump of the rICKT2CODE tuple.

Table 3 Error Code: LOSTODD

Error Code:	LOSTODD
Error Description:	The audit detected a missing static ODD tuple in the RLCKT2CODE relation.
Possible Error Effect:	The call processing will not be able to use the ports associated with that "portcode" at fault.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel need to look the ODD for a missing rICKT2CODE tuple.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The circuit name which is the key to the rICKT2CODE tuple that is missing.
<i>Logical Index =</i>	The "condenum" field which is also a key to the rICKT2CODE tuple.
<i>Dump =</i>	The rIPORTCLSS tuple will be dumped.

PORTLA-A

Software Release: 5E10 - 5E11

1. PORT LINKAGE AREA (PORTLA) AUDIT

Each port has a port linkage area tuple that describes the status and associations of the port. These port linkage areas contain information such as the owner of the port, processes that are waiting to access the port, the status of the port (precut or postcut, and so forth), and a preemption flag that shows whether the port is being preempted.

The port linkage area audit (PORTLA) verifies the data contained in the port linkage areas. When the PORTLA audit is invoked, it checks all ports (both lines and trunks).

2. DATA STRUCTURES

The data structure checked by the PORTLA audit is the port linkage area relation (RLPORTLA). The RLPORTLA relation has one-to-one linkages with the pending history relation (RLPHISTORY) and the process control block linkage area relation (RLPCBLA). It also has owner-member linkage with the coupler relation (RLCPLR) and member linkage with the port list relation (RLPRTLST). The RLPRTLST relation has 16 head cells, each of which owns a list of port linkage areas. Each list indicates a particular port status (for example, in service, out of service, to be idled, and so forth). The port linkage area list item contains an ID that indicates the list to which the port linkage area belongs. The PORTLA audit uses the feature control port name to service class relation (RLFC_PORTTYP) to perform an existence check. Analog modem pool ports will verify the existence of the related packet switch rIORTLA tuple and its corresponding rIDPB tuple. Additional multipoint modem pool checks have been added for the following data structures:

- The terminal equipment data block relation (RLTEDB).
- The packet switching X.25 Level 3 data for DSL's relation (RLPSX25LN).
- The terminal endpoint identifier linkage area relation (RLTEILA).

The PORTLA audit added checks of the following SLC[®] 96 and network management data structures:

- The trunk group selective incoming load control relation (RLTKGSILC).
- The switch maintenance equipment status table relation (RLSMEST).
- The circuit data block common area relation (RLCDBCOM).
- The SLC[®] 96 equipage relation (RLS96EQUIP).
- The SLC[®] 96 T1 usage relation (RLS96T1USG).

Figure 1 shows these data structures.

The PORTLA audit insures the integrity of the port linkage area relation (RLPORTLA). A field in the rIORTLA tuple checks for datalink trunks and lines. If the port is of a particular service class, the datalink bit should be set. If it is not, this check prints an error and then sets the bit. It then continues processing and it does not recover the tuple.

3. AUDIT CHECKS

The PORTLA audit uses the static driver (DRSTATIC) to step through the RLPORTLA relation.

- (a) Initially, the audit calls a dynamic to static existence check function to perform the following checks.

- (1) This existence check verifies that, for every dynamic rIORTLA tuple, a static port name to service class tuple (rIFC_PORTTYP) exists. If the static rIFC_PORTTYP tuple does not exist, the extra rIORTLA tuple is deleted (EXTRATUP).
 - (2) For each tuple in RLPORTLA that is associated with a circuit, the existence check verifies that a corresponding tuple exists in the static circuit data relation (RLCKTDATA) for those service classes that should have a tuple (TRUNK_LINE).
- (b) The audit calls a semantic check function to perform the following checks.
- (1) The audit checks to see that the rIORTLST owning head cell of the port linkage area, as determined from the port linkage area's owner linkage item, agrees with the list item (LIST).
 - (2) The audit checks the cutover status of the office and port. If the office is in a cutover state, the audit checks that the rIORTLA tuple is on the correct list (CUTOVER).
 - (3) The port status (pstatus) item contains the primary maintenance status of the port. The audit verifies that the list to which the port linkage area belongs is consistent with the pstatus item (PSTATUS).
 - (4) The audit verifies that the hunt item in the rIORTLA relation is consistent with the hunt item in the rIFC_PORTTYP relation (ODD_DATA).
 - (5) The audit verifies that the port type (ptype) item in the rIORTLA relation is consistent with the port type (porttype) item in the rIFC_PORTTYP relation (ODD_DATA).
 - (6) The audit verifies the datalink item with valid service class (BLINK).
 - (7) If the port is a digital carrier port, the audit verifies that the list to which the port linkage area belongs is consistent with the state of the digital facility interface (DFI) circuit. An active red or yellow alarm should correspond to ports on the facility failure list. Family-of-equipment and to-be-idled lists are also allowed because they are higher priority lists. If both the red and yellow alarms are inactive, the port linkage area can be on any list except the facility failure list (ALARM).
 - (8) Common channel signaling ports are verified to be on valid lists by this audit (WRONGLIST).
 - (9) If the rIORTLA tuple is on a network management list, the corresponding rITKGSILC tuple is read (MISSTUP). Several fields, including the port and quarantine state fields, are checked (PORT_TM).
 - (10) The audit verifies that OSPS Autoquote DSL's have a mate analog rIORTLA tuple (OOS_MATE).
 - (11) The audit verifies that nailup ports have a mate rINAILUP tuple.
 - (12) During linkage checking of the rIORTLA tuple the function AUptlchk() performs point-to point-back checks between the rIORTLA and rIADB relations. If an error is found, an error report is printed to the Receive/Read Only Printer (ROP) and recovery is started.
 - (13) PORTLA audits the linkage between the rIORTLA relation and the rIADB relation for shared Directory Number (DN) analog. When the PORTLA audit detects the error, the rIORTLA tuple is linked to the rIADB tuple and the PORTLA audit checks to make sure the sh_analog bit in the rIFC_PORTTYP relation is set to DBYES. If sh_analog bit is not set, the PORTLA audit prints an error on the ROP and returns AUFALSE to start recovery (BDLINK).
 - (14) A port linkage area may be linked to one process control block linkage area directly or to one or more process control block linkage areas by means of couplers. Also, it may not be linked to any process control block linkage areas. The port linkage area state item indicates which of these two states exists.

The audit checks to see that the state item agrees with the actual linkage (STATE_IDL or STATE_LNK or STATE).

- (15) In a remote switching module (RSM) environment, if a port is idle, hunted, and in service, the routing and terminal allocation (RTA) subsystem is called to mark the port idle in the hunted port status relations.
 - (16) If the logical port type field of rIFC_PORTTYP indicates that this is a SLC[®] 96 special line, the rIS96T1USG relation is checked to ensure that one dedicated time assignment unit (TAU) is associated with the port (BADTSCOUNT). If it is determined that one does not exist (or more than one exists), the current hardware connection is torn down and a new one is established (SLCLOSTTUP). If an rIS96SHELF tuple is not found, the error is reported (SHELFREAD).
 - (17) For SLC[®] 96 ports, the audit determines whether the port is associated with a 4AB1 or 4AC1 line unit or SLC[®] 96 digital carrier line unit (DCLU). If the port is associated with a DCLU, the key of the rIS96EQUIP tuple is calculated and the tuple is read. The bank status is checked to determine the grid identification, the associated rSMEST tuple is read (SMEST), and the port is conditioned.
 - (18) There is a list of precut status ports contained in the switching module (SM) protected memory. The audit calls a function to determine whether the port associated with the port linkage area being checked is on this list. If it is, the audit checks whether the port linkage area is linked to the rIPRTLST head cell that owns precut status port linkage areas.
 - (19) If the port is an ISDN port, the audit checks the parent circuit(s). It reports an error if the port's maintenance state is inconsistent with the maintenance state of the parent circuit. The audit also checks the validity of family-of-equipment states.
 - (20) This audit also has special checks for modem pool ports. If the port is a modem pool port, the analog and corresponding packet switch ports will be synchronized. For multipoint modem pool ports, the CPE status will be synchronized.
- (c) The PORTLA audit then calls a function to maintain consistency between the monitor busy & idle (B&I) data table, TMbitable[] and the rIPORTLA tuple. The TMbitable[] contains the port and TM process ID information for any ports involved in monitoring using the monitor (B&I) feature.

4. ERROR HANDLING

For all errors except those encountered when the audit is running in the initialization mode, the port linkage area is placed on the to-be-idled list (this list is owned by an rIPRTLST head cell). At the same time, a terminal maintenance process is activated. This process checks the hardware state and the contents of the switch maintenance equipment status table relation (RLSMEST) to determine what the status should be for a given port. For circuits marked out of service (OOS) in rSMEST, the associated ports are given an OOS, family of equipment (FE) status. DFI ports with active red or yellow carrier alarms are given an OOS, facility failure (FAF) status. Ports whose scanning circuitry indicate that they are off-hook are given an OOS, high and wet (HW) status. All other ports are given an in-service status.

For errors encountered when the audit is running in the initialization mode, the port linkage area is linked to the in-service list except for the ports as determined by the service class. Common channel signaling (CCS) trunks are linked to the disabled list and recovered by terminal maintenance (TM) subsystem. D-channels, inter-switch trunks, ISM nailups, X.75 and X75' trunks, and packet switching B-channels are linked to the DSL initialization list and recovered by the TM subsystem. B-channels are linked to the D-channel out-of-service list if the D-channel is out-of-service, else they are linked to the in-service list. In all cases except during initializations, errors are reported by means of output messages.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALARM

Error Code:	ALARM
Error Description:	The list the rIORTLA is on is inconsistent with the state of the digital carrier alarm. Digital carrier ports only.
Possible Error Effect:	If carrier alarm is ON, the port is incorrectly in service and if used by call processing, would cause call failures. If carrier alarm is OFF, the port is incorrectly out of service and thus unavailable for call processing use.
Corrective Action Taken:	The port will be placed on the to-be-idled list for later idling. The software maintenance status of the port will be made consistent with the hardware (carrier) alarm state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	List owner for rIORTLA tuple.
<i>Good Data =</i>	Carrier group alarm status.
<i>Logical Index =</i>	Key of the rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIORMEST tuple.

Table 2 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rIORMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rIORMEST tuple.

Table 3 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	An rIORTLA exists for a port whose service class indicates that an rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Service class of the port.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 4 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by

	recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 5 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	An rSMEST tuple for the circuit under audit or the parent/mate circuit for this circuit was found to be in an invalid state with respect to the information in the MRA deferred work recovery queue. The qual_2 state of SMAUOOS is used to indicate when audits or SI has recovered a circuit, parent or mate to the MRA queue. Refer to <i>Good Data</i> for details.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly by MRA.
Corrective Action Taken:	The audit will attempt to recovery the circuit using the MRA queue and/or leave it in a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	qual_2 field in the SMEST tuple.
<i>Good Data</i> =	Return value from AUMRARET (au/hdr/AUglobckt.L).
	0 = Circuit was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	2 = Parent/mate was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	4 = Circuit is marked AUOOS but not in queue, parent is not in queue, circuit/parent/mate is not MRA BUSY.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	parent/mate rSMEST tuple.

Table 6 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	An rDS0B tuple is involved in a hairpin connection but the rPORTLA.dsob field does not have the correct value.
Possible Error Effect:	Call processing will not be able to use that rPORTLA tuple.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	rDS0B.use field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The DS0B key.
<i>Dump 1</i> =	rDS0B tuple.
<i>Dump 2</i> =	rPORTLA tuple.

Table 7 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The rDS0B tuple is involved in a hairpin or nailup connection but the rDS0B.portname field tuple is not matching the rNAILUP.port1 field.
Possible Error Effect:	There will be no hairpin or nailup connection established since a wrong port is

	being used.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDS0B tuple.
<i>Bad Data =</i>	The rIDS0B tuple key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIDS0B tuple.
<i>Dump 1 =</i>	rIDS0B tuple.
<i>Dump 2 =</i>	rIORTLA tuple.

Table 8 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	For CCS7 trunks, there are four possible conditions for which this error is reported. (1) There is a reset in progress, but the reset_inprog bit in PORTLA is not set to DBYES. (2) There is a reset in progress with the reset_inprog bit in PORTLA set to DBYES, but the port is not owned. (3) There is no reset in progress, but the reset_inprog bit in PORTLA is set to DBYES. (4) There is no entry for this port, but the reset_inprog bit in PORTLA is set to DBYES. For CCS6 trunks, the following fields in PORTLA are not set correctly. These fields include protocol, cic, pci, reset_inprog.
Possible Error Effect:	Incorrect port usage.
Corrective Action Taken:	If the problem is in regards to the reset_inprog bit, then the bit will be set to its correct value. If the error is in regards to the CCS6 trunk, then the following will occur. protocol will be set to DBCCIS6, cic will be set to CCNULCIC, pci will be set to CCNULPCI, and reset_inprog will be set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple for CCS6 trunks. Error flag value for CCS7 trunks.
<i>Bad Data =</i>	Value of in_sigtype from rIFC_TRUNK for CCS6 trunks. Value of mtype returned from TM for CCS7 trunks.
<i>Good Data =</i>	Value of out_sigtype from rIFC_TRUNK for CCS6 trunks. Value of type returned from TM for CCS7 trunks.
<i>Logical Index =</i>	Logical key for the rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 9 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	There are two possible errors: (1) The port is on the TBI (to be idled) list and is still waiting to be run; but the tbi_flag is set, indicating that the job is running. (2) The port is not on the TBI list and the flag is set.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	There are two possible solutions.

	(1) The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status. (2) The tbi_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The to-be-idled flag of the rIORTLA tuple (rIORTLA.tbi_flag).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 10 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The D-port is linked to the unassigned list but the service class taken from static data does not indicate that the port should be on unassigned list.
Possible Error Effect:	The rIORTLA is unavailable for call processing.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to the proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Owner of rIORTLA tuple.
<i>Good Data</i> =	Service class of port.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 11 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	A SLC [®] 96 time slot was found to have more that one time assignment unit assigned to it.
Possible Error Effect:	Possible origination problems for some mode II SLC [®] 96 special lines.
Corrective Action Taken:	Reconstruct the rIS96T1USG tuple with one time assignment unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS96T1USG tuple.
<i>Bad Data</i> =	Count of T1TS lines for this time assignment unit.
<i>Good Data</i> =	Number of time slots allowed (1).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIS96T1USG tuple.

Table 12 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An rIDS0B tuple was used for hairpin or nailup connection but the rIDS0B.use field was not set to indicate that.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDS0B tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	rIDS0B tuple.
<i>Dump 2</i> =	rIORTLA tuple.

Table 13 Error Code: BAS_STAT

Error Code:	BAS_STAT

Error Description:	The rIORTLA tuple is on the wrong list. The port could be stuck on an out-of-service list when it should be in-service or on an is-service list when it can not be used because of the parent circuit. There are two possible errors: (1) The port is not OOS-FE, but the parents circuits are OOS. (2) The port is OOS-FE, but the parents circuits are not OOS. If the port is on a IDCU TR08 the audit will dump the shelf/bank that was checked. For analog ports on an IDCU TR303, one DS1 will be dumped but all DS1's on the RT303 will be checked.
Possible Error Effect:	Either the port will be used for a call which cannot complete because of out-of-service equipment or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	List owner of rIORTLA tuple.
<i>Good Data =</i>	Logical port type.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rSMEST tuple with port as key.
<i>Dump 3 =</i>	rSMEST tuple for PH if available.
<i>Dump 4 =</i>	rSMEST tuple for PSU if available.

Table 14 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The rIORTLA was found linked to the rIADB tuple when static data (rIFC_PORTTYP) did not have the shared analog bit set.
Possible Error Effect:	Call processing cannot access the port correctly.
Corrective Action Taken:	The rIORTLA tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Linkage field to the rIADB tuple.
<i>Good Data =</i>	Shared analog bit in rIFC_PORTTYP tuple.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 15 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The packet switching modem pool port is not bound (rITEDB tuple to rTEILA tuple).
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITEDB tuple.
<i>Bad Data =</i>	Key to rTEILA tuple (0 means no rTEILA tuple).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical Key of rITEDB tuple.
<i>Dump =</i>	rITEDB tuple.

Table 16 Error Code: BINDWRONG

Error Code:	BINDWRONG
Error Description:	The packet switching modem pool port is not bound to the correct rITEDB tuple.
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call

	processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA tuple.
<i>Bad Data =</i>	TEDB key from rTEILA tuple.
<i>Good Data =</i>	Key of rTEDB tuple (logical port).
<i>Logical Index =</i>	Logical key of rTEILA tuple.
<i>Dump =</i>	rTEILA tuple.

Table 17 Error Code: BLINK

Error Code:	BLINK
Error Description:	The rPORTLA service class was of the datalink type but the datalink field was not set to DBYES.
Possible Error Effect:	The line or trunk would not be treated as a datalink for removes/restores.
Corrective Action Taken:	The datalink bit will be set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rPORTLA tuple.
<i>Bad Data =</i>	datalink field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rPORTLA tuple.

Table 18 Error Code: BTRANCK

Error Code:	BTRANCK
Error Description:	The associated D-port is out-of-service and this B-port is not on a list to indicate the unavailability of the D-port. This is considered to be valid for a short transient period. This error is not reported for at least 5 minutes.
Possible Error Effect:	Call processing cannot complete service on this port.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPORTLA tuple.
<i>Bad Data =</i>	Port owner.
<i>Good Data =</i>	D-port owner.
<i>Logical Index =</i>	Logical key of rPORTLA tuple.
<i>Dump 1 =</i>	rPORTLA tuple.
<i>Dump 2 =</i>	D-channel rPORTLA tuple.

Table 19 Error Code: CADNPORT

Error Code:	CADNPORT
Error Description:	The port is either on the CADN list when the cadn field in FCPORTTYP is set to TMIIS or the port is not on the CADN list when the cadn field in FCPORTTYP is not TMIIS. For both of these cases, there error is only reported if the port is not owned.
Possible Error Effect:	The port may not be able to be used by call processing.
Corrective Action Taken:	The rPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rPORTLA tuple.
<i>Bad Data =</i>	cadn field in rFC_PORTTYP tuple.
<i>Good Data =</i>	list field in rPORTLA tuple.
<i>Logical Index =</i>	Logical key for the rPORTLA tuple.
<i>Dump =</i>	rPORTLA tuple.

Table 20 Error Code: CKTDATA (1)

Error Code:	CKTDATA (1)
Error Description:	The PORTLA audit is attempting to read a nonexistent rCKTDATA tuple. The

	key to the rCKTDATA tuple is derived from the rFC PORTTYP tuple.
Possible Error Effect:	Call processing could be affected.
Corrective Action Taken:	The rPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 21 Error Code: CKTDATA (2)

Error Code:	CKTDATA (2)
Error Description:	The rCKTDATA tuple is missing from the database. The key used to read the tuple was derived from the path2_ds0.ds1name in the rNAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rNAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rCKTDATA tuple.
<i>Dump 1</i> =	First rNAILUP tuple
<i>Dump 2</i> =	Second rNAILUP tuple

Table 22 Error Code: CUTOVER

Error Code:	CUTOVER
Error Description:	A port is on the wrong list based on the office's or this ports cutover status.
Possible Error Effect:	Pre-cut ports could incorrectly be receiving service from this office or post-cut (active) ports could be denied service.
Corrective Action Taken:	Port is conditioned and returned to correct list (maintenance status).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	List owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 23 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit failed a read of the rDSLEQUIP or rIISMPATH relation associated with this rPORTLA.
Possible Error Effect:	There could be an ODD inconsistency.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	If the error address is null, check the rDSLEQUIP relation. The key used is printed in "LOGICAL INDEX". If the error address is not null, check the rIISMPATH relation. The key used is printed in "BAD DATA".
Dump Description:	
<i>Error Address</i> =	NA if the failed read is rDSLEQUIP. If it is rIISMPATH, the port status supplementary information (rPORTLA.suppl info).
<i>Bad Data</i> =	If the failed read is rDSLEQUIP, DSL GROUP. If it is rIISMPATH, the index to the port (rFC PORTTYP.port indx).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name (rPORTLA.port).
<i>Dump</i> =	The contents of the rPORTLA tuple.

Table 24 Error Code: DLI_FE

Error Code:	DLI_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the DLI status or NLI status on the major ONTC side. The inter-SM port indicated out-of-service family of equipment but the DLI or all NLIs on the major ONTC side is not out-of-service.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data</i> =	The DLI status in data delivery.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 25 Error Code: DLOSTTUP

Error Code:	DLOSTTUP
Error Description:	The mate D-port tuple does not exist for this rIORTLA. A major problem but recovery will be completed in the PORTTYP audit.
Possible Error Effect:	Unable to complete packet switching calls.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	portkey of packet rIORTLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 26 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	An rIDPB does not exist for the packet switching port for this analog modem pool port.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate (DPB/PORTTYP) audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of packet switch rIORTLA tuple.
<i>Bad Data</i> =	Key of analog rIORTLA.
<i>Good Data</i> =	Key of rIDPB.
<i>Logical Index</i> =	Key of packet switch rIORTLA tuple.
<i>Dump</i> =	Packet switch rIORTLA tuple.

Table 27 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The rIDSLEQUIP tuple read failed.
Possible Error Effect:	The rIORTLA tuple has been garbaged, which caused the reading of the wrong rIDSLEQUIP tuple or the rIDSLEQUIP tuple has been garbaged which would give an incorrect state for the rIDSLEQUIP tuple.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	Check ODD to determine if the port is really a member of a DSL. If it is not, verify that the lp_type and class of service fields in the rIORTLA relation are correct, and do not indicate that the port belongs to a DSL. If none of these errors is found, and the other audits do not clean up the problem, seek help.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 28 Error Code: DSLGROUP

Error Code:	DSLGROUP
Error Description:	The packet switching port list does not match the status of its DSL group. There are two possible errors: (1) The port is not OOS-FE, but the DSL group is OOS. (2) The port is OOS-FE, but the DSL group is not OOS.
Possible Error Effect:	Either the port will be used for a call which cannot complete because the DSL group is out-of-service or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	DSL group.
<i>Good Data</i> =	Logical port type.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIORMEST tuple for the PH (if available).
<i>Dump 3</i> =	rIORMEST tuple for the PSU (if available).

Table 29 Error Code: DS0B_LINK

Error Code:	DS0B_LINK
Error Description:	The field rIDS0B.use was set to PCD7SEC but the rIDS0B.portname field was not equal to rIORTLA.port field.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIORTLA.port field.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 30 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	The audit tried to read the rIDS0B tuple for a given rIORTLA.ds0b field but the tuple was missing from the database.
Possible Error Effect:	The call processing could not use that DS0B time slot.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the rIDS0B tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	First rINAILUP tuple.
<i>Dump 2</i> =	Second rINAILUP tuple.

Table 31 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rIORTLA exists for a port whose service class indicates that a rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 32 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The grid group member number derived from the rIGSDB tuple does not agree with the equivalent number (grpname) from the rIORTLA tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The grpname in the rIORTLA tuple will be set to the group member number derived from the rIGSDB tuple in an attempt to recover consistency.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIORTLA unit type.
<i>Bad Data =</i>	rIORTLA tuple grpname.
<i>Good Data =</i>	grid group member from rIGSDB.
<i>Logical Index =</i>	Logical key of the rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 33 Error Code: GPORT

Error Code:	GPORT
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but the monitoring bit in the rIORTLA tuple is not set.
Possible Error Effect:	Calls on this port may not be monitored.
Corrective Action Taken:	The monitoring bit is set to indicate that monitoring is active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the TMbitable[].
<i>Bad Data =</i>	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data =</i>	Test position involved in monitor B&I.
<i>Logical Index =</i>	Port name involved in monitor B&I.
<i>Dump =</i>	rIORTLA tuple involved in monitor B&I.

Table 34 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The rIPRISTAT tuple cannot be found from the rIORTLA tuple.
Possible Error Effect:	The B-port cannot be checked against the D-port.
Corrective Action Taken:	Report the error and return; no corrective action taken.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Key of rIORTLA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The logical port type.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 35 Error Code: INV

Error Code:	INV
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the

	corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data</i> =	Value found in that address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The Digital Carrier Line Unit number which is used to index the array AUs96inhp[].
<i>Dump</i> =	The whole array AUs96inhp[].

Table 36 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The status of the inter-SM nailup in the rIORTLA is not valid.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status supplementary information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 37 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	This rIORTLA is linked to an invalid list based on the logical port type.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA.
<i>Dump</i> =	rIORTLA tuple.

Table 38 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The rIORTLA tuple is linked to an invalid list. The tuple is associated with a D-channel, and the logical port type indicates remote protocol test access (rIFC_PORTTYP.lp_type == PCRDTA_D or PCRDTA_UD), but the port is on the level 2 or level 3 error port status list (rIORTLA.list == TMLTLV2ERR or TMLTLV3ERR). This is an illegal condition.
Possible Error Effect:	Remote digital test access (RDTA) actions and operations can not take place.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIORTLA tuple.
<i>Bad Data</i> =	The port status list (rIORTLA.list) which was found.
<i>Good Data</i> =	The logical port type in the corresponding rIFC_PORTTYP tuple (rIFC_PORTTYP.lp_type where rIFC_PORTTYP.port.member = (key) rIORTLA.port).
<i>Logical Index</i> =	The key of the rIORTLA tuple (port).

<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 39 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The T316 index is set (rIORTLA.t316_idx != TMTNULLIDX) but the T316 timer is not active.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	Clear the T316 index in the rIORTLA tuple (rIORTLA.t316_idx = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The T316 index of the rIORTLA tuple (rIORTLA.t316_idx).
<i>Good Data</i> =	The return value from the TM function that checks for a timer.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 40 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	An entry in TMbitable[] is not empty but the module attribute in bigport is invalid and indicates garbage in the table.
Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].
<i>Bad Data</i> =	Module in TMbitable[].bigport.module.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	None.

Table 41 Error Code: ISDNTUP

Error Code:	ISDNTUP
Error Description:	Circuits have been found in the rICKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIACT_SPR tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIACT_SPR tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIACT_SPR tuple.
<i>Dump</i> =	NA.

Table 42 Error Code: LIST

Error Code:	LIST
Error Description:	The list to which the portla should be linked does not agree with the list field used by the call processing and terminal maintenance programs.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes in error.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.

<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 43 Error Code: LNK_FE

Error Code:	LNK_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMS or NCT link status.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA tuple is recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data</i> =	The TMS link status in data delivery.
<i>Logical Index</i> =	Logical key to rIORTLA (port name).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 44 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The rCKTDATA tuple is missing from the database. The key used to read the tuple was derived from the path1_ds0.ds1name in the rNAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rNAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rCKTDATA tuple.
<i>Dump 1</i> =	First rNAILUP tuple
<i>Dump 2</i> =	Second rNAILUP tuple

Table 45 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	An rIORTLA tuple is missing from the database. The port used to read the tuple was taken from the rIORTLA.port1 or rIORTLA.port3_member field in the rNAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that port.
Corrective Action Taken:	The hairpin or nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the missing rIORTLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	First rNAILUP tuple.
<i>Dump 2</i> =	Second rNAILUP tuple.

Table 46 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The packet switching modem pool port does not have a mate rTEDB or rTEILA tuple.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA.

<i>Bad Data</i> =	Key to missing tuple (logical port - rITEDB or party - rTEILA).
<i>Good Data</i> =	Key of missing tuple (rITEDB or rTEILA).
<i>Logical Index</i> =	Key of tuple with missing link.
<i>Dump</i> =	rPORTLA tuple (if missing rITEDB) or rITEDB tuple (if missing rTEILA).

Table 47 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The status of the inter-SM nailup in the portla does not agree with the "pstat" field in the TMdlt_data[] array.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 48 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The status of the inter-SM nailup in the portla does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	"master" field in TMdlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 49 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	PORTLA audit is attempting to read a nonexistent tuple. There is a bad linkage between rPORTLA and rTKGSILC.
Possible Error Effect:	None. Call processing could be affected.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 50 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	The rPORTLA is on an invalid list.
Possible Error Effect:	The port may be lost for call processing.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPORTLA tuple.
<i>Bad Data</i> =	rPORTLA owner.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIPORTLA tuple.

Table 51 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	The rIPORTLA.ds0b field is a none zero value and this implies that the rIDS0B.use field should be set to PCDHP or PCD7SEC. It was found to be set to a different value.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIDS0B.use field.
<i>Logical Index</i> =	Key of rIPORTLA tuple.
<i>Dump 1</i> =	rIPORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 52 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A holding circuit was found to have fields in the rIPORTLA tuple that were not null.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIPORTLA tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIPORTLA tuple.
<i>Dump</i> =	rIPORTLA tuple.

Table 53 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but no rIPORTLA tuple with the same port key can be found.
Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].
<i>Bad Data</i> =	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	None.

Table 54 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	This is an OSPS B-port that is in-service idle.
Possible Error Effect:	OSPS calls cannot be processed to this BST.
Corrective Action Taken:	The rIPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 55 Error Code: NOTNU

Error Code:	NOTNU
Error Description:	An rIORTLA tuple was found to be used for hairpin or nailup connection but the rIORTLA.nailup field in it was not set to DBYES.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple will be placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 56 Error Code: ODD_DATA (1)

Error Code:	ODD_DATA (1)
Error Description:	Copies of ODD for this port (hunt and porttype) kept in the rIORTLA are corrupted.
Possible Error Effect:	Hunt bit incorrectly set to "hunted" - IDLPRT audit will attempt to idle a non-hunted port in AM routing data. Hunt bit incorrectly set to "non-hunted" - IDLPRT audit will not insure AM routing data for port agrees with module busy/idle status of port porttype bit set to wrong porttype.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Not used (see dump of rIORTLA).
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 57 Error Code: ODD_DATA (2)

Error Code:	ODD_DATA (2)
Error Description:	The circuit information (unit, unit type, circuit type, ckt_idx) contained in the rIORTLA tuple does not agree with the circuit information contained in the rICKTDATA tuple. The key to the rICKTDATA tuple is derived from the rIFC PORTTYP tuple.
Possible Error Effect:	The circuit cannot be used by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rICKTDATA tuple.

Table 58 Error Code: ODD_DATA (3)

Error Code:	ODD_DATA (3)
Error Description:	The CCS7 information (protocol, cic, pci) contained in the rIORTLA tuple does not agree with the information contained in the rIFC TRUNK tuple.
Possible Error Effect:	Incorrect port information.

Corrective Action Taken:	The information is updated and the rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	The value of cic in rIORTLA tuple.
<i>Good Data =</i>	The value of pci in rIORTLA tuple.
<i>Logical Index =</i>	Logical key for the rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIFC_TRUNK tuple.

Table 59 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	There is a mismatch between static data relations for this rIORTLA tuple.
Possible Error Effect:	There is no way for the switch to figure out if the port should have this service.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	The craft should determine which relation is missing and insert it using ODBE. If the port is really supposed to be nailed up, the craft must populate the rINAILUP tuple for this port using ODBE. If the port should not be nailed up, the craft should use ODBE to clear the nail-up data from the rIPRTTYP tuple for this port. If this is a modem pool port, the missing relations should be inserted for this rIORTLA. If the rIFC_PORTTYP is dumped, then there is a missing rIFC_TRUNK tuple which needs to be inserted.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Relation ID for the missing tuple. • rIEQUIPDSL. • rINAILUP. • rIORT_GROUP. • rIPSGP_PRT. • rIPSX25LN. Otherwise, the servclass field in rIFC_PORTTYP.
<i>Good Data =</i>	NA. If missing rIFC_TRUNK, then this is lp_type in rIFC_PORTTYP.
<i>Logical Index =</i>	Logical key of rIORTLA tuple (port name).
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIFC_PORTTYP tuple if missing rIFC_TRUNK.

Table 60 Error Code: OOS_MATE

Error Code:	OOS_MATE
Error Description:	An OSPS Autoquote DSL incorrectly has a different basic port status from its associated mate analog line.
Possible Error Effect:	If the Autoquote DSL is incorrectly in-service, it will be hunted but it will not be possible to get control of the mate analog line. Autoquote call failures will result whenever this pair is hunted. If the Autoquote DSL is incorrectly out-of-service, the pair will never be hunted and therefore will never be used for Autoquote calls.
Corrective Action Taken:	The Autoquote DSL is recovered to the same basic state as its mate analog line.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the Autoquote DSL rIORTLA tuple.
<i>Bad Data =</i>	Port status basic state for the Autoquote DSL.
<i>Good Data =</i>	Port status basic state for the Autoquote mate analog line.
<i>Logical Index =</i>	Port name of the Autoquote DSL.

<i>Dump 1 =</i>	rIPORTLA tuple for the Autoquote DSL.
<i>Dump 2 =</i>	rIPORTLA tuple for the Autoquote mate analog line.

Table 61 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	This rIPORTLA is linked to an invalid list.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPORTLA tuple.
<i>Bad Data =</i>	The rIPORTLA owner linkage proc cplr.st headk.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key (port name) of the rIPORTLA tuple.
<i>Dump =</i>	The contents of the rIPORTLA tuple.

Table 62 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Something is wrong with the logical key of the owner of the rIPORTLA tuple.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPORTLA tuple.
<i>Bad Data =</i>	The port list of the rIPORTLA tuple (rIPORTLA.list).
<i>Good Data =</i>	The owner link of the rIPORTLA tuple (rIPORTLA.prlist.st_ownk - DBSFRTLST).
<i>Logical Index =</i>	Key of rIPORTLA tuple.
<i>Dump =</i>	The rIPORTLA tuple.

Table 63 Error Code: PARENT

Error Code:	PARENT
Error Description:	The PORTLA audit is attempting to determine the facility circuit associated with the port. However, the SMIM conversion macro has returned a value of Null Circuit.
Possible Error Effect:	The port may be unusable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return value from SMIM function.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key to the rISMEST tuple.
<i>Dump =</i>	The rICKTDATA tuple.

Table 64 Error Code: P_UNKSTAT1

Error Code:	P_UNKSTAT1
Error Description:	The status of the inter-SM nailup in the rIPORTLA does not agree with the port status contained by the Terminal Process system.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rIPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPORTLA tuple.
<i>Bad Data =</i>	Port status.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIPORTLA.

<i>Dump</i> =	rIORTLA tuple.

Table 65 Error Code: PHNOOS

Error Code:	PHNOOS
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the PH status.
Possible Error Effect:	Call processing could be affected because the port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	PH status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 66 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the rICKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIPICB tuple is not in the database.
Possible Error Effect:	Manual asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIPICB tuple.
<i>Dump</i> =	NA.

Table 67 Error Code: PORT_TM

Error Code:	PORT_TM
Error Description:	The terminal maintenance list fields do not agree with the rIORTLA list fields.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The port is placed on the to-be-idled for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	rIORTLA owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 68 Error Code: PORT

Error Code:	PORT
Error Description:	The PORTLA audit attempted to convert a port name to a circuit name, but the PC utility macro returned a null value indicating that the circuit is not equipped.
Possible Error Effect:	The Port will not be usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA (0)
<i>Good Data</i> =	Circuit name.

<i>Logical Index</i> =	The key to the rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 69 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	An attempt to convert a port name to a port index has resulted in an indication that the port does not exist (null index).
Possible Error Effect:	The port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Port name.
<i>Good Data</i> =	NA (0)
<i>Logical Index</i> =	Port index.
<i>Dump</i> =	NA.

Table 70 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The port status in the rIORTLA implies the port should be on a different list than it is actually on.
Possible Error Effect:	Attempts to change a port's mtce status will result in asserts.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	If port had been manually removed from service, it must be removed from service again.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	rIORTLA list.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 71 Error Code: SHELF_SLC

Error Code:	SHELF_SLC
Error Description:	An attempt to determine the DCLU unit number from the S96Shelf circuit name has resulted in a return value of null unit.
Possible Error Effect:	The TAU line hardware path will not be torn down and re-established. The TAU line to TSI path will remain connected.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA (0)
<i>Good Data</i> =	Shelf name.
<i>Logical Index</i> =	Return value from SMIM function.
<i>Dump</i> =	NA.

Table 72 Error Code: SHELFREAD

Error Code:	SHELFREAD
Error Description:	Cannot read rIS96SHELF tuple.
Possible Error Effect:	No checking can be done on this port.
Corrective Action Taken:	None.
Manual Action Required:	Check ODD information for this relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	RLS96SHELF relation ID.
<i>Logical Index</i> =	Bank name.
<i>Dump</i> =	NA.

Table 73 Error Code: SLAVE_LINK

Error Code:	SLAVE_LINK
Error Description:	The status of the inter-SM nailup in the rIPORTLA does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	"master" field in TMdlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIPORTLA tuple.

Table 74 Error Code: SLC_PORT

Error Code:	SLC_PORT
Error Description:	The PORTLA audit attempted to convert a port name into the corresponding shelf or bank name. The return from the PC function indicated that the bank or shelf does not exist.
Possible Error Effect:	The port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Port name.
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Return value from the PC function.
<i>Dump</i> =	NA.

Table 75 Error Code: SLCLOSTTUP

Error Code:	SLCLOSTTUP
Error Description:	Cannot read the rIS96T1USG tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Skip auditing this tuple and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS96T1USG tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	rIS96T1USG tuple relation ID.
<i>Logical Index</i> =	Bank name.
<i>Dump</i> =	NA.

Table 76 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for this rIPORTLA cannot be found.
Possible Error Effect:	This rIPORTLA cannot be checked for correct FE state.
Corrective Action Taken:	Related audits will recover this port if required.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA (0).
<i>Bad Data</i> =	NA (0).
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIPORTLA tuple.

Table 77 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the

	linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Contents of state.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 78 Error Code: STATE_IDL

Error Code:	STATE IDL
Error Description:	The state field indicates idle linkage, but the linkage does not indicate that the rIORTLA is linked to its owner (as indicated by the contents of the list field).
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Contents of owner link.
<i>Good Data =</i>	Contents of list.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 79 Error Code: STATE_LNK

Error Code:	STATE LNK
Error Description:	The state field indicates that linkage exists but the link fields are empty.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Contents of the bad link field.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 80 Error Code: STATE_VAR

Error Code:	STATE VAR
Error Description:	A rIORTLA tuple was found linked to an unknown service list. The rIORTLA.list field must be marked as one of the legal service lists (either TMIS for in-service, or one a variety of possible out of service lists) as specified by DMLIST. This error may occur if a new service list has been added by terminal maintenance, but the PORTLA audit was never modified to recognize it.
Possible Error Effect:	The port will be unavailable to call processing and terminal maintenance activities since the associated rIORTLA tuple is in an unknown state.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple.
<i>Bad Data =</i>	The rIORTLA.list field found by the audit.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIORTLA tuple (port name).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 81 Error Code: STATUS

Error Code:	STATUS
Error Description:	The port status supplementary information in the rIORTLA is invalid for the relationship between this SM and its mate (rISMPATH.mate_sm). If rIORTLA.suppl_info is TMDLI , TMSNLI , or TMTMSCONN , the two SMs must not be in an HSM-RSM relationship or an RSM-RSM relationship under the same HSM.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The port status supplementary information (rIORTLA.suppl_info).
<i>Bad Data =</i>	The relationship between the SM and its mate.
<i>Good Data =</i>	The mate SM number which is served by the inter-SM path (rISMPATH.mate_sm).
<i>Logical Index =</i>	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 82 Error Code: STUCK

Error Code:	STUCK
Error Description:	The multipoint analog and digital ports have list fields which indicate in-service but the CPE is out-of-service.
Possible Error Effect:	Multipoint modem pool ports are on an out-of-service CPE. If this CPE should be in-service, the CPE should be power cycled.
Corrective Action Taken:	The analog port is idled for recovery to valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of analog rIORTLA tuple.
<i>Bad Data =</i>	CPE state.
<i>Good Data =</i>	Analog rIORTLA list.
<i>Logical Index =</i>	Key of analog rIORTLA tuple.
<i>Dump 1 =</i>	Analog rIORTLA tuple.
<i>Dump 2 =</i>	Digital rIORTLA tuple.
<i>Dump 3 =</i>	rTEILA tuple.

Table 83 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	The analog modem pool port is in service but the corresponding digital one is not.
Possible Error Effect:	Modem pool call could be failing.
Corrective Action Taken:	Both ports are recovered by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	List the analog port is on.
<i>Good Data =</i>	List the digital port is on.
<i>Logical Index =</i>	Analog port name.
<i>Dump 1 =</i>	Analog rIORTLA tuple.
<i>Dump 2 =</i>	The corresponding digital rIORTLA tuple.

Table 84 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The D-port is on the TMLTWAIT list but the T321 index is not set (rIPRISTAT.t321_idx = TMTNULLIDX).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The list from rIORTLA tuple (rIORTLA.list).
<i>Bad Data</i> =	The T321 index from the rIPRISTAT tuple (rIPRISTAT.t321_idx).
<i>Good Data</i> =	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	The rIORTLA tuple.
<i>Dump 2</i> =	The rIPRISTAT tuple.

Table 85 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	When the state of the rIORTLA tuple is DBPSIDLE, the wait for restoration bit should be cleared (rIORTLA.rstawait = DBNO).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rstawait attribute is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The waiting for service acknowledge message flag of the rIORTLA tuple (rIORTLA.srvawait).
<i>Bad Data</i> =	The waiting for restoration flag of the rIORTLA tuple (rIORTLA.rstawait).
<i>Good Data</i> =	The logical port type.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 86 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The B-port has been in a transient state for too long (longer than audit cycle).
Possible Error Effect:	This tuple is in a stuck OOS state and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	port_owner.
<i>Good Data</i> =	d_owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIORTLA tuple for D-port.

Table 87 Error Code: TRUNK_LINE

Error Code:	TRUNK_LINE
Error Description:	The port type for this port claims that it is a trunk but the rICKTDATA tuple that should exist for the trunk is missing.
Possible Error Effect:	There is no way for the audit to recover this port without the rICKTDATA information.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Determine whether the port is supposed to be a trunk. If it is, the add the rICKTDATA tuple using ODBE, or delete the rPORTTYP tuple. If a line, the rPORTTYP tuple needs to be updated to indicate that the port is a line.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Service class of rIORTLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 88 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The logical key of the owner of the rIORTLA tuple points to an unassigned port, but the service class is not unassigned (rIFC_PORTTYP.servclass != RC_PORTUNA).
Possible Error Effect:	The port will remain unusable by call processing.

Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The owner link of the rIORTLA tuple (rIORTLA.ptl1st.st_ownk - DBSFPTLST).
<i>Good Data</i> =	The service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 89 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	There are two possible errors: (1) A CCS trunk is found on a CCSINIT or a trunk maintenance list for non-CCS trunks. (2) A non-CCS trunk is found on the CCSINIT or the trunk maintenance continuity list for only CCS trunks.
Possible Error Effect:	There are two possible error effects: (1) The routing and terminal allocation subsystem won't hunt this trunk. Trunk maintenance may have difficulty recovering this trunk. (2) Any continuity testing done on a non-CCS trunk will fail.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status. When the CCS trunk is put on the to-be-idled list, a reset message is sent to the far-end office.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Contents of list.
<i>Good Data</i> =	Contents of owner link.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

PORTLA-B

Software Release: 5E12 only

1. PORT LINKAGE AREA (PORTLA) AUDIT

Each port has a port linkage area tuple that describes the status and associations of the port. These port linkage areas contain information such as the owner of the port, processes that are waiting to access the port, the status of the port (precut or postcut, and so forth), and a preemption flag that shows whether the port is being preempted.

The port linkage area audit (PORTLA) verifies the data contained in the port linkage areas. When the PORTLA audit is invoked, it checks all ports (both lines and trunks).

2. DATA STRUCTURES

The data structure checked by the PORTLA audit is the port linkage area relation (RLPORTLA). The RLPORTLA relation has one-to-one linkages with the pending history relation (RLPHISTORY) and the process control block linkage area relation (RLPCBLA). It also has owner-member linkage with the coupler relation (RLCPLR) and member linkage with the port list relation (RLPRTLST). The RLPRTLST relation has 16 head cells, each of which owns a list of port linkage areas. Each list indicates a particular port status (for example, in service, out of service, to be idled, and so forth). The port linkage area list item contains an ID that indicates the list to which the port linkage area belongs. The PORTLA audit uses the feature control port name to service class relation (RLFC_PORTTYP) to perform an existence check. Analog modem pool ports will verify the existence of the related packet switch rIORTLA tuple and its corresponding rIDPB tuple. Additional multipoint modem pool checks have been added for the following data structures:

- The terminal equipment data block relation (RLTEDB).
- The packet switching X.25 level 3 data for DSL's relation (RLPSX25LN).
- The terminal endpoint identifier linkage area relation (RLTEILA).

The PORTLA audit added checks of the following SLC[®] 96 and network management data structures:

- The trunk group selective incoming load control relation (RLTKGSILC).
- The switch maintenance equipment status table relation (RLSMEST).
- The circuit data block common area relation (RLCDBCOM).
- The SLC[®] 96 equipage relation (RLS96EQUIP).
- The SLC[®] 96 T1 usage relation (RLS96T1USG).

Figure 1 shows these data structures.

The PORTLA audit insures the integrity of the port linkage area relation (RLPORTLA). A field in the rIORTLA tuple checks for datalink trunks and lines. If the port is of a particular service class, the datalink bit should be set. If it is not, this check prints an error and then sets the bit. It then continues processing and it does not recover the tuple.

3. AUDIT CHECKS

The PORTLA audit uses the static driver (DRSTATIC) to step through the RLPORTLA relation.

- (a) Initially, the audit calls a dynamic to static existence check function to perform the following checks.

- (1) This existence check verifies that, for every dynamic rIORTLA tuple, a static port name to service class tuple (rLFC_PORTTYP) exists. If the static rLFC_PORTTYP tuple does not exist, the extra rIORTLA tuple is deleted (EXTRATUP).
 - (2) For each tuple in RLPORTLA that is associated with a circuit, the existence check verifies that a corresponding tuple exists in the static circuit data relation (RLCKTDATA) for those service classes that should have a tuple (TRUNK_LINE).
- (b) The audit calls a semantic check function to perform the following checks.
- (1) The audit checks to see that the rIORTLST owning head cell of the port linkage area, as determined from the port linkage area's owner linkage item, agrees with the list item (LIST).
 - (2) The audit checks the cutover status of the office and port. If the office is in a cutover state, the audit checks that the rIORTLA tuple is on the correct list (CUTOVER).
 - (3) The port status (pstatus) item contains the primary maintenance status of the port. The audit verifies that the list to which the port linkage area belongs is consistent with the pstatus item (PSTATUS).
 - (4) The audit verifies that the hunt item in the RLPORTLA relation is consistent with the hunt item in the RLFC_PORTTYP relation (ODD_DATA).
 - (5) The audit verifies that the port type (ptype) item in the RLPORTLA relation is consistent with the port type (porttype) item in the RLFC_PORTTYP relation (ODD_DATA).
 - (6) The audit verifies the datalink item with valid service class (BLINK).
 - (7) If the port is a digital carrier port, the audit verifies that the list to which the port linkage area belongs is consistent with the state of the digital facility interface (DFI) circuit. An active red or yellow alarm should correspond to ports on the facility failure list. Family-of-equipment and to-be-idled lists are also allowed because they are higher priority lists. If both the red and yellow alarms are inactive, the port linkage area can be on any list except the facility failure list (ALARM).
 - (8) Common channel signaling ports are verified to be on valid lists by this audit (WRONGLIST).
 - (9) If the rIORTLA tuple is on a network management list, the corresponding rITKGSILC tuple is read (MISSTUP). Several fields, including the port and quarantine state fields, are checked (PORT_TM).
 - (10) The audit verifies that OSPS Autoquote DSL's have a mate analog rIORTLA tuple (OOS_MATE).
 - (11) The audit verifies that nailup ports have a mate rINAILUP tuple.
 - (12) During linkage checking of the rIORTLA tuple the function AUptlchk() performs point-to point-back checks between the rIORTLA and rIADB relations. If an error is found, an error report is printed to the read-only printer (ROP) and recovery is started.
 - (13) PORTLA audits the linkage between the rIORTLA relation and the RLADB relation for shared directory number (DN) analog. When the PORTLA audit detects the error, the rIORTLA tuple is linked to the rIADB tuple and the PORTLA audit checks to make sure the sh_analog bit in the RLFC_PORTTYP relation is set to DBYES. If sh_analog bit is not set, the PORTLA audit prints an error on the ROP and returns AUFALSE to start recovery (BDLINK).
 - (14) A port linkage area may be linked to one process control block linkage area directly or to one or more process control block linkage areas by means of couplers. Also, it may not be linked to any process control block linkage areas. The port linkage area state item indicates which of these two states exists.

The audit checks to see that the state item agrees with the actual linkage (STATE_IDL or STATE_LNK or STATE).

- (15) In a remote switching module (RSM) environment, if a port is idle, hunted, and in service, the routing and terminal allocation (RTA) subsystem is called to mark the port idle in the hunted port status relations.
 - (16) If the logical port type field of rIFC_PORTTYP indicates that this is a SLC[®] 96 special line, the RLS96T1USG relation is checked to ensure that one dedicated time assignment unit (TAU) is associated with the port (BADTSCOUNT). If it is determined that one does not exist (or more than one exists), the current hardware connection is torn down and a new one is established (SLCLOSTTUP). If an rIS96SHELF tuple is not found, the error is reported (SHELFREAD).
 - (17) For SLC[®] 96 ports, the audit determines whether the port is associated with a 4AB1 or 4AC1 line unit or SLC[®] 96 digital carrier line unit (DCLU). If the port is associated with a DCLU, the key of the rIS96EQUIP tuple is calculated and the tuple is read. The bank status is checked to determine the grid identification, the associated rSMEST tuple is read (SMEST), and the port is conditioned.
 - (18) There is a list of precut status ports contained in the switching module (SM) protected memory. The audit calls a function to determine whether the port associated with the port linkage area being checked is on this list. If it is, the audit checks whether the port linkage area is linked to the rIPRTLST head cell that owns precut status port linkage areas.
 - (19) If the port is an ISDN port, the audit checks the parent circuit(s). It reports an error if the port's maintenance state is inconsistent with the maintenance state of the parent circuit. The audit also checks the validity of family-of-equipment states.
 - (20) This audit also has special checks for modem pool ports. If the port is a modem pool port, the analog and corresponding packet switch ports will be synchronized. For multipoint modem pool ports, the CPE status will be synchronized.
- (c) The PORTLA audit then calls a function to maintain consistency between the monitor busy & idle (B&I) data table, TMbitable[] and the rIPORTLA tuple. The TMbitable[] contains the port and TM process ID information for any ports involved in monitoring using the monitor (B&I) feature.

4. ERROR HANDLING

For all errors except those encountered when the audit is running in the initialization mode, the port linkage area is placed on the to-be-idled list (this list is owned by an rIPRTLST head cell). At the same time, a terminal maintenance process is activated. This process checks the hardware state and the contents of the switch maintenance equipment status table relation (RLSMEST) to determine what the status should be for a given port. For circuits marked out of service (OOS) in rSMEST, the associated ports are given an OOS, family of equipment (FE) status. DFI ports with active red or yellow carrier alarms are given an OOS, facility failure (FAF) status. Ports whose scanning circuitry indicate that they are off-hook are given an OOS, high and wet (HW) status. All other ports are given an in-service status.

For errors encountered when the audit is running in the initialization mode, the port linkage area is linked to the in-service list except for the ports as determined by the service class. Common channel signaling (CCS) trunks are linked to the disabled list and recovered by terminal maintenance (TM) subsystem. D-channels, inter-switch trunks, ISM nailups, X.75 and X75' trunks, and packet switching B-channels are linked to the DSL initialization list and recovered by the TM subsystem. B-channels are linked to the D-channel out-of-service list if the D-channel is out-of-service, else they are linked to the in-service list. In all cases except during initializations, errors are reported by means of output messages.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALARM

Error Code:	ALARM
Error Description:	The list the rIORTLA is on is inconsistent with the state of the digital carrier alarm. Digital carrier ports only.
Possible Error Effect:	If carrier alarm is ON, the port is incorrectly in service and if used by call processing, would cause call failures. If carrier alarm is OFF, the port is incorrectly out of service and thus unavailable for call processing use.
Corrective Action Taken:	The port will be placed on the to-be-idled list for later idling. The software maintenance status of the port will be made consistent with the hardware (carrier) alarm state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	List owner for rIORTLA tuple.
<i>Good Data =</i>	Carrier group alarm status.
<i>Logical Index =</i>	Key of the rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIORMEST tuple.

Table 2 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rIORMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rIORMEST tuple.

Table 3 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	An rIORTLA exists for a port whose service class indicates that an rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Service class of the port.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 4 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by

	recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 5 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	An rSMEST tuple for the circuit under audit or the parent/mate circuit for this circuit was found to be in an invalid state with respect to the information in the MRA deferred work recovery queue. The qual_2 state of SMAUOOS is used to indicate when audits or SI has recovered a circuit, parent or mate to the MRA queue. Note, refer to <i>Good Data</i> for details.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly by MRA.
Corrective Action Taken:	The audit will attempt to recovery the circuit using the MRA queue and/or leave it in a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	qual_2 field in the SMEST tuple.
<i>Good Data</i> =	Return value from AUMRARET (au/hdr/AUglobckt.L).
	0 = Circuit was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	2 = Parent/mate was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	4 = Circuit is marked AUOOS but not in queue, parent is not in queue, circuit/parent/mate is not MRA BUSY.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	parent/mate rSMEST tuple.

Table 6 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	An rDS0B tuple is involved in a hairpin connection but the rPORTLA.dsob field does not have the correct value.
Possible Error Effect:	Call processing will not be able to use that rPORTLA tuple.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	rDS0B.use field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The DS0B key.
<i>Dump 1</i> =	rDS0B tuple.
<i>Dump 2</i> =	rPORTLA tuple.

Table 7 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The rDS0B tuple is involved in a hairpin or nailup connection but the rDS0B.portname field tuple is not matching the rNAILUP.port1 field.
Possible Error Effect:	There will be no hairpin or nailup connection established since a wrong port is

	being used.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDS0B tuple.
<i>Bad Data =</i>	The rIDS0B tuple key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIDS0B tuple.
<i>Dump 1 =</i>	rIDS0B tuple.
<i>Dump 2 =</i>	rIORTLA tuple.

Table 8 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	An rIORTLA tuple contains the key of an rIGPFTIMER tuple in one of it's timer key fields but the rIGPFTIMER tuple is either not linked back (rIGPFTIMER.udata contains a different relation ID or key from the owning tuple) or the timer type (rIGPFTIMER.timer_type) does not have the expected value for the timer key field being used.
Possible Error Effect:	Varies according to conditions including the rIORTLA usage and which timer key field is being checked.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data =</i>	The relation ID stored in rIGPFTIMER.udata.
<i>Good Data =</i>	The tuple key stored in rIGPFTIMER.udata.
<i>Logical Index =</i>	The rIORTLA tuple logical key (rIORTLA.port).
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIGPFTIMER tuple.

Table 9 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	For CCS7 trunks, there are four possible conditions for which this error is reported. (1) There is a reset in progress, but the <i>reset_inprog</i> bit in PORTLA is not set to DBYES. (2) There is a reset in progress with the <i>reset_inprog</i> bit in PORTLA set to DBYES, but the port is not owned. (3) There is no reset in progress, but the <i>reset_inprog</i> bit in PORTLA is set to DBYES. (4) There is no entry for this port, but the <i>reset_inprog</i> bit in PORTLA is set to DBYES. For CCS6 trunks, the following fields in PORTLA are not set correctly. These fields include <i>protocol</i>, <i>cic</i>, <i>pci</i>, <i>reset_inprog</i>.
Possible Error Effect:	Incorrect port usage.
Corrective Action Taken:	If the problem is in regards to the <i>reset_inprog</i> bit, then the bit will be set to its correct value. If the error is in regards to the CCS6 trunk, then the following will occur. <i>protocol</i> will be set to DBCCIS6, <i>cic</i> will be set to CCNULCIC, <i>pci</i> will be set to CCNULPCI, and <i>reset_inprog</i> will be set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple for CCS6 trunks. Error flag value for CCS7

	trunks.
<i>Bad Data</i> =	Value of in_sigtype from rIFC_TRUNK for CCS6 trunks. Value of mtype returned from TM for CCS7 trunks.
<i>Good Data</i> =	Value of out_sigtype from rIFC_TRUNK for CCS6 trunks. Value of type returned from TM for CCS7 trunks.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 10 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	One of two possible errors: (1) The port is on the TBI (to be idled) list and is still waiting to be run but the tbi_flag is set, indicating that the job is running. (2) The port is not on the TBI list and the flag is set.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	Corresponds to the Error Description as follows: (1) The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status. (2) The tbi_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	When <i>Good Data</i> is 0, the to-be-idled flag of the rIORTLA tuple (rIORTLA.tbi_flag). When <i>Good Data</i> is 1, the return value from TMisqueued() function call.
<i>Good Data</i> =	0 or 1.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 11 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The D-port is linked to the unassigned list but the service class taken from static data does not indicate that the port should be on unassigned list.
Possible Error Effect:	The rIORTLA is unavailable for call processing.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to the proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Owner of rIORTLA tuple.
<i>Good Data</i> =	Service class of port.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 12 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	A SLC [®] 96 time slot was found to have more than one time assignment unit assigned to it.
Possible Error Effect:	Possible origination problems for some mode II SLC [®] 96 special lines.
Corrective Action Taken:	Reconstruct the rIS96T1USG tuple with one time assignment unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS96T1USG tuple.
<i>Bad Data</i> =	Count of T1TS lines for this time assignment unit.

<i>Good Data</i> =	Number of time slots allowed (1).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIS96T1USG tuple.

Table 13 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An rIDS0B tuple was used for hairpin or nailup connection but the rIDS0B.use field was not set to indicate that.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDS0B tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	rIDS0B tuple.
<i>Dump 2</i> =	rIPORTLA tuple.

Table 14 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The rIPORTLA tuple is on the wrong list. The port could be stuck on an out-of-service list when it should be in-service or on an is-service list when it can not be used because of the parent circuit. There are two possible errors: (1) The port is not OOS-FE, but the parents circuits are OOS. (2) The port is OOS-FE, but the parents circuits are not OOS. If the port is on a IDCU TR08 the audit will dump the shelf/bank that was checked. For analog ports on an IDCU TR303, one DS1 will be dumped but all DS1's on the RT303 will be checked.
Possible Error Effect:	Either the port will be used for a call which cannot complete because of out-of-service equipment or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	List owner of rIPORTLA tuple.
<i>Good Data</i> =	Logical port type.
<i>Logical Index</i> =	Logical key of rIPORTLA tuple.
<i>Dump 1</i> =	rIPORTLA tuple.
<i>Dump 2</i> =	rISMEST tuple with port as key.
<i>Dump 3</i> =	rISMEST tuple for PH if available.
<i>Dump 4</i> =	rISMEST tuple for PSU if available.

Table 15 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The rIPORTLA was found linked to the rIADB tuple when static data (rIFC_PORTTYP) did not have the shared analog bit set.
Possible Error Effect:	Call processing cannot access the port correctly.
Corrective Action Taken:	The rIPORTLA tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPORTLA tuple.
<i>Bad Data</i> =	Linkage field to the rIADB tuple.
<i>Good Data</i> =	Shared analog bit in rIFC_PORTTYP tuple.

<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 16 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The packet switching modem pool port is not bound (rITEDB tuple to rTEILA tuple).
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITEDB tuple.
<i>Bad Data</i> =	Key to rTEILA tuple (0 means no rTEILA tuple).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rITEDB tuple.
<i>Dump</i> =	rITEDB tuple.

Table 17 Error Code: BINDWRONG

Error Code:	BINDWRONG
Error Description:	The packet switching modem pool port is not bound to the correct rITEDB tuple.
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA tuple.
<i>Bad Data</i> =	TEDB key from rTEILA tuple.
<i>Good Data</i> =	Key of rITEDB tuple (logical port).
<i>Logical Index</i> =	Logical key of rTEILA tuple.
<i>Dump</i> =	rTEILA tuple.

Table 18 Error Code: BLINK

Error Code:	BLINK
Error Description:	The rIORTLA service class was of the datalink type but the datalink field was not set to DBYES.
Possible Error Effect:	The line or trunk would not be treated as a datalink for removes/restores.
Corrective Action Taken:	The datalink bit will be set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	datalink field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 19 Error Code: BTRANCK

Error Code:	BTRANCK
Error Description:	The associated D-port is out-of-service and this B-port is not on a list to indicate the unavailability of the D-port. This is considered to be valid for a short transient period. This error is not reported for at least 5 minutes.
Possible Error Effect:	Call processing cannot complete service on this port.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	D-port owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.

<i>Dump 2</i> =	D-channel rIORTLA tuple.
<i>Dump 3</i> =	rIPRISTAT tuple.
<i>Dump 4</i> =	when the <i>lp_type</i> is SVC, rIDPB tuple.

Table 20 Error Code: CADNPORT

Error Code:	CADNPORT
Error Description:	The port is either on the CADN list when the <i>cadn</i> field in FCPORTTYP is set to TMIIS or the port is not on the CADN list when the <i>cadn</i> field in FCPORTTYP is not TMIIS. For both of these cases, there error is only reported if the port is not owned.
Possible Error Effect:	The port may not be able to be used by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	<i>cadn</i> field in rIFC_PORTTYP tuple.
<i>Good Data</i> =	<i>list</i> field in rIORTLA tuple.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 21 Error Code: CKTDATA (1)

Error Code:	CKTDATA (1)
Error Description:	The PORTLA audit is attempting to read a nonexistent rICKTDATA tuple. The key to the rICKTDATA tuple is derived from the rIFC_PORTTYP tuple.
Possible Error Effect:	Call processing could be affected.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 22 Error Code: CKTDATA (2)

Error Code:	CKTDATA (2)
Error Description:	The rICKTDATA tuple is missing from the database. The key used to read the tuple was derived from the <i>path2_ds0.ds1name</i> in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rICKTDATA tuple.
<i>Dump 1</i> =	First rINAILUP tuple
<i>Dump 2</i> =	Second rINAILUP tuple

Table 23 Error Code: CUTOVER

Error Code:	CUTOVER
Error Description:	A port is on the wrong list based on the office's or this ports cutover status.
Possible Error Effect:	Pre-cut ports could incorrectly be receiving service from this office or post-cut (active) ports could be denied service.
Corrective Action Taken:	Port is conditioned and returned to correct list (maintenance status).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.

<i>Bad Data</i> =	List owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 24 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit failed a read of the RLDSLEQUIP or RLISMPATH relation associated with this RLPORTLA.
Possible Error Effect:	There could be an ODD inconsistency.
Corrective Action Taken:	The RLPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	If the error address is null, check the RLDSLEQUIP relation. If the error address is not null, check the RLISMPATH relation. The key used is printed in <i>Bad Data</i> .
Dump Description:	
<i>Error Address</i> =	NA if the failed read is rIDSLEQUIP. If it is rIISMPATH, the port status supplementary information (rIORTLA.suppl. info).
<i>Bad Data</i> =	If the failed read is rIDSLEQUIP, DSL GROUP. If it is rIISMPATH, the index to the port (rIFC. PORTTYP.port. indx).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name (rIORTLA.port).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 25 Error Code: DLI_FE

Error Code:	DLI_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the DLI status or NLI status on the major ONTC side. The inter-SM port indicated out-of-service family of equipment but the DLI or all NLIs on the major ONTC side is not out-of-service.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data</i> =	The DLI status in data delivery.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 26 Error Code: DLOSTTUP

Error Code:	DLOSTTUP
Error Description:	The mate D-port tuple does not exist for this rIORTLA. A major problem but recovery will be completed in the PORTTYP audit.
Possible Error Effect:	Unable to complete packet switching calls.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	portkey of packet rIORTLA tuple.
<i>Good Data</i> =	Logical port type (rIFC. PORTTYP.lp. type).
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 27 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	An rIDPB does not exist for the packet switching port for this analog modem pool port.

Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate (DPB/PORTTYP) audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of packet switch rIORTLA tuple.
<i>Bad Data =</i>	Key of analog rIORTLA.
<i>Good Data =</i>	Key of rIDPB.
<i>Logical Index =</i>	Key of packet switch rIORTLA tuple.
<i>Dump =</i>	Packet switch rIORTLA tuple.

Table 28 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The rIDSLEQUIP tuple read failed.
Possible Error Effect:	The rIORTLA tuple has been garbaged, which caused the reading of the wrong rIDSLEQUIP tuple or the rIDSLEQUIP tuple has been garbaged which would give an incorrect state for the rIDSLEQUIP tuple.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	Check ODD to determine if the port is really a member of a DSL. If it is not, verify that the Ip_type and class of service fields in the RLPORTLA relation are correct, and do not indicate that the port belongs to a DSL. If none of these errors is found, and the other audits do not clean up the problem, seek help.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 29 Error Code: DSLGROUP

Error Code:	DSLGROUP
Error Description:	The packet switching port list does not match the status of its DSL group. There are two possible errors: (1) The port is not OOS-FE, but the DSL group is OOS. (2) The port is OOS-FE, but the DSL group is not OOS.
Possible Error Effect:	Either the port will be used for a call which cannot complete because the DSL group is out-of-service or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	DSL group.
<i>Good Data =</i>	Logical port type.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rSMEST tuple for the PH (if available).
<i>Dump 3 =</i>	rSMEST tuple for the PSU (if available).

Table 30 Error Code: DS0B_LINK

Error Code:	DS0B_LINK
Error Description:	The field rIDS0B.use was set to PCD7SEC but the rIDS0B.portname field was not equal to rIORTLA.port field.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIORTLA.port field.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 31 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	The audit tried to read the rIDS0B tuple for a given rIORTLA.ds0b field but the tuple was missing from the database.
Possible Error Effect:	The call processing could not use that DS0B time slot.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the rIDS0B tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	First rINAILUP tuple.
<i>Dump 2</i> =	Second rINAILUP tuple.

Table 32 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rIORTLA exists for a port whose service class indicates that a rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 33 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The grid group member number derived from the rIGSDB tuple does not agree with the equivalent number (grpname) from the rIORTLA tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The grpname in the rIORTLA tuple will be set to the group member number derived from the rIGSDB tuple in an attempt to recover consistency.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIORTLA unit type.
<i>Bad Data</i> =	rIORTLA tuple grpname.
<i>Good Data</i> =	grid group member from rIGSDB.
<i>Logical Index</i> =	Logical key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 34 Error Code: GPORT

Error Code:	GPORT
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but the monitoring bit in the rIORTLA tuple is not set.
Possible Error Effect:	Calls on this port may not be monitored.
Corrective Action Taken:	The monitoring bit is set to indicate that monitoring is active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].

<i>Bad Data</i> =	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	rIPORTLA tuple involved in monitor B&I.

Table 35 Error Code: GRP2INFO

Error Code:	GRP2INFO
Error Description:	A port was found with an invalid trunk group number (<i>grpnum</i>).
Possible Error Effect:	This port may be unusable for call traffic.
Corrective Action Taken:	The <i>grpnum</i> field corrected the value as indicated in the corresponding FC_TRUNK tuple, or changed to a null value when no corresponding FC_TRUNK tuple could be found.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rIPORTLA tuple.
<i>Bad Data</i> =	Invalid group number (rIPORTLA.grpnum).
<i>Good Data</i> =	Good value of <i>grpnum</i> (rIFC_TRUNK.tgroup) or null value.
<i>Logical Index</i> =	Port name involved.
<i>Dump 1</i> =	rIPORTLA tuple involved.
<i>Dump 2</i> =	rIFC_TRUNK tuple involved if present.

Table 36 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The rIPRISTAT tuple cannot be found from the rIPORTLA tuple.
Possible Error Effect:	The B-port cannot be checked against the D-port.
Corrective Action Taken:	Report the error and return no corrective action taken.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Key of rIPORTLA tuple.
<i>Bad Data</i> =	Pointer to the rIPRISTAT tuple.
<i>Good Data</i> =	The logical port type.
<i>Logical Index</i> =	Key of rIPORTLA tuple.
<i>Dump</i> =	The rIPORTLA tuple.

Table 37 Error Code: INT_DATA_BAD

Error Code:	INT_DATA_BAD
Error Description:	The value of rIPORTLA.dlci is not as expected when <i>Good Data</i> is 1, the logical port type is a SVC trunk and it should match the value obtained when the PS macro is called when <i>Good Data</i> is 2, the value of rIPORTLA.dlci should be zero.
Possible Error Effect:	The port may be sending messages to the wrong frame relay.
Corrective Action Taken:	Reset the rIPORTLA.dlci as expected.
Manual Action Required:	None
Dump Description:	
<i>Error Address</i> =	When <i>Good Data</i> is 1, the expected value of the rIPORTLA.dlci when <i>Good Data</i> is 2, the logical port type (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	the rIPORTLA.dlci.
<i>Good Data</i> =	1 or 2.
<i>Logical Index</i> =	Key of rIPORTLA tuple (rIPORTLA.port).
<i>Dump</i> =	rIPORTLA tuple.

Table 38 Error Code: INV

Error Code:	INV
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.

Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data</i> =	Value found in that address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump</i> =	The whole array AUs96inhp[].

Table 39 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The status of the inter-SM nailup in the rIORTLA is not valid.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status supplementary information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 40 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	This rIORTLA is linked to an invalid list based on the logical port type.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA.
<i>Dump</i> =	rIORTLA tuple.

Table 41 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The rIORTLA tuple is linked to an invalid list. The tuple is associated with a D-channel, and the logical port type indicates remote protocol test access (rIFC_PORTTYP.lp_type == PCRDTA_D or PCRDTA_UD), but the port is on the level 2 or level 3 error port status list (rIORTLA.list == TMLTLV2ERR or TMLTLV3ERR). This is an illegal condition.
Possible Error Effect:	Remote digital test access (RDTA) actions and operations can not take place.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The port status list (rIORTLA.list) which was found.
<i>Bad Data</i> =	The key to the rIPRISTAT tuple (rIPRISTAT.prigrp)
<i>Good Data</i> =	Return code from validation of the PRI D-port status.
<i>Logical Index</i> =	The key of the rIORTLA tuple (port).
<i>Dump 1</i> =	The contents of the rIORTLA tuple.
<i>Dump 2</i> =	If this dump exists, it holds the contents of the D-channel's rIORTLA tuple.

Table 42 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	An entry in TMbtable[] is not empty but the module attribute in bigport is invalid and indicates garbage in the table.

Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the TMbitable[].
<i>Bad Data =</i>	Module in TMbitable[].bigport.module.
<i>Good Data =</i>	Test position involved in monitor B&I.
<i>Logical Index =</i>	Port name involved in monitor B&I.
<i>Dump =</i>	None.

Table 43 Error Code: ISDNTUP

Error Code:	ISDNTUP
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIACT_SPR tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIACT_SPR tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIACT_SPR tuple.
<i>Dump =</i>	NA.

Table 44 Error Code: LIST

Error Code:	LIST
Error Description:	The list to which the portla should be linked does not agree with the list field used by the call processing and terminal maintenance programs.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes in error.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	rIORTLA list.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 45 Error Code: LNK_FE

Error Code:	LNK_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMS or NCT link status.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA tuple is recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data =</i>	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data =</i>	The TMS link status in data delivery.
<i>Logical Index =</i>	Logical key to rIORTLA (port name).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 46 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The rCKTDATA tuple is missing from the database. The key used to read the

	tuple was derived from the path1 ds0.ds1name in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rICKTDATA tuple.
<i>Dump 1</i> =	First rINAILUP tuple
<i>Dump 2</i> =	Second rINAILUP tuple

Table 47 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	An rIORTLA tuple is missing from the database. The port used to read the tuple was taken from the rIORTLA.port1 or rIORTLA.port3_member field in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that port.
Corrective Action Taken:	The hairpin or nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the missing rIORTLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	First rINAILUP tuple.
<i>Dump 2</i> =	Second rINAILUP tuple.

Table 48 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The packet switching modem pool port does not have a mate rITEDB or rITEILA tuple.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA.
<i>Bad Data</i> =	Key to missing tuple (logical port - rITEDB or party - rITEILA).
<i>Good Data</i> =	Relation ID of missing tuple (rITEDB or rITEILA).
<i>Logical Index</i> =	Key of tuple with missing link.
<i>Dump</i> =	rIORTLA tuple (if missing rITEDB) or rITEDB tuple (if missing rITEILA).

Table 49 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The status of the inter-SM nailup in the portla does not agree with the "pstat" field in the TMDlt_data[] array.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 50 Error Code: MASTER_LINK

Error Code:	MASTER_LINK

Error Description:	The status of the inter-SM nailup in the portla does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	"master" field in TMdlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 51 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	PORTLA audit is attempting to read a nonexistent tuple. There is a bad linkage between rIORTLA and rTKGSILC.
Possible Error Effect:	None. Call processing could be affected.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 52 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	The rIORTLA is on an invalid list.
Possible Error Effect:	The port may be lost for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 53 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	An rIORTLA contains an invalid telephone status monitored value (rIORTLA.tsmcd). If the port is a trunk (rIFC_PORTTYPE.porttype = TM_TRK), either the port is marked as being monitored (rIORTLA.tsmcd != DBNO) or status is not idle (rIORTLA.tsm_stat != TSM_IDLE). If the port is a line, the monitored status is inconsistent with the ODD (no rIFC_LINE tuple exists and rIORTLA.tsmcd != DBNO or the rIFC_LINE tuple exists and rIORTLA.tsmcd != rIFC_LINE.tsmcd).
Possible Error Effect:	If the port is a trunk, asserts from telephone status monitoring code are likely. If the port is a line, telephone status monitoring may not work correctly for an end user.
Corrective Action Taken:	The audit corrects the value(s) in the rIORTLA tuple (rIORTLA.tsmcd = DBNO and rIORTLA.tsm_stat = TSM_IDLE for a trunk or rIORTLA.tsmcd = rIFC_LINE.tsmcd).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The port's telephone monitor status (rIORTLA.tsm_stat).
<i>Bad Data</i> =	The port's monitored status value (rIORTLA.tsmcd).

<i>Good Data</i> =	Value indicating whether the rIFC_LINE tuple exists. GLSUCCESS implies that it does. DBNO_MATCH implies that it does not.
<i>Logical Index</i> =	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump 1</i> =	The rIORTLA tuple.
<i>Dump 2</i> =	The rIFC_PORTTYPE tuple.

Table 54 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	The rIORTLA.ds0b field is a none zero value and this implies that the rIDS0B.use field should be set to PCDHP or PCD7SEC. It was found to be set to a different value.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIDS0B.use field.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 55 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A holding circuit was found to have fields in the rIORTLA tuple that were not null.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 56 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but no rIORTLA tuple with the same port key can be found.
Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].
<i>Bad Data</i> =	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	None.

Table 57 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	This is an OSPS B-port that is in-service idle.
Possible Error Effect:	OSPS calls cannot be processed to this BST.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Good Data</i> =	Service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 58 Error Code: NOTNU

Error Code:	NOTNU
Error Description:	An rIORTLA tuple was found to be used for hairpin or nailup connection but the rIORTLA.nailup field in it was not set to DBYES.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple will be placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 59 Error Code: ODD_DATA (1)

Error Code:	ODD_DATA (1)
Error Description:	Copies of ODD for this port (hunt and porttype) kept in the rIORTLA are corrupted.
Possible Error Effect:	Hunt bit incorrectly set to "hunted" - IDLPRT audit will attempt to idle a non-hunted port in AM routing data. Hunt bit incorrectly set to "non-hunted" - IDLPRT audit will not insure AM routing data for port agrees with module busy/idle status of port porttype bit set to wrong porttype.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Not used (see dump of rIORTLA).
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 60 Error Code: ODD_DATA (2)

Error Code:	ODD_DATA (2)
Error Description:	The circuit information (unit, unit type, circuit type, ckt_idx) contained in the rIORTLA tuple does not agree with the circuit information contained in the rCKTDATA tuple. The key to the rCKTDATA tuple is derived from the rIFC_PORTTYP tuple.
Possible Error Effect:	The circuit cannot be used by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rCKTDATA tuple.

Table 61 Error Code: ODD_DATA (3)

Error Code:	ODD_DATA (3)

Error Description:	The CCS7 information (protocol, cic, pci) contained in the rIORTLA tuple does not agree with the information contained in the rIFC_TRUNK tuple.
Possible Error Effect:	Incorrect port information.
Corrective Action Taken:	The information is updated and the rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The value of <i>cic</i> in rIORTLA tuple.
<i>Good Data</i> =	The value of <i>pci</i> in rIORTLA tuple.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIFC_TRUNK tuple.

Table 62 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	There is a mismatch between static data relations for this rIORTLA tuple.
Possible Error Effect:	There is no way for the switch to figure out if the port should have this service.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Office personnel should determine which relation is missing and insert it using ODBE. If the port is really supposed to be nailed up, office personnel must populate the rINAILUP tuple for this port using ODBE. If the port should not be nailed up, office personnel should use ODBE to clear the nail-up data from the rIPRTTYP tuple for this port. If this is a modem pool port, the missing relations should be inserted for this rIORTLA. If the rIFC_PORTTYP is dumped, then there is a missing rIFC_TRUNK or rIPP_MEM tuple which needs to be inserted. (If the <i>Good Data</i> is the relation ID for rIPP_MEM, insert that tuple.)
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple or, when <i>Good Data</i> is the relation ID for RLPP_MEM, the key of the missing rIPP_MEM tuple.
<i>Bad Data</i> =	Relation ID for the missing tuple. • RLEQUIPDSL. • RLNAILUP. • RLPORT_GROUP. • RLPSGP_PRT. • RLPSX25LN. Otherwise, the <i>servclass</i> field in rIFC_PORTTYP or, when <i>Good Data</i> is the relation ID for RLPP_MEM, the rIFC_PORTTYP.lp_type.
<i>Good Data</i> =	NA or, if missing rIFC_TRUNK, then this is <i>lp_type</i> in rIFC_PORTTYP or the relation ID RLPP_MEM when missing that tuple.
<i>Logical Index</i> =	Logical key of rIORTLA tuple (port name).
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIFC_PORTTYP tuple if missing rIFC_TRUNK.

Table 63 Error Code: OOS_MATE

Error Code:	OOS_MATE
Error Description:	An OSPS Autoquote DSL incorrectly has a different basic port status from its associated mate analog line.
Possible Error Effect:	If the Autoquote DSL is incorrectly in-service, it will be hunted but it will not be possible to get control of the mate analog line. Autoquote call failures will result whenever this pair is hunted. If the Autoquote DSL is incorrectly out-of-service, the pair will never be hunted and therefore will never be used for Autoquote

	calls.
Corrective Action Taken:	The Autoquote DSL is recovered to the same basic state as its mate analog line.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the Autoquote DSL rIORTLA tuple.
<i>Bad Data =</i>	Port status basic state for the Autoquote DSL.
<i>Good Data =</i>	Port status basic state for the Autoquote mate analog line.
<i>Logical Index =</i>	Port name of the Autoquote DSL.
<i>Dump 1 =</i>	rIORTLA tuple for the Autoquote DSL.
<i>Dump 2 =</i>	rIORTLA tuple for the Autoquote mate analog line.

Table 64 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	This rIORTLA is linked to an invalid list.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	The rIORTLA owner linkage proc_cplr.st_headk.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key (port name) of the rIORTLA tuple.
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 65 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Something is wrong with the logical key of the owner of the rIORTLA tuple.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	The port list of the rIORTLA tuple (rIORTLA.list).
<i>Good Data =</i>	The owner link of the rIORTLA tuple (rIORTLA.prlist.st_ownk - DBSFPTLST).
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 66 Error Code: PARENT

Error Code:	PARENT
Error Description:	The PORTLA audit is attempting to determine the facility circuit associated with the port. However, the SMIM conversion macro has returned a value of null circuit.
Possible Error Effect:	The port may be unusable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return value from SMIM function.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key to the rISMEST tuple.
<i>Dump =</i>	The rICKTDATA tuple.

Table 67 Error Code: P_UNKSTAT1

Error Code:	P_UNKSTAT1
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the port status contained by the terminal process system.
Possible Error Effect:	Packet switching inter-SM routing could be failing.

Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA.
<i>Dump</i> =	rIORTLA tuple.

Table 68 Error Code: PHNOOS

Error Code:	PHNOOS
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the PH status.
Possible Error Effect:	Call processing could be affected because the port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	PH status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 69 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIPICB tuple is not in the database.
Possible Error Effect:	Manual asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIPICB tuple.
<i>Dump</i> =	NA.

Table 70 Error Code: PORT_TM

Error Code:	PORT_TM
Error Description:	The terminal maintenance list fields do not agree with the rIORTLA list fields.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The port is placed on the to-be-idled for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	rIORTLA owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 71 Error Code: PORT

Error Code:	PORT
Error Description:	The PORTLA audit attempted to convert a port name to a circuit name, but the

	PC utility macro returned a null value indicating that the circuit is not equipped.
Possible Error Effect:	The port will not be usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA (0)
<i>Good Data =</i>	Circuit name.
<i>Logical Index =</i>	The key to the rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 72 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	An attempt to convert a port name to a port index has resulted in an indication that the port does not exist (null index).
Possible Error Effect:	The Port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Port name.
<i>Good Data =</i>	NA (0)
<i>Logical Index =</i>	Port index.
<i>Dump =</i>	NA.

Table 73 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The port status in the rIORTLA implies the port should be on a different list than it is actually on.
Possible Error Effect:	Attempts to change a port's mtce status will result in asserts.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	If port had been manually removed from service, it must be removed from service again.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Port status.
<i>Good Data =</i>	rIORTLA list.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 74 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve an rIGPFTIMER from the key stored in one of the rIORTLA tuple's timer key fields.
Possible Error Effect:	Varies according to conditions including which timer field was being accessed.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data =</i>	The rIGPFTIMER key returned from the read attempt.
<i>Good Data =</i>	The rIGPFTIMER key from the timer key field.
<i>Logical Index =</i>	The rIORTLA tuple logical key (rIORTLA.port).
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIGPFTIMER tuple buffer resulting from the read attempt.

Table 75 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The audit calls macro PSSVCPDLCL() to get the dlci for a frame relay SVC trunk, and PSBADDLCI is returned because the rIFC_PORTTYP or

	rIFRBC_SVC tuple is not found, or the port is not found within the rIFRBC_SVC tuple.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly.
Corrective Action Taken:	None -- see Manual Action.
Manual Action Required:	1 = Check for, and if missing, insert the missing rIFC_PORTTYP tuple with rIFC_PORTTYP.port.member set to rIORTLA.port and rIFC_PORTTYP.port.module set to Glpcr_id. 2 = Check for, and if missing, insert the missing rIFRBC_SVC tuple with rIFRBC_SVC.gport set to rIFC_PORTTYP.sleevid. 3 = Check for, and if missing, add this port (rIORTLA.port) to an empty rIFRBC_SVC.mport[] in the rIFRBC_SVC tuple keyed by rIFC_PORTTYP.sleevid.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Local "dcli" returned from PS macro call.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple (rIORTLA.port).
<i>Dump</i> =	rIORTLA tuple.

Table 76 Error Code: SHELF_SLC

Error Code:	SHELF_SLC
Error Description:	An attempt to determine the DCLU unit number from the S96Shelf circuit name has resulted in a return value of null unit.
Possible Error Effect:	The TAU line hardware path will not be torn down and re-established. The TAU line to TSI path will remain connected.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA (0)
<i>Good Data</i> =	Shelf name.
<i>Logical Index</i> =	Return value from SMIM function.
<i>Dump</i> =	NA.

Table 77 Error Code: SHELFREAD

Error Code:	SHELFREAD
Error Description:	Cannot read rIS96SHELF tuple.
Possible Error Effect:	No checking can be done on this port.
Corrective Action Taken:	None.
Manual Action Required:	Check ODD information for this relation.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	RLS96SHELF relation ID.
<i>Logical Index</i> =	Bank name.
<i>Dump</i> =	NA.

Table 78 Error Code: SLAVE_LINK

Error Code:	SLAVE_LINK
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.

<i>Bad Data</i> =	"master" field in TMDlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 79 Error Code: SLC_PORT

Error Code:	SLC_PORT
Error Description:	The PORTLA audit attempted to convert a port name into the corresponding shelf or bank name. The return from the PC function indicated that the bank or shelf does not exist.
Possible Error Effect:	The port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Port name.
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Return value from the PC function.
<i>Dump</i> =	NA.

Table 80 Error Code: SLCLOSTTUP

Error Code:	SLCLOSTTUP
Error Description:	Cannot read the rS96T1USG tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Skip auditing this tuple and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rS96T1USG tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	rS96T1USG tuple relation ID.
<i>Logical Index</i> =	Bank name.
<i>Dump</i> =	NA.

Table 81 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for this rPORTLA cannot be found.
Possible Error Effect:	This rPORTLA cannot be checked for correct FE state.
Corrective Action Taken:	Related audits will recover this port if required.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA (0).
<i>Bad Data</i> =	NA (0).
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 82 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	Contents of state.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 83 Error Code: STATE_IDL

Error Code:	STATE IDL
Error Description:	The state field indicates idle linkage, but the linkage does not indicate that the rIORTLA is linked to its owner (as indicated by the contents of the list field).
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Contents of owner link.
<i>Good Data =</i>	Contents of list.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 84 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	The state field indicates that linkage exists but the link fields are empty.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Contents of the bad link field.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 85 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rIORTLA tuple was found linked to an unknown service list. The rIORTLA.list field must be marked as one of the legal service lists (either TMIS for in-service, or one a variety of possible out of service lists) as specified by DMLIST. This error may occur if a new service list has been added by terminal maintenance, but the PORTLA audit was never modified to recognize it.
Possible Error Effect:	The port will be unavailable to call processing and terminal maintenance activities since the associated rIORTLA tuple is in an unknown state.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple.
<i>Bad Data =</i>	The rIORTLA.list field found by the audit.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIORTLA tuple (port name).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 86 Error Code: STATUS

Error Code:	STATUS
Error Description:	The port status supplementary information in the rIORTLA is invalid for the relationship between this SM and its mate (rIISMPATH.mate_sm). If rIORTLA.suppl_info is <i>TMDLI</i> , <i>TMSNLI</i> , or <i>TMTMSCONN</i> , the two SMs must not be in an HSM-RSM relationship or an RSM-RSM relationship under the same HSM.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance

	status will be cleared out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The port status supplementary information (rIORTLA.suppl_info).
<i>Bad Data</i> =	The relationship between the SM and its mate.
<i>Good Data</i> =	The mate SM number which is served by the inter-SM path (rISMPATH.mate_sm).
<i>Logical Index</i> =	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 87 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	An rIORTLA tuple contains an invalid telephone status monitoring value (rIORTLA.tsm_stat). The status is inconsistent with a value derived from the associated rIPCBLA tuple or the rIPCBLA tuple was not accessible.
Possible Error Effect:	Telephone status monitoring may not work correctly for an end user.
Corrective Action Taken:	If the correct status was accessible, the audit corrects the value in the rIORTLA tuple (rIORTLA.tsm_stat). Otherwise the port is reinitialized and put on the to-be-idled list to be recovered by terminal maintenance.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A boolean indicating if the associated status data was accessible.
<i>Bad Data</i> =	The rIORTLA tuple's telephone monitoring status (rIORTLA.tsm_stat).
<i>Good Data</i> =	The correct value of the port's monitor status value (rIORTLA.tsm_stat).
<i>Logical Index</i> =	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump 1</i> =	The rIORTLA tuple.
<i>Dump 2</i> =	The rIFC_PORTTYPE tuple.

Table 88 Error Code: STUCK

Error Code:	STUCK
Error Description:	The multipoint analog and digital ports have list fields which indicate in-service but the CPE is out-of-service.
Possible Error Effect:	Multipoint modem pool ports are on an out-of-service CPE. If this CPE should be in-service, the CPE should be power cycled.
Corrective Action Taken:	The analog port is idled for recovery to valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of analog rIORTLA tuple.
<i>Bad Data</i> =	CPE state.
<i>Good Data</i> =	Analog rIORTLA list.
<i>Logical Index</i> =	Key of analog rIORTLA tuple.
<i>Dump 1</i> =	Analog rIORTLA tuple.
<i>Dump 2</i> =	Digital rIORTLA tuple.
<i>Dump 3</i> =	rITEILA tuple.

Table 89 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	The analog modem pool port is in service but the corresponding digital one is not.
Possible Error Effect:	Modem pool call could be failing.
Corrective Action Taken:	Both ports are recovered by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	List the analog port is on.
<i>Good Data</i> =	List the digital port is on.
<i>Logical Index</i> =	Analog port name.
<i>Dump 1</i> =	Analog rIORTLA tuple.
<i>Dump 2</i> =	The corresponding digital rIORTLA tuple.

Table 90 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The D-port is on the TMLTWAIT list but the T321 index is not set (rIPRISTAT.t321_idx = TMTNULLIDX).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The list from rIORTLA tuple (rIORTLA.list).
<i>Bad Data =</i>	The T321 index from the rIPRISTAT tuple (rIPRISTAT.t321_idx).
<i>Good Data =</i>	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIPRISTAT tuple.

Table 91 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	When the state of the rIORTLA tuple is DBPSIDLE, the wait for restoration bit should be cleared (rIORTLA.rstawait = DBNO).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The <i>rstawait</i> attribute is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The waiting for service acknowledge message flag of the rIORTLA tuple (rIORTLA.srvawait).
<i>Bad Data =</i>	The waiting for restoration flag of the rIORTLA tuple (rIORTLA.rstawait).
<i>Good Data =</i>	The logical port type.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 92 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The B-port has been in a transient state for too long (longer than audit cycle).
Possible Error Effect:	This tuple is in a stuck OOS state and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	port_owner.
<i>Good Data =</i>	d_owner.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIORTLA tuple for D-port.

Table 93 Error Code: TRUNK_LINE

Error Code:	TRUNK_LINE
Error Description:	The port type for this port claims that it is a trunk but the rICKTDATA tuple that should exist for the trunk is missing.
Possible Error Effect:	There is no way for the audit to recover this port without the rICKTDATA information.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Determine whether the port is supposed to be a trunk. If it is, the add the rICKTDATA tuple using ODBE, or delete the rIORTTYP tuple. If a line, the rIORTTYP tuple needs to be updated to indicate that the port is a line.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Service class of rIORTLA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 94 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The logical key of the owner of the rPORTLA tuple points to an unassigned port, but the service class is not unassigned (rIFC_PORTTYP.servclass != RC_PORTUNA).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPORTLA tuple.
<i>Bad Data =</i>	The owner link of the rPORTLA tuple (rPORTLA.prtlst.st_ownk - DBSFRTLST).
<i>Good Data =</i>	The service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index =</i>	Key of rPORTLA tuple.
<i>Dump =</i>	The rPORTLA tuple.

Table 95 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	One of two possible errors: (1) A CCS trunk is found on a CCSINIT or a trunk maintenance list for non-CCS trunks. (2) A non-CCS trunk is found on the CCSINIT or the trunk maintenance continuity list for only CCS trunks.
Possible Error Effect:	Corresponds to Error Description as follows: (1) The routing and terminal allocation subsystem won't hunt this trunk. Trunk maintenance may have difficulty recovering this trunk. (2) Any continuity testing done on a non-CCS trunk will fail.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status. When the CCS trunk is put on the to-be-idled list, a reset message is sent to the far-end office.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPORTLA tuple.
<i>Bad Data =</i>	Contents of list.
<i>Good Data =</i>	Contents of owner link.
<i>Logical Index =</i>	Key of rPORTLA tuple.
<i>Dump =</i>	rPORTLA tuple.

PORTLA-C

Software Release: 5E13 only

1. PORT LINKAGE AREA (PORTLA) AUDIT

Each port has a port linkage area tuple that describes the status and associations of the port. These port linkage areas contain information such as the owner of the port, processes that are waiting to access the port, the status of the port (precut or postcut, and so forth), and a preemption flag that shows whether the port is being preempted.

The port linkage area audit (PORTLA) verifies the data contained in the port linkage areas. When the PORTLA audit is invoked, it checks all ports (both lines and trunks).

2. DATA STRUCTURES

The data structure checked by the PORTLA audit is the port linkage area relation (RLPORTLA). The RLPORTLA relation has one-to-one linkages with the pending history relation (RLPHISTORY) and the process control block linkage area relation (RLPCBLA). It also has owner-member linkage with the coupler relation (RLCPLR) and member linkage with the port list relation (RLPRTLST). The RLPRTLST relation has 16 head cells, each of which owns a list of port linkage areas. Each list indicates a particular port status (for example, in service, out of service, to be idled, and so forth). The port linkage area list item contains an ID that indicates the list to which the port linkage area belongs. The PORTLA audit uses the feature control port name to service class relation (RLFC_PORTTYP) to perform an existence check.

Analog modem pool ports will verify the existence of the related packet switch rIORTLA tuple and its corresponding rIDPB tuple. Additional multipoint modem pool checks have been added for the following data structures:

- The terminal equipment data block relation (RLTEDB).
- The packet switching X.25 level 3 data for DSL's relation (RLPSX25LN).
- The terminal endpoint identifier linkage area relation (RLTEILA).

The PORTLA audit added checks of the following SLC[®] 96 and network management data structures:

- The trunk group selective incoming load control relation (RLTKGSILC).
- The switch maintenance equipment status table relation (RLSMEST).
- The circuit data block common area relation (RLCDBCOM).
- The SLC[®] 96 equipage relation (RLS96EQUIP).
- The SLC[®] 96 T1 usage relation (RLS96T1USG).

Figure 1 shows these data structures.

The PORTLA audit insures the integrity of the port linkage area relation (RLPORTLA). A field in the rIORTLA tuple checks for datalink trunks and lines. If the port is of a particular service class, the datalink bit should be set. If it is not, this check prints an error and then sets the bit. It then continues processing and it does not recover the tuple.

3. AUDIT CHECKS

The PORTLA audit uses the static driver (DRSTATIC) to step through the RLPORTLA relation.

- (a) Initially, the audit calls a dynamic to static existence check function to perform the following checks.
 - (1) This existence check verifies that, for every dynamic rIORTLA tuple, a static port name to service class tuple (rLFC_PORTTYP) exists. If the static rLFC_PORTTYP tuple does not exist, the extra rIORTLA tuple is deleted (EXTRATUP).
 - (2) For each tuple in RLPORTLA that is associated with a circuit, the existence check verifies that a corresponding tuple exists in the static circuit data relation (RLCKTDATA) for those service classes that should have a tuple (TRUNK_LINE).
- (b) The audit calls a semantic check function to perform the following checks.
 - (1) The audit checks to see that the rIORTLST owning head cell of the port linkage area, as determined from the port linkage area's owner linkage item, agrees with the list item (LIST).
 - (2) The audit checks the cutover status of the office and port. If the office is in a cutover state, the audit checks that the rIORTLA tuple is on the correct list (CUTOVER).
 - (3) The port status (pstatus) item contains the primary maintenance status of the port. The audit verifies that the list to which the port linkage area belongs is consistent with the pstatus item (PSTATUS).
 - (4) The audit verifies that the hunt item in the RLPORTLA relation is consistent with the hunt item in the RLFC_PORTTYP relation (ODD_DATA).
 - (5) The audit verifies that the port type (ptype) item in the RLPORTLA relation is consistent with the port type (porttype) item in the RLFC_PORTTYP relation (ODD_DATA).
 - (6) The audit verifies the datalink item with valid service class (BLINK).
 - (7) If the port is a digital carrier port, the audit verifies that the list to which the port linkage area belongs is consistent with the state of the digital facility interface (DFI) circuit. An active red or yellow alarm should correspond to ports on the facility failure list. Family-of-equipment and to-be-idled lists are also allowed because they are higher priority lists. If both the red and yellow alarms are inactive, the port linkage area can be on any list except the facility failure list (ALARM).
 - (8) Common channel signaling ports are verified to be on valid lists by this audit (WRONGLIST).
 - (9) If the rIORTLA tuple is on a network management list, the corresponding rITKGSILC tuple is read (MISSTUP). Several fields, including the port and quarantine state fields, are checked (PORT_TM).
 - (10) The audit verifies that OSPS Autoquote DSL's have a mate analog rIORTLA tuple (OOS_MATE).
 - (11) The audit verifies that nailup ports have a mate rINAILUP tuple.
 - (12) During linkage checking of the rIORTLA tuple the function AUptlchk() performs point-to point-back checks between the rIORTLA and rIADB relations. If an error is found, an error report is printed to the read-only printer (ROP) and recovery is started.
 - (13) PORTLA audits the linkage between the rIORTLA relation and the RLADB relation for shared directory number (DN) analog. When the PORTLA audit detects the error, the rIORTLA tuple is linked to the rIADB tuple and the PORTLA audit checks to make sure the sh_analog bit in the RLFC_PORTTYP relation is set to DBYES. If sh_analog bit is not set, the PORTLA audit prints an error on the ROP and returns AUFALSE to start recovery (BDLINK).
 - (14) A port linkage area may be linked to one process control block linkage area directly or to one or more process control block linkage areas by means of couplers. Also, it may not be linked to any process

control block linkage areas. The port linkage area state item indicates which of these two states exists. The audit checks to see that the state item agrees with the actual linkage (STATE_IDL or STATE_LNK or STATE).

- (15) In a remote switching module (RSM) environment, if a port is idle, hunted, and in service, the routing and terminal allocation (RTA) subsystem is called to mark the port idle in the hunted port status relations.
 - (16) If the logical port type field of rLFC_PORTTYP indicates that this is a SLC[®] 96 special line, the RLS96T1USG relation is checked to ensure that one dedicated time assignment unit (TAU) is associated with the port (BADTSCOUNT). If it is determined that one does not exist (or more than one exists), the current hardware connection is torn down and a new one is established (SLCLOSTTUP). If an rLS96SHELF tuple is not found, the error is reported (SHELFREAD).
 - (17) For SLC[®] 96 ports, the audit determines whether the port is associated with a 4AB1 or 4AC1 line unit or SLC[®] 96 digital carrier line unit (DCLU). If the port is associated with a DCLU, the key of the rLS96EQUIP tuple is calculated and the tuple is read. The bank status is checked to determine the grid identification, the associated rSMEST tuple is read (SMEST), and the port is conditioned.
 - (18) There is a list of precut status ports contained in the switching module (SM) protected memory. The audit calls a function to determine whether the port associated with the port linkage area being checked is on this list. If it is, the audit checks whether the port linkage area is linked to the rLPRTLST head cell that owns precut status port linkage areas.
 - (19) If the port is an ISDN port, the audit checks the parent circuit(s). It reports an error if the port's maintenance state is inconsistent with the maintenance state of the parent circuit. The audit also checks the validity of family-of-equipment states.
 - (20) This audit also has special checks for modem pool ports. If the port is a modem pool port, the analog and corresponding packet switch ports will be synchronized. For multipoint modem pool ports, the CPE status will be synchronized.
- (c) The PORTLA audit then calls a function to maintain consistency between the monitor busy & idle (B&I) data table, TMbitable[] and the rIPORTLA tuple. The TMbitable[] contains the port and TM process ID information for any ports involved in monitoring using the monitor (B&I) feature.

4. ERROR HANDLING

For all errors except those encountered when the audit is running in the initialization mode, the port linkage area is placed on the to-be-idled list (this list is owned by an rLPRTLST head cell). At the same time, a terminal maintenance process is activated. This process checks the hardware state and the contents of the switch maintenance equipment status table relation (RLSMEST) to determine what the status should be for a given port. For circuits marked out of service (OOS) in rSMEST, the associated ports are given an OOS, family of equipment (FE) status. DFI ports with active red or yellow carrier alarms are given an OOS, facility failure (FAF) status. Ports whose scanning circuitry indicate that they are off-hook are given an OOS, high and wet (HW) status. All other ports are given an in-service status.

For errors encountered when the audit is running in the initialization mode, the port linkage area is linked to the in-service list except for the ports as determined by the service class. Common channel signaling (CCS) trunks are linked to the disabled list and recovered by terminal maintenance (TM) subsystem. D-channels, inter-switch trunks, ISM nailups, X.75 and X75' trunks, and packet switching B-channels are linked to the DSL initialization list and recovered by the TM subsystem. B-channels are linked to the D-channel out-of-service list if the D-channel is out-of-service, else they are linked to the in-service list. In all cases except during initializations, errors are reported

by means of output messages.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALARM

Error Code:	ALARM
Error Description:	The list the rIORTLA is on is inconsistent with the state of the digital carrier alarm. Digital carrier ports only.
Possible Error Effect:	If carrier alarm is ON, the port is incorrectly in service and if used by call processing, would cause call failures. If carrier alarm is OFF, the port is incorrectly out of service and thus unavailable for call processing use.
Corrective Action Taken:	The port will be placed on the to-be-idled list for later idling. The software maintenance status of the port will be made consistent with the hardware (carrier) alarm state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	List owner for rIORTLA tuple.
<i>Good Data</i> =	Carrier group alarm status.
<i>Logical Index</i> =	Key of the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rSMEST tuple.

Table 2 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rSMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rSMEST tuple.

Table 3 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	An rIORTLA exists for a port whose service class indicates that an rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Service class of the port.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 4 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.

Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	Relation ID of bad tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	Tuple.

Table 5 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	An rSMEST tuple for the circuit under audit or the parent/mate circuit for this circuit was found to be in an invalid state with respect to the information in the MRA deferred work recovery queue. The qual_2 state of SMAUOOS is used to indicate when audits or SI has recovered a circuit, parent or mate to the MRA queue. Note, refer to <i>Good Data</i> for details.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly by MRA.
Corrective Action Taken:	The audit will attempt to recovery the circuit using the MRA queue and/or leave it in a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple.
<i>Bad Data</i> =	qual_2 field in the SMEST tuple.
<i>Good Data</i> =	Return value from AUMRARET (au/hdr/AUglobckt.L).
	0 = Circuit was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	2 = Parent/mate was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	4 = Circuit is marked AUOOS but not in queue, parent is not in queue, circuit/parent/mate is not MRA BUSY.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump 1</i> =	rSMEST tuple.
<i>Dump 2</i> =	parent/mate rSMEST tuple.

Table 6 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	An rIDS0B tuple is involved in a hairpin connection but the rIORTLA.dsob field does not have the correct value.
Possible Error Effect:	Call processing will not be able to use that rIORTLA tuple.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	rIDS0B.use field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The DS0B key.
<i>Dump 1</i> =	rIDS0B tuple.
<i>Dump 2</i> =	rIORTLA tuple.

Table 7 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The rIDS0B tuple is involved in a hairpin or nailup connection but the rIDS0B.portname field tuple is not matching the rNAILUP.port1 field.

Possible Error Effect:	There will be no hairpin or nailup connection established since a wrong port is being used.
Corrective Action Taken:	The rPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDS0B tuple.
<i>Bad Data =</i>	The rIDS0B tuple key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIDS0B tuple.
<i>Dump 1 =</i>	rIDS0B tuple.
<i>Dump 2 =</i>	rPORTLA tuple.

Table 8 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	An rPORTLA tuple contains the key of an rGPFTIMER tuple in one of it's timer key fields but the rGPFTIMER tuple is either not linked back (rGPFTIMER.udata contains a different relation ID or key from the owning tuple) or the timer type (rGPFTIMER.timer_type) does not have the expected value for the timer key field being used.
Possible Error Effect:	Varies according to conditions including the rPORTLA usage and which timer key field is being checked.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data =</i>	The relation ID stored in rGPFTIMER.udata.
<i>Good Data =</i>	The tuple key stored in rGPFTIMER.udata.
<i>Logical Index =</i>	The rPORTLA tuple logical key (rPORTLA.port).
<i>Dump 1 =</i>	The rPORTLA tuple.
<i>Dump 2 =</i>	The rGPFTIMER tuple.

Table 9 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	For CCS7 trunks, there are four possible conditions for which this error is reported. (1) There is a reset in progress, but the <i>reset_inprog</i> bit in PORTLA is not set to DBYES. (2) There is a reset in progress with the <i>reset_inprog</i> bit in PORTLA set to DBYES, but the port is not owned. (3) There is no reset in progress, but the <i>reset_inprog</i> bit in PORTLA is set to DBYES. (4) There is no entry for this port, but the <i>reset_inprog</i> bit in PORTLA is set to DBYES. For CCS6 trunks, the following fields in PORTLA are not set correctly. These fields include <i>protocol</i>, <i>cic</i>, <i>pci</i>, <i>reset_inprog</i>.
Possible Error Effect:	Incorrect port usage.
Corrective Action Taken:	If the problem is in regards to the <i>reset_inprog</i> bit, then the bit will be set to its correct value. If the error is in regards to the CCS6 trunk, then the following will occur. <i>protocol</i> will be set to DBCCIS6, <i>cic</i> will be set to CCNULCIC, <i>pci</i> will be set to CCNULPCI, and <i>reset_inprog</i> will be set to DBNO.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple for CCS6 trunks. Error flag value for CCS7 trunks.
<i>Bad Data</i> =	Value of in_sigtype from rIFC_TRUNK for CCS6 trunks. Value of mtype returned from TM for CCS7 trunks.
<i>Good Data</i> =	Value of out_sigtype from rIFC_TRUNK for CCS6 trunks. Value of type returned from TM for CCS7 trunks.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 10 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	One of two possible errors: (1) The port is on the TBI (to be idled) list and is still waiting to be run but the tbi_flag is set, indicating that the job is running. (2) The port is not on the TBI list and the flag is set.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	Corresponds to Error Description as follows: (1) The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status. (2) The tbi_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	When <i>Good Data</i> is 0, the to-be-idled flag of the rIORTLA tuple (rIORTLA.tbi_flag). When <i>Good Data</i> is 1, the return value from TMisqueued() function call.
<i>Good Data</i> =	0 or 1.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 11 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The D-port is linked to the unassigned list but the service class taken from static data does not indicate that the port should be on unassigned list.
Possible Error Effect:	The rIORTLA is unavailable for call processing.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to the proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Owner of rIORTLA tuple.
<i>Good Data</i> =	Service class of port.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 12 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT
Error Description:	A SLC [®] 96 time slot was found to have more that one time assignment unit assigned to it.
Possible Error Effect:	Possible origination problems for some mode II SLC [®] 96 special lines.
Corrective Action Taken:	Reconstruct the rIS96T1USG tuple with one time assignment unit.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rIS96T1USG tuple.
<i>Bad Data</i> =	Count of T1TS lines for this time assignment unit.
<i>Good Data</i> =	Number of time slots allowed (1).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIS96T1USG tuple.

Table 13 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An rIDS0B tuple was used for hairpin or nailup connection but the rIDS0B.use field was not set to indicate that.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDS0B tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	rIDS0B tuple.
<i>Dump 2</i> =	rIPORTLA tuple.

Table 14 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The rIPORTLA tuple is on the wrong list. The port could be stuck on an out-of-service list when it should be in-service or on an is-service list when it can not be used because of the parent circuit. There are two possible errors: (1) The port is not OOS-FE, but the parents circuits are OOS. (2) The port is OOS-FE, but the parents circuits are not OOS. If the port is on a IDCU TR08 the audit will dump the shelf/bank that was checked. For analog ports on an IDCU TR303, one DS1 will be dumped but all DS1's on the RT303 will be checked.
Possible Error Effect:	Either the port will be used for a call which cannot complete because of out-of-service equipment or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	List owner of rIPORTLA tuple.
<i>Good Data</i> =	Logical port type.
<i>Logical Index</i> =	Logical key of rIPORTLA tuple.
<i>Dump 1</i> =	rIPORTLA tuple.
<i>Dump 2</i> =	rSMEST tuple with port as key.
<i>Dump 3</i> =	rSMEST tuple for PH if available.
<i>Dump 4</i> =	rSMEST tuple for PSU if available.

Table 15 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The rIPORTLA was found linked to the rIADB tuple when static data (rIFC_PORTTYP) did not have the shared analog bit set.
Possible Error Effect:	Call processing cannot access the port correctly.
Corrective Action Taken:	The rIPORTLA tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPORTLA tuple.

<i>Bad Data</i> =	Linkage field to the rIADB tuple.
<i>Good Data</i> =	Shared analog bit in rIFC_PORTTYP tuple.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 16 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The packet switching modem pool port is not bound (rITEDB tuple to rTEILA tuple).
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITEDB tuple.
<i>Bad Data</i> =	Key to rTEILA tuple (0 means no rTEILA tuple).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rITEDB tuple.
<i>Dump</i> =	rITEDB tuple.

Table 17 Error Code: BINDWRONG

Error Code:	BINDWRONG
Error Description:	The packet switching modem pool port is not bound to the correct rITEDB tuple.
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA tuple.
<i>Bad Data</i> =	TEDB key from rTEILA tuple.
<i>Good Data</i> =	Key of rITEDB tuple (logical port).
<i>Logical Index</i> =	Logical key of rTEILA tuple.
<i>Dump</i> =	rTEILA tuple.

Table 18 Error Code: BLINK

Error Code:	BLINK
Error Description:	The rIORTLA service class was of the datalink type but the datalink field was not set to DBYES.
Possible Error Effect:	The line or trunk would not be treated as a datalink for removes/restores.
Corrective Action Taken:	The datalink bit will be set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	datalink field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 19 Error Code: BTRANCK

Error Code:	BTRANCK
Error Description:	The associated D-port is out-of-service and this B-port is not on a list to indicate the unavailability of the D-port. This is considered to be valid for a short transient period. This error is not reported for at least 5 minutes.
Possible Error Effect:	Call processing cannot complete service on this port.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	D-port owner.

<i>Logical Index</i> =	Logical key of rIPORTLA tuple.
<i>Dump 1</i> =	rIPORTLA tuple.
<i>Dump 2</i> =	D-channel rIPORTLA tuple.
<i>Dump 3</i> =	rIPRISTAT tuple.
<i>Dump 4</i> =	when the <i>Ip_type</i> is SVC, rIDPB tuple.

Table 20 Error Code: CADNPORT

Error Code:	CADNPORT
Error Description:	The port is either on the CADN list when the <i>cadn</i> field in FCPORRTYP is set to TMIIS or the port is not on the CADN list when the <i>cadn</i> field in FCPORRTYP is not TMIIS. For both of these cases, there error is only reported if the port is not owned.
Possible Error Effect:	The port may not be able to be used by call processing.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPORTLA tuple.
<i>Bad Data</i> =	<i>cadn</i> field in rIFC_PORTTYP tuple.
<i>Good Data</i> =	<i>list</i> field in rIPORTLA tuple.
<i>Logical Index</i> =	Logical key for the rIPORTLA tuple.
<i>Dump</i> =	rIPORTLA tuple.

Table 21 Error Code: CKTDATA (1)

Error Code:	CKTDATA (1)
Error Description:	The PORTLA audit is attempting to read a nonexistent rICKTDATA tuple. The key to the rICKTDATA tuple is derived from the rIFC_PORTTYP tuple.
Possible Error Effect:	Call processing could be affected.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIPORTLA tuple.
<i>Dump</i> =	rIPORTLA tuple.

Table 22 Error Code: CKTDATA (2)

Error Code:	CKTDATA (2)
Error Description:	The rICKTDATA tuple is missing from the database. The key used to read the tuple was derived from the <i>path2_ds0.ds1name</i> in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rICKTDATA tuple.
<i>Dump 1</i> =	First rINAILUP tuple
<i>Dump 2</i> =	Second rINAILUP tuple

Table 23 Error Code: CUTOVER

Error Code:	CUTOVER
Error Description:	A port is on the wrong list based on the office's or this ports cutover status.
Possible Error Effect:	Pre-cut ports could incorrectly be receiving service from this office or post-cut (active) ports could be denied service.
Corrective Action Taken:	Port is conditioned and returned to correct list (maintenance status).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	List owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 24 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit failed a read of the RLDSLEQUIP or RLISMPATH relation associated with this RLPORTLA.
Possible Error Effect:	There could be an ODD inconsistency.
Corrective Action Taken:	The RLPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	If the error address is null, check the RLDSLEQUIP relation. If the error address is not null, check the RLISMPATH relation. The key used is printed in <i>Bad Data</i> .
Dump Description:	
<i>Error Address</i> =	NA if the failed read is rIDSLEQUIP. If it is rIISMPATH, the port status supplementary information (rIORTLA.suppl. info).
<i>Bad Data</i> =	If the failed read is rIDSLEQUIP, DSL GROUP. If it is rIISMPATH, the index to the port (rIFC_PORTTYP.port_indx).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name (rIORTLA.port).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 25 Error Code: DLI_FE

Error Code:	DLI_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the DLI status or NLI status on the major ONTC side. The inter-SM port indicated out-of-service family of equipment but the DLI or all NLIs on the major ONTC side is not out-of-service.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data</i> =	The DLI status in data delivery.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 26 Error Code: DLOSTTUP

Error Code:	DLOSTTUP
Error Description:	The mate D-port tuple does not exist for this rIORTLA. A major problem but recovery will be completed in the PORTTYP audit.
Possible Error Effect:	Unable to complete packet switching calls.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	portkey of packet rIORTLA tuple.
<i>Good Data</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 27 Error Code: DPB_LINK

Error Code:	DPB_LINK

Error Description:	An rIDPB does not exist for the packet switching port for this analog modem pool port.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate (DPB/PORTTYP) audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of packet switch rIORTLA tuple.
<i>Bad Data</i> =	Key of analog rIORTLA.
<i>Good Data</i> =	Key of rIDPB.
<i>Logical Index</i> =	Key of packet switch rIORTLA tuple.
<i>Dump</i> =	Packet switch rIORTLA tuple.

Table 28 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The rIDSLEQUIP tuple read failed.
Possible Error Effect:	The rIORTLA tuple has been garbaged, which caused the reading of the wrong rIDSLEQUIP tuple or the rIDSLEQUIP tuple has been garbaged which would give an incorrect state for the rIDSLEQUIP tuple.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	Check ODD to determine if the port is really a member of a DSL. If it is not, verify that the Ip_type and class of service fields in the RLPORTLA relation are correct, and do not indicate that the port belongs to a DSL. If none of these errors is found, and the other audits do not clean up the problem, seek help.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 29 Error Code: DSLGROUP

Error Code:	DSLGROUP
Error Description:	The packet switching port list does not match the status of its DSL group. There are two possible errors: (1) The port is not OOS-FE, but the DSL group is OOS. (2) The port is OOS-FE, but the DSL group is not OOS.
Possible Error Effect:	Either the port will be used for a call which cannot complete because the DSL group is out-of-service or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	DSL group.
<i>Good Data</i> =	Logical port type.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rSMEST tuple for the PH (if available).
<i>Dump 3</i> =	rSMEST tuple for the PSU (if available).

Table 30 Error Code: DS0B_LINK

Error Code:	DS0B_LINK
Error Description:	The field rIDS0B.use was set to PCD7SEC but the rIDS0B.portname field was not equal to rIORTLA.port field.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and

	restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIORTLA.port field.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 31 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	The audit tried to read the rIDS0B tuple for a given rIORTLA.ds0b field but the tuple was missing from the database.
Possible Error Effect:	The call processing could not use that DS0B time slot.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the rIDS0B tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	First rINAILUP tuple.
<i>Dump 2</i> =	Second rINAILUP tuple.

Table 32 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rIORTLA exists for a port whose service class indicates that a rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 33 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The grid group member number derived from the rIGSDB tuple does not agree with the equivalent number (grpname) from the rIORTLA tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The grpname in the rIORTLA tuple will be set to the group member number derived from the rIGSDB tuple in an attempt to recover consistency.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIORTLA unit type.
<i>Bad Data</i> =	rIORTLA tuple grpname.
<i>Good Data</i> =	grid group member from rIGSDB.
<i>Logical Index</i> =	Logical key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 34 Error Code: GPORT

Error Code:	GPORT
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but the monitoring bit in the rIORTLA tuple is not set.
Possible Error Effect:	Calls on this port may not be monitored.
Corrective Action Taken:	The monitoring bit is set to indicate that monitoring is active.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the TMbtable[].
<i>Bad Data</i> =	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	rIPORTLA tuple involved in monitor B&I.

Table 35 Error Code: GRP2INFO

Error Code:	GRP2INFO
Error Description:	A port was found with an invalid trunk group number (<i>grpnum</i>).
Possible Error Effect:	This port may be unusable for call traffic.
Corrective Action Taken:	The <i>grpnum</i> field corrected the value as indicated in the corresponding FC_TRUNK tuple, or changed to a null value when no corresponding FC_TRUNK tuple could be found.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rIPORTLA tuple.
<i>Bad Data</i> =	Invalid group number (rIPORTLA.grpnum).
<i>Good Data</i> =	Good value of <i>grpnum</i> (rIFC_TRUNK.tgroup) or null value.
<i>Logical Index</i> =	Port name involved.
<i>Dump 1</i> =	rIPORTLA tuple involved.
<i>Dump 2</i> =	rIFC_TRUNK tuple involved if present.

Table 36 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The rIPRISTAT tuple cannot be found from the rIPORTLA tuple.
Possible Error Effect:	The B-port cannot be checked against the D-port.
Corrective Action Taken:	Report the error and return no corrective action taken.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Key of rIPORTLA tuple.
<i>Bad Data</i> =	Pointer to the rIPRISTAT tuple.
<i>Good Data</i> =	The logical port type.
<i>Logical Index</i> =	Key of rIPORTLA tuple.
<i>Dump</i> =	The rIPORTLA tuple.

Table 37 Error Code: INT_DATA_BAD

Error Code:	INT_DATA_BAD
Error Description:	The value of rIPORTLA.dlci is not as expected when <i>Good Data</i> is 1, the logical port type is a SVC trunk and it should match the value obtained when the PS macro is called when <i>Good Data</i> is 2, the value of rIPORTLA.dlci should be zero.
Possible Error Effect:	The port may be sending messages to the wrong frame relay.
Corrective Action Taken:	Reset the rIPORTLA.dlci as expected.
Manual Action Required:	None
Dump Description:	
<i>Error Address</i> =	When <i>Good Data</i> is 1, the expected value of the rIPORTLA.dlci when <i>Good Data</i> is 2, the logical port type (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	the rIPORTLA.dlci.
<i>Good Data</i> =	1 or 2.
<i>Logical Index</i> =	Key of rIPORTLA tuple (rIPORTLA.port).
<i>Dump</i> =	rIPORTLA tuple.

Table 38 Error Code: INV

Error Code:	INV
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a <i>SLC</i> [®] 96 call which goes through that S-DFI is torn down.

Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data =</i>	Value found in that address.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump =</i>	The whole array AUs96inhp[].

Table 39 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The status of the inter-SM nailup in the rIORTLA is not valid.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Port status supplementary information.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 40 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	This rIORTLA is linked to an invalid list based on the logical port type.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	rIORTLA list.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIORTLA.
<i>Dump =</i>	rIORTLA tuple.

Table 41 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The rIORTLA tuple is linked to an invalid list. The tuple is associated with a D-channel, and the logical port type indicates remote protocol test access (rIFC_PORTTYP.lp_type == PCRDTA_D or PCRDTA_UD), but the port is on the level 2 or level 3 error port status list (rIORTLA.list == TMLTLV2ERR or TMLTLV3ERR). This is an illegal condition.
Possible Error Effect:	Remote digital test access (RDTA) actions and operations can not take place.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The port status list (rIORTLA.list) which was found.
<i>Bad Data =</i>	The key to the rIPRISTAT tuple (rIPRISTAT.prigrp)
<i>Good Data =</i>	Return code from validation of the PRI D-port status.
<i>Logical Index =</i>	The key of the rIORTLA tuple (port).
<i>Dump 1 =</i>	The contents of the rIORTLA tuple.
<i>Dump 2 =</i>	If this dump exists, it holds the contents of the D-channel's rIORTLA tuple.

Table 42 Error Code: INVMEM

Error Code:	INVMEM

Error Description:	An entry in TMbitable[] is not empty but the module attribute in bigport is invalid and indicates garbage in the table.
Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].
<i>Bad Data</i> =	Module in TMbitable[].bigport.module.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	None.

Table 43 Error Code: ISDNTUP

Error Code:	ISDNTUP
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIACT_SPR tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIACT_SPR tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIACT_SPR tuple.
<i>Dump</i> =	NA.

Table 44 Error Code: LIST

Error Code:	LIST
Error Description:	The list to which the portla should be linked does not agree with the list field used by the call processing and terminal maintenance programs.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes in error.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 45 Error Code: LNK_FE

Error Code:	LNK_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMS or NCT link status.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA tuple is recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data</i> =	The TMS link status in data delivery.
<i>Logical Index</i> =	Logical key to rIORTLA (port name).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 46 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The rCKTDATA tuple is missing from the database. The key used to read the tuple was derived from the path1_ds0.ds1name in the rNAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rNAILUP tuple.
<i>Bad Data =</i>	Loop number.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rCKTDATA tuple.
<i>Dump 1 =</i>	First rNAILUP tuple
<i>Dump 2 =</i>	Second rNAILUP tuple

Table 47 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	An rPORTLA tuple is missing from the database. The port used to read the tuple was taken from the rPORTLA.port1 or rPORTLA.port3_member field in the rNAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that port.
Corrective Action Taken:	The hairpin or nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rPORTLA tuple.
<i>Bad Data =</i>	The key used to read the missing rPORTLA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rPORTLA tuple.
<i>Dump 1 =</i>	First rNAILUP tuple.
<i>Dump 2 =</i>	Second rNAILUP tuple.

Table 48 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The packet switching modem pool port does not have a mate rTEDB or rTEILA tuple.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPORTLA.
<i>Bad Data =</i>	Key to missing tuple (logical port - rTEDB or party - rTEILA).
<i>Good Data =</i>	Relation ID of missing tuple (rTEDB or rTEILA).
<i>Logical Index =</i>	Key of tuple with missing link.
<i>Dump =</i>	rPORTLA tuple (if missing rTEDB) or rTEDB tuple (if missing rTEILA).

Table 49 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The status of the inter-SM nailup in the portla does not agree with the "pstat" field in the TMdlt_data[] array.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rPORTLA tuple.
<i>Bad Data =</i>	Port status.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rPORTLA tuple.

Table 50 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The status of the inter-SM nailup in the portla does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	"master" field in TMdlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 51 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	PORTLA audit is attempting to read a nonexistent tuple. There is a bad linkage between rIORTLA and rTKGSILC.
Possible Error Effect:	None. Call processing could be affected.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 52 Error Code: NOIDLELIST

Error Code:	NOIDLELIST
Error Description:	The rIORTLA is on an invalid list.
Possible Error Effect:	The port may be lost for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 53 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	An rIORTLA contains an invalid telephone status monitored value (rIORTLA.tsmmed). If the port is a trunk (rIFC_PORTTYPE.porttype = TM_TRK), either the port is marked as being monitored (rIORTLA.tsmmed != DBNO) or status is not idle (rIORTLA.tsm_stat != TSM_IDLE). If the port is a line, the monitored status is inconsistent with the ODD (no rIFC_LINE tuple exists and rIORTLA.tsmmed != DBNO or the rIFC_LINE tuple exists and rIORTLA.tsmmed != rIFC_LINE.tsmmed).
Possible Error Effect:	If the port is a trunk, asserts from telephone status monitoring code are likely. If the port is a line, telephone status monitoring may not work correctly for an end user.
Corrective Action Taken:	The audit corrects the value(s) in the rIORTLA tuple (rIORTLA.tsmmed = DBNO and rIORTLA.tsm_stat = TSM_IDLE for a trunk or rIORTLA.tsmmed = rIFC_LINE.tsmmed).

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The port's telephone monitor status (rIORTLA.tsm_stat).
<i>Bad Data =</i>	The port's monitored status value (rIORTLA.tsmmed).
<i>Good Data =</i>	Value indicating whether the rIFC_LINE tuple exists. GLSUCCESS implies that it does. DBNO_MATCH implies that it does not.
<i>Logical Index =</i>	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIFC_PORTTYPE tuple.

Table 54 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	The rIORTLA.ds0b field is a none zero value and this implies that the rIDS0B.use field should be set to PCDHP or PCD7SEC. It was found to be set to a different value.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	The rIDS0B.portname field.
<i>Good Data =</i>	The rIDS0B.use field.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIDS0B tuple.

Table 55 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A holding circuit was found to have fields in the rIORTLA tuple that were not null.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 56 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but no rIORTLA tuple with the same port key can be found.
Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the TMbitable[].
<i>Bad Data =</i>	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data =</i>	Test position involved in monitor B&I.
<i>Logical Index =</i>	Port name involved in monitor B&I.
<i>Dump =</i>	None.

Table 57 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	This is an OSPS B-port that is in-service idle.
Possible Error Effect:	OSPS calls cannot be processed to this BST.

Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Good Data</i> =	Service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 58 Error Code: NOTNU

Error Code:	NOTNU
Error Description:	An rIORTLA tuple was found to be used for hairpin or nailup connection but the rIORTLA.nailup field in it was not set to DBYES.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple will be placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 59 Error Code: ODD_DATA (1)

Error Code:	ODD_DATA (1)
Error Description:	Copies of ODD for this port (hunt and porttype) kept in the rIORTLA are corrupted.
Possible Error Effect:	Hunt bit incorrectly set to "hunted" - IDLPRT audit will attempt to idle a non-hunted port in AM routing data. Hunt bit incorrectly set to "non-hunted" - IDLPRT audit will not insure AM routing data for port agrees with module busy/idle status of port porttype bit set to wrong porttype.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Not used (see dump of rIORTLA).
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 60 Error Code: ODD_DATA (2)

Error Code:	ODD_DATA (2)
Error Description:	The circuit information (unit, unit type, circuit type, ckt_idx) contained in the rIORTLA tuple does not agree with the circuit information contained in the rICKTDATA tuple. The key to the rICKTDATA tuple is derived from the rIFC_PORTTYP tuple.
Possible Error Effect:	The circuit cannot be used by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rICKTDATA tuple.

Table 61 Error Code: ODD_DATA (3)

Error Code:	ODD_DATA (3)
Error Description:	The CCS7 information (protocol, cic, pci) contained in the rIORTLA tuple does not agree with the information contained in the rIFC_TRUNK tuple.
Possible Error Effect:	Incorrect port information.
Corrective Action Taken:	The information is updated and the rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	The value of <i>cic</i> in rIORTLA tuple.
<i>Good Data =</i>	The value of <i>pci</i> in rIORTLA tuple.
<i>Logical Index =</i>	Logical key for the rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIFC_TRUNK tuple.

Table 62 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	There is a mismatch between static data relations for this rIORTLA tuple.
Possible Error Effect:	There is no way for the switch to figure out if the port should have this service.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Office personnel should determine which relation is missing and insert it using ODBE. If the port is really supposed to be nailed up, office personnel must populate the rINAILUP tuple for this port using ODBE. If the port should not be nailed up, office personnel should use ODBE to clear the nail-up data from the rIPRTTYP tuple for this port. If this is a modem pool port, the missing relations should be inserted for this rIORTLA. If the rIFC_PORTTYP is dumped, then there is a missing rIFC_TRUNK or rIPP_MEM tuple which needs to be inserted. (If the <i>Good Data</i> is the relation Id for rIPP_MEM, insert that tuple.)
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple or, when <i>Good Data</i> is the relation ID for RLPP_MEM, the key of the missing rIPP_MEM tuple.
<i>Bad Data =</i>	Relation ID for the missing tuple. • RLEQUIPDSL. • RLNAILUP. • RLPORT_GROUP. • RLPSGP_PRT. • RLPSX25LN. Otherwise, the <i>servclass</i> field in rIFC_PORTTYP or, when <i>Good Data</i> is the relation ID for RLPP_MEM, the rIFC_PORTTYP.lp_type.
<i>Good Data =</i>	NA or, if missing rIFC_TRUNK, then this is <i>lp_type</i> in rIFC_PORTTYP or the relation ID RLPP_MEM when missing that tuple.
<i>Logical Index =</i>	Logical key of rIORTLA tuple (port name).
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIFC_PORTTYP tuple if missing rIFC_TRUNK.

Table 63 Error Code: OOS_MATE

Error Code:	OOS_MATE
Error Description:	An OSPS Autoquote DSL incorrectly has a different basic port status from its associated mate analog line.
Possible Error Effect:	If the Autoquote DSL is incorrectly in-service, it will be hunted but it will not be

	possible to get control of the mate analog line. Autoquote call failures will result whenever this pair is hunted. If the Autoquote DSL is incorrectly out-of-service, the pair will never be hunted and therefore will never be used for Autoquote calls.
Corrective Action Taken:	The Autoquote DSL is recovered to the same basic state as its mate analog line.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the Autoquote DSL rIORTLA tuple.
<i>Bad Data =</i>	Port status basic state for the Autoquote DSL.
<i>Good Data =</i>	Port status basic state for the Autoquote mate analog line.
<i>Logical Index =</i>	Port name of the Autoquote DSL.
<i>Dump 1 =</i>	rIORTLA tuple for the Autoquote DSL.
<i>Dump 2 =</i>	rIORTLA tuple for the Autoquote mate analog line.

Table 64 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	This rIORTLA is linked to an invalid list.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	The rIORTLA owner linkage proc_cplr.st_headk.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key (port name) of the rIORTLA tuple.
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 65 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Something is wrong with the logical key of the owner of the rIORTLA tuple.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	The port list of the rIORTLA tuple (rIORTLA.list).
<i>Good Data =</i>	The owner link of the rIORTLA tuple (rIORTLA.prtlst.st_ownk - DBSFPTLST).
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 66 Error Code: PARENT

Error Code:	PARENT
Error Description:	The PORTLA audit is attempting to determine the facility circuit associated with the port. However, the SMIM conversion macro has returned a value of null circuit.
Possible Error Effect:	The port may be unusable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return value from SMIM function.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key to the rISMEST tuple.
<i>Dump =</i>	The rICKTDATA tuple.

Table 67 Error Code: P_UNKSTAT1

Error Code:	P_UNKSTAT1

Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the port status contained by the terminal process system.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA.
<i>Dump</i> =	rIORTLA tuple.

Table 68 Error Code: PHNOOS

Error Code:	PHNOOS
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the PH status.
Possible Error Effect:	Call processing could be affected because the port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	PH status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 69 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the database.
Possible Error Effect:	Manual asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 70 Error Code: PORT_TM

Error Code:	PORT_TM
Error Description:	The terminal maintenance list fields do not agree with the rIORTLA list fields.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The port is placed on the to-be-idled for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	rIORTLA owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 71 Error Code: PORT

Error Code:	PORT
Error Description:	The PORTLA audit attempted to convert a port name to a circuit name, but the PC utility macro returned a null value indicating that the circuit is not equipped.
Possible Error Effect:	The port will not be usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA (0)
<i>Good Data =</i>	Circuit name.
<i>Logical Index =</i>	The key to the rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 72 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	An attempt to convert a port name to a port index has resulted in an indication that the port does not exist (null index).
Possible Error Effect:	The Port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Port name.
<i>Good Data =</i>	NA (0)
<i>Logical Index =</i>	Port index.
<i>Dump =</i>	NA.

Table 73 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The port status in the rIORTLA implies the port should be on a different list than it is actually on.
Possible Error Effect:	Attempts to change a port's mnce status will result in asserts.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	If port had been manually removed from service, it must be removed from service again.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	Port status.
<i>Good Data =</i>	rIORTLA list.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 74 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve an rIGPFTIMER from the key stored in one of the rIORTLA tuple's timer key fields.
Possible Error Effect:	Varies according to conditions including which timer field was being accessed.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data =</i>	The rIGPFTIMER key returned from the read attempt.
<i>Good Data =</i>	The rIGPFTIMER key from the timer key field.
<i>Logical Index =</i>	The rIORTLA tuple logical key (rIORTLA.port).
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIGPFTIMER tuple buffer resulting from the read attempt.

Table 75 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The audit calls macro PSSVCPDLCL() to get the dlci for a frame relay SVC trunk, and PSBADDLCL is returned because the rIFC_PORTTYP or rIFRBC_SVC tuple is not found, or the port is not found within the rIFRBC_SVC tuple.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly.
Corrective Action Taken:	None -- see Manual Action.
Manual Action Required:	1 = Check for, and if missing, insert the missing rIFC_PORTTYP tuple with rIFC_PORTTYP.port.member set to rIORTLA.port and rIFC_PORTTYP.port.module set to Glpcr_id. 2 = Check for, and if missing, insert the missing rIFRBC_SVC tuple with rIFRBC_SVC.gport set to rIFC_PORTTYP.sleevid. 3 = Check for, and if missing, add this port (rIORTLA.port) to an empty rIFRBC_SVC.mport[] in the rIFRBC_SVC tuple keyed by rIFC_PORTTYP.sleevid.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Local "dlci" returned from PS macro call.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIORTLA tuple (rIORTLA.port).
<i>Dump =</i>	rIORTLA tuple.

Table 76 Error Code: SHELF_SLC

Error Code:	SHELF_SLC
Error Description:	An attempt to determine the DCLU unit number from the S96Shelf circuit name has resulted in a return value of null unit.
Possible Error Effect:	The TAU line hardware path will not be torn down and re-established. The TAU line to TSI path will remain connected.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA (0)
<i>Good Data =</i>	Shelf name.
<i>Logical Index =</i>	Return value from SMIM function.
<i>Dump =</i>	NA.

Table 77 Error Code: SHELFREAD

Error Code:	SHELFREAD
Error Description:	Cannot read rIS96SHELF tuple.
Possible Error Effect:	No checking can be done on this port.
Corrective Action Taken:	None.
Manual Action Required:	Check ODD information for this relation.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	RLS96SHELF relation ID.
<i>Logical Index =</i>	Bank name.
<i>Dump =</i>	NA.

Table 78 Error Code: SLAVE_LINK

Error Code:	SLAVE_LINK
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to

	proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	"master" field in TMDlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 79 Error Code: SLC_PORT

Error Code:	SLC_PORT
Error Description:	The PORTLA audit attempted to convert a port name into the corresponding shelf or bank name. The return from the PC function indicated that the bank or shelf does not exist.
Possible Error Effect:	The port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Port name.
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Return value from the PC function.
<i>Dump</i> =	NA.

Table 80 Error Code: SLCLOSTTUP

Error Code:	SLCLOSTTUP
Error Description:	Cannot read the rIS96T1USG tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Skip auditing this tuple and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS96T1USG tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	rIS96T1USG tuple relation ID.
<i>Logical Index</i> =	Bank name.
<i>Dump</i> =	NA.

Table 81 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rISMEST tuple for this rIORTLA cannot be found.
Possible Error Effect:	This rIORTLA cannot be checked for correct FE state.
Corrective Action Taken:	Related audits will recover this port if required.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA (0).
<i>Bad Data</i> =	NA (0).
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 82 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Contents of state.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 83 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The state field indicates idle linkage, but the linkage does not indicate that the rIORTLA is linked to its owner (as indicated by the contents of the list field).
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Contents of owner link.
<i>Good Data</i> =	Contents of list.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 84 Error Code: STATE_LNK

Error Code:	STATE_LNK
Error Description:	The state field indicates that linkage exists but the link fields are empty.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Contents of the bad link field.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 85 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rIORTLA tuple was found linked to an unknown service list. The rIORTLA.list field must be marked as one of the legal service lists (either TMIS for in-service, or one a variety of possible out of service lists) as specified by DMLIST. This error may occur if a new service list has been added by terminal maintenance, but the PORTLA audit was never modified to recognize it.
Possible Error Effect:	The port will be unavailable to call processing and terminal maintenance activities since the associated rIORTLA tuple is in an unknown state.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIORTLA.list field found by the audit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIORTLA tuple (port name).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 86 Error Code: STATUS

Error Code:	STATUS
Error Description:	The port status supplementary information in the rIORTLA is invalid for the relationship between this SM and its mate (rIISMPATH.mate_sm). If rIORTLA.suppl_info is <i>TMDLI</i> , <i>TMSNLI</i> , or <i>TMTMSCONN</i> , the two SMs must not be in an HSM-RSM relationship or an RSM-RSM relationship under the

	same HSM.
Possible Error Effect:	The rIPORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The port status supplementary information (rIPORTLA.suppl_info).
<i>Bad Data =</i>	The relationship between the SM and its mate.
<i>Good Data =</i>	The mate SM number which is served by the inter-SM path (rIISMPATH.mate_sm).
<i>Logical Index =</i>	Logical key of rIPORTLA tuple (rIPORTLA.port).
<i>Dump =</i>	The contents of the rIPORTLA tuple.

Table 87 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	An rIPORTLA tuple contains an invalid telephone status monitoring value (rIPORTLA.tsm_stat). The status is inconsistent with a value derived from the associated rIPCBLA tuple or the rIPCBLA tuple was not accessible.
Possible Error Effect:	Telephone status monitoring may not work correctly for an end user.
Corrective Action Taken:	If the correct status was accessible, the audit corrects the value in the rIPORTLA tuple (rIPORTLA.tsm_stat). Otherwise the port is reinitialized and put on the to-be-idled list to be recovered by terminal maintenance.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	A boolean indicating if the associated status data was accessible.
<i>Bad Data =</i>	The rIPORTLA tuple's telephone monitoring status (rIPORTLA.tsm_stat).
<i>Good Data =</i>	The correct value of the port's monitor status value (rIPORTLA.tsm_stat).
<i>Logical Index =</i>	Logical key of rIPORTLA tuple (rIPORTLA.port).
<i>Dump 1 =</i>	The rIPORTLA tuple.
<i>Dump 2 =</i>	The rIFC_PORTTYPE tuple.
<i>Dump 3 =</i>	The associated rIPCBLA tuple if available.

Table 88 Error Code: STUCK

Error Code:	STUCK
Error Description:	The multipoint analog and digital ports have list fields which indicate in-service but the CPE is out-of-service.
Possible Error Effect:	Multipoint modem pool ports are on an out-of-service CPE. If this CPE should be in-service, the CPE should be power cycled.
Corrective Action Taken:	The analog port is idled for recovery to valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of analog rIPORTLA tuple.
<i>Bad Data =</i>	CPE state.
<i>Good Data =</i>	Analog rIPORTLA list.
<i>Logical Index =</i>	Key of analog rIPORTLA tuple.
<i>Dump 1 =</i>	Analog rIPORTLA tuple.
<i>Dump 2 =</i>	Digital rIPORTLA tuple.
<i>Dump 3 =</i>	rITEILA tuple.

Table 89 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	The analog modem pool port is in service but the corresponding digital one is not.
Possible Error Effect:	Modem pool call could be failing.
Corrective Action Taken:	Both ports are recovered by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPORTLA tuple.
<i>Bad Data =</i>	List the analog port is on.
<i>Good Data =</i>	List the digital port is on.
<i>Logical Index =</i>	Analog port name.

<i>Dump 1</i> =	Analog rIORTLA tuple.
<i>Dump 2</i> =	The corresponding digital rIORTLA tuple.

Table 90 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The D-port is on the TMLTWAIT list but the T321 index is not set (rIPRISTAT.t321_idx = TMTNULLIDX).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The list from rIORTLA tuple (rIORTLA.list).
<i>Bad Data</i> =	The T321 index from the rIPRISTAT tuple (rIPRISTAT.t321_idx).
<i>Good Data</i> =	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	The rIORTLA tuple.
<i>Dump 2</i> =	The rIPRISTAT tuple.

Table 91 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	When the state of the rIORTLA tuple is DBPSIDLE, the wait for restoration bit should be cleared (rIORTLA.rstawait = DBNO).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The <i>rstawait</i> attribute is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The waiting for service acknowledge message flag of the rIORTLA tuple (rIORTLA.srvawait).
<i>Bad Data</i> =	The waiting for restoration flag of the rIORTLA tuple (rIORTLA.rstawait).
<i>Good Data</i> =	The logical port type.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 92 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The B-port has been in a transient state for too long (longer than audit cycle).
Possible Error Effect:	This tuple is in a stuck OOS state and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	port_owner.
<i>Good Data</i> =	d_owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIORTLA tuple for D-port.

Table 93 Error Code: TRUNK_LINE

Error Code:	TRUNK_LINE
Error Description:	The port type for this port claims that it is a trunk but the rCKTDATA tuple that should exist for the trunk is missing.
Possible Error Effect:	There is no way for the audit to recover this port without the rCKTDATA information.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Determine whether the port is supposed to be a trunk. If it is, the add the rCKTDATA tuple using ODBE, or delete the rIORTTYP tuple. If a line, the rIORTTYP tuple needs to be updated to indicate that the port is a line.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.

<i>Bad Data</i> =	Service class of rIPORTLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIPORTLA tuple.

Table 94 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The logical key of the owner of the rIPORTLA tuple points to an unassigned port, but the service class is not unassigned (rIFC_PORTTYP.servclass != RC_PORTUNA).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	The owner link of the rIPORTLA tuple (rIPORTLA.prtlst.st_ownd - DBSFPTLST).
<i>Good Data</i> =	The service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index</i> =	Key of rIPORTLA tuple.
<i>Dump</i> =	The rIPORTLA tuple.

Table 95 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	One of two possible errors: (1) A CCS trunk is found on a CCSINIT or a trunk maintenance list for non-CCS trunks. (2) A non-CCS trunk is found on the CCSINIT or the trunk maintenance continuity list for only CCS trunks.
Possible Error Effect:	Corresponds to Error Description as follows: (1) The routing and terminal allocation subsystem won't hunt this trunk. Trunk maintenance may have difficulty recovering this trunk. (2) Any continuity testing done on a non-CCS trunk will fail.
Corrective Action Taken:	The rIPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status. When the CCS trunk is put on the to-be-idled list, a reset message is sent to the far-end office.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	Contents of list.
<i>Good Data</i> =	Contents of owner link.
<i>Logical Index</i> =	Key of rIPORTLA tuple.
<i>Dump</i> =	rIPORTLA tuple.

PORTLA-D

Software Release: 5E14 and later

1. PORT LINKAGE AREA (PORTLA) AUDIT

Each port has a port linkage area tuple that describes the status and associations of the port. These port linkage areas contain information such as the owner of the port, processes that are waiting to access the port, the status of the port (precut or postcut, and so forth), and a preemption flag that shows whether the port is being preempted.

The port linkage area audit (PORTLA) verifies the data contained in the port linkage areas. When the PORTLA audit is invoked, it checks all ports (both lines and trunks).

2. DATA STRUCTURES

The data structure checked by the PORTLA audit is the port linkage area relation (RLPORTLA). The RLPORTLA relation has one-to-one linkages with the pending history relation (RLPHISTORY) and the process control block linkage area relation (RLPCBLA). It also has owner-member linkage with the coupler relation (RLCPLR) and member linkage with the port list relation (RLPRTLST). The RLPRTLST relation has 16 head cells, each of which owns a list of port linkage areas. Each list indicates a particular port status (for example, in service, out of service, to be idled, and so forth). The port linkage area list item contains an ID that indicates the list to which the port linkage area belongs. The PORTLA audit uses the feature control port name to service class relation (RLFC_PORTTYP) to perform an existence check.

Analog modem pool ports will verify the existence of the related packet switch rIORTLA tuple and its corresponding rIDPB tuple. Additional multipoint modem pool checks have been added for the following data structures:

- The terminal equipment data block relation (RLTEDB).
- The packet switching X.25 level 3 data for DSL's relation (RLPSX25LN).
- The terminal endpoint identifier linkage area relation (RLTEILA).

The PORTLA audit added checks of the following SLC[®] 96 and network management data structures:

- The trunk group selective incoming load control relation (RLTKGSILC).
- The switch maintenance equipment status table relation (RLSMEST).
- The circuit data block common area relation (RLCDBCOM).
- The SLC[®] 96 equipage relation (RLS96EQUIP).
- The SLC[®] 96 T1 usage relation (RLS96T1USG).

Figure 1 shows these data structures.

The PORTLA audit insures the integrity of the port linkage area relation (RLPORTLA). A field in the rIORTLA tuple checks for datalink trunks and lines. If the port is of a particular service class, the datalink bit should be set. If it is not, this check prints an error and then sets the bit. It then continues processing and it does not recover the tuple.

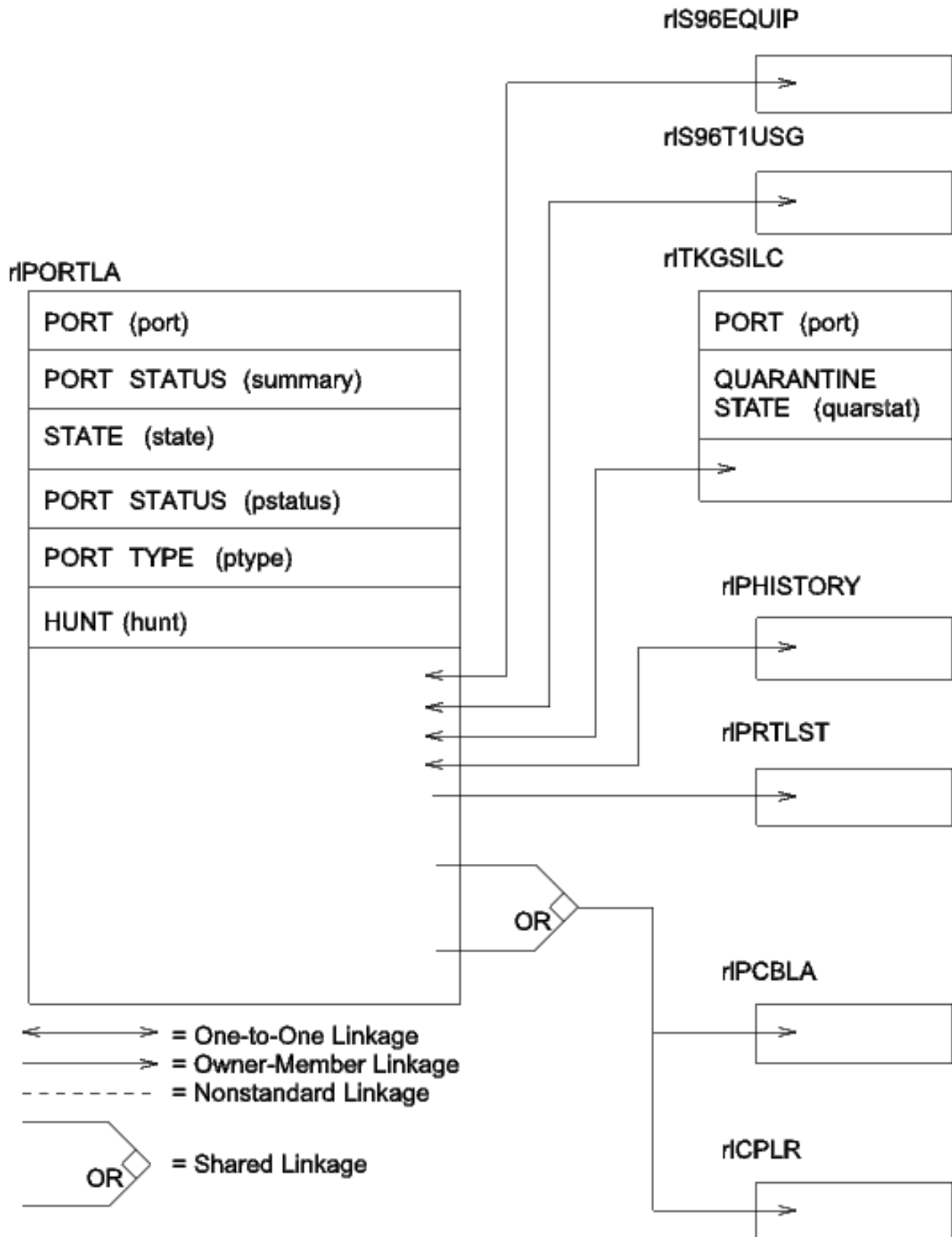


Figure 1 (PORTLA-1) Data Structures**3. AUDIT CHECKS**

The PORTLA audit uses the static driver (DRSTATIC) to step through the RLPORTLA relation.

- (a) Initially, the audit calls a dynamic to static existence check function to perform the following checks.
 - (1) This existence check verifies that, for every dynamic rIORTLA tuple, a static port name to service class tuple (rLFC_PORTTYP) exists. If the static rLFC_PORTTYP tuple does not exist, the extra rIORTLA tuple is deleted (EXTRATUP).
 - (2) For each tuple in RLPORTLA that is associated with a circuit, the existence check verifies that a corresponding tuple exists in the static circuit data relation (RLCKTDATA) for those service classes that should have a tuple (TRUNK_LINE).
- (b) The audit calls a semantic check function to perform the following checks.
 - (1) The audit checks to see that the rIORTLST owning head cell of the port linkage area, as determined from the port linkage area's owner linkage item, agrees with the list item (LIST).
 - (2) The audit checks the cutover status of the office and port. If the office is in a cutover state, the audit checks that the rIORTLA tuple is on the correct list (CUTOVER).
 - (3) The port status (pstatus) item contains the primary maintenance status of the port. The audit verifies that the list to which the port linkage area belongs is consistent with the pstatus item (PSTATUS).
 - (4) The audit verifies that the hunt item in the RLPORTLA relation is consistent with the hunt item in the RLFC_PORTTYP relation (ODD_DATA).
 - (5) The audit verifies that the port type (ptype) item in the RLPORTLA relation is consistent with the port type (porttype) item in the RLFC_PORTTYP relation (ODD_DATA).
 - (6) The audit verifies the datalink item with valid service class (BLINK).
 - (7) If the port is a digital carrier port, the audit verifies that the list to which the port linkage area belongs is consistent with the state of the digital facility interface (DFI) circuit. An active red or yellow alarm should correspond to ports on the facility failure list. Family-of-equipment and to-be-idled lists are also allowed because they are higher priority lists. If both the red and yellow alarms are inactive, the port linkage area can be on any list except the facility failure list (ALARM).
 - (8) Common channel signaling ports are verified to be on valid lists by this audit (WRONGLIST).
 - (9) If the rIORTLA tuple is on a network management list, the corresponding rITKGSILC tuple is read (MISSTUP). Several fields, including the port and quarantine state fields, are checked (PORT_TM).
 - (10) The audit verifies that OSPS Autoquote DSL's have a mate analog rIORTLA tuple (OOS_MATE).
 - (11) The audit verifies that nailup ports have a mate rINAILUP tuple.
 - (12) During linkage checking of the rIORTLA tuple the function AUptlchk() performs point-to point-back checks between the rIORTLA and rIADB relations. If an error is found, an error report is printed to the read-only printer (ROP) and recovery is started.
 - (13) PORTLA audits the linkage between the rIORTLA relation and the RLADB relation for shared directory number (DN) analog. When the PORTLA audit detects the error, the rIORTLA tuple is

linked to the rIADB tuple and the PORTLA audit checks to make sure the sh_analog bit in the RLFC_PORTTYP relation is set to DBYES. If sh_analog bit is not set, the PORTLA audit prints an error on the ROP and returns AUFALSE to start recovery (BDLINK).

- (14) A port linkage area may be linked to one process control block linkage area directly or to one or more process control block linkage areas by means of couplers. Also, it may not be linked to any process control block linkage areas. The port linkage area state item indicates which of these two states exists. The audit checks to see that the state item agrees with the actual linkage (STATE_IDL or STATE_LNK or STATE).
 - (15) In a remote switching module (RSM) environment, if a port is idle, hunted, and in service, the routing and terminal allocation (RTA) subsystem is called to mark the port idle in the hunted port status relations.
 - (16) If the logical port type field of rLFC_PORTTYP indicates that this is a SLC[®] 96 special line, the RLS96T1USG relation is checked to ensure that one dedicated time assignment unit (TAU) is associated with the port (BADTSCOUNT). If it is determined that one does not exist (or more than one exists), the current hardware connection is torn down and a new one is established (SLCLOSTTUP). If an rIS96SHELF tuple is not found, the error is reported (SHELFREAD).
 - (17) For SLC[®] 96 ports, the audit determines whether the port is associated with a 4AB1 or 4AC1 line unit or SLC[®] 96 digital carrier line unit (DCLU). If the port is associated with a DCLU, the key of the rIS96EQUIP tuple is calculated and the tuple is read. The bank status is checked to determine the grid identification, the associated rSMEST tuple is read (SMEST), and the port is conditioned.
 - (18) There is a list of precut status ports contained in the switching module (SM) protected memory. The audit calls a function to determine whether the port associated with the port linkage area being checked is on this list. If it is, the audit checks whether the port linkage area is linked to the rIPRTLST head cell that owns precut status port linkage areas.
 - (19) If the port is an ISDN port, the audit checks the parent circuit(s). It reports an error if the port's maintenance state is inconsistent with the maintenance state of the parent circuit. The audit also checks the validity of family-of-equipment states.
 - (20) This audit also has special checks for modem pool ports. If the port is a modem pool port, the analog and corresponding packet switch ports will be synchronized. For multipoint modem pool ports, the CPE status will be synchronized.
- (c) The PORTLA audit then calls a function to maintain consistency between the monitor busy & idle (B&I) data table, TMbitable[] and the rIPORTLA tuple. The TMbitable[] contains the port and TM process ID information for any ports involved in monitoring using the monitor (B&I) feature.

4. ERROR HANDLING

For all errors except those encountered when the audit is running in the initialization mode, the port linkage area is placed on the to-be-idled list (this list is owned by an rIPRTLST head cell). At the same time, a terminal maintenance process is activated. This process checks the hardware state and the contents of the switch maintenance equipment status table relation (RLSMEST) to determine what the status should be for a given port. For circuits marked out of service (OOS) in rSMEST, the associated ports are given an OOS, family of equipment (FE) status. DFI ports with active red or yellow carrier alarms are given an OOS, facility failure (FAF) status. Ports whose scanning circuitry indicate that they are off-hook are given an OOS, high and wet (HW) status. All other ports are given an in-service status.

For errors encountered when the audit is running in the initialization mode, the port linkage area is linked to the in-service list except for the ports as determined by the service class. Common channel signaling (CCS) trunks are linked to the disabled list and recovered by terminal maintenance (TM) subsystem. D-channels, inter-switch trunks, ISM nailups, X.75 and X75' trunks, and packet switching B-channels are linked to the DSL initialization list and recovered by the TM subsystem. B-channels are linked to the D-channel out-of-service list if the D-channel is out-of-service, else they are linked to the in-service list. In all cases except during initializations, errors are reported by means of output messages.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALARM

Error Code:	ALARM
Error Description:	The list the rIORTLA is on is inconsistent with the state of the digital carrier alarm. Digital carrier ports only.
Possible Error Effect:	If carrier alarm is ON, the port is incorrectly in service and if used by call processing, would cause call failures. If carrier alarm is OFF, the port is incorrectly out of service and thus unavailable for call processing use.
Corrective Action Taken:	The port will be placed on the to-be-idled list for later idling. The software maintenance status of the port will be made consistent with the hardware (carrier) alarm state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	List owner for rIORTLA tuple.
<i>Good Data =</i>	Carrier group alarm status.
<i>Logical Index =</i>	Key of the rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIORMEST tuple.

Table 2 Error Code: ASSOC_EST

Error Code:	ASSOC_EST
Error Description:	An rIORMEST tuple associated with a circuit of a particular type is being recovered because another tuple associated to the circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit will recover the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated to the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORMEST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	rIORMEST tuple.

Table 3 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	An rIORTLA exists for a port whose service class indicates that an rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Service class of the port.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 4 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	A tuple associated with a circuit of a particular type is being recovered because another tuple associated with that circuit was found bad.
Possible Error Effect:	The circuit is unusable.
Corrective Action Taken:	The audit recovers the circuit to its proper in-service or out-of-service state by recovering all of the tuples associated with that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the tuple.
<i>Bad Data =</i>	Relation ID of bad tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump =</i>	Tuple.

Table 5 Error Code: AUOOS

Error Code:	AUOOS
Error Description:	An rSMEST tuple for the circuit under audit or the parent/mate circuit for this circuit was found to be in an invalid state with respect to the information in the MRA deferred work recovery queue. The qual_2 state of SMAUOOS is used to indicate when audits or SI has recovered a circuit, parent or mate to the MRA queue. Note, refer to <i>Good Data</i> for details.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly by MRA.
Corrective Action Taken:	The audit will attempt to recovery the circuit using the MRA queue and/or leave it in a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	qual_2 field in the SMEST tuple.
<i>Good Data =</i>	Return value from AUMRARET (au/hdr/AUglobckt.L).
	0 = Circuit was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	2 = Parent/mate was found in the MRA queue, placed there by a non-audit process. Try to clean up SMEST and place it on the queue.
	4 = Circuit is marked AUOOS but not in queue, parent is not in queue, circuit/parent/mate is not MRA BUSY.
<i>Logical Index =</i>	Logical key of the tuple.
<i>Dump 1 =</i>	rSMEST tuple.
<i>Dump 2 =</i>	parent/mate rSMEST tuple.

Table 6 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	An rDS0B tuple is involved in a hairpin connection but the rPORTLA.dsob field does not have the correct value.
Possible Error Effect:	Call processing will not be able to use that rPORTLA tuple.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rPORTLA tuple.
<i>Bad Data =</i>	rDS0B.use field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The DS0B key.
<i>Dump 1 =</i>	rDS0B tuple.

<i>Dump 2 =</i>	rIORTLA tuple.

Table 7 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	The rIDS0B tuple is involved in a hairpin or nailup connection but the rIDS0B.portname field tuple is not matching the rNAILUP.port1 field.
Possible Error Effect:	There will be no hairpin or nailup connection established since a wrong port is being used.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDS0B tuple.
<i>Bad Data =</i>	The rIDS0B tuple key.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rIDS0B tuple.
<i>Dump 1 =</i>	rIDS0B tuple.
<i>Dump 2 =</i>	rIORTLA tuple.

Table 8 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	An rIORTLA tuple contains the key of an rIGPFTIMER tuple in one of it's timer key fields but the rIGPFTIMER tuple is either not linked back (rIGPFTIMER.udata contains a different relation ID or key from the owning tuple) or the timer type (rIGPFTIMER.timer_type) does not have the expected value for the timer key field being used.
Possible Error Effect:	Varies according to conditions including the rIORTLA usage and which timer key field is being checked.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data =</i>	The relation ID stored in rIGPFTIMER.udata.
<i>Good Data =</i>	The tuple key stored in rIGPFTIMER.udata.
<i>Logical Index =</i>	The rIORTLA tuple logical key (rIORTLA.port).
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIGPFTIMER tuple.

Table 9 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	For CCS7 trunks, there are four possible conditions for which this error is reported. (1) There is a reset in progress, but the <i>reset_inprog</i> bit in PORTLA is not set to DBYES. (2) There is a reset in progress with the <i>reset_inprog</i> bit in PORTLA set to DBYES, but the port is not owned. (3) There is no reset in progress, but the <i>reset_inprog</i> bit in PORTLA is set to DBYES. (4) There is no entry for this port, but the <i>reset_inprog</i> bit in PORTLA is set to DBYES. For CCS6 trunks, the following fields in PORTLA are not set correctly. These fields include <i>protocol</i>, <i>cic</i>, <i>pci</i>, <i>reset_inprog</i>.

Possible Error Effect:	Incorrect port usage.
Corrective Action Taken:	If the problem is in regards to the <i>reset_inprog</i> bit, then the bit will be set to its correct value. If the error is in regards to the CCS6 trunk, then the following will occur. <i>protocol</i> will be set to DBCCIS6, <i>cic</i> will be set to CCNULCIC, <i>pci</i> will be set to CCNULPCI, and <i>reset_inprog</i> will be set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple for CCS6 trunks. Error flag value for CCS7 trunks.
<i>Bad Data</i> =	Value of in_sigtype from rIFC_TRUNK for CCS6 trunks. Value of mtype returned from TM for CCS7 trunks.
<i>Good Data</i> =	Value of out_sigtype from rIFC_TRUNK for CCS6 trunks. Value of type returned from TM for CCS7 trunks.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 10 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	One of two possible errors: (1) The port is on the TBI (to be idled) list and is still waiting to be run but the tbi_flag is set, indicating that the job is running. (2) The port is not on the TBI list and the flag is set.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	Corresponds to Error Description as follows: (1) The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status. (2) The tbi_flag is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	When <i>Good Data</i> is 0, the to-be-idled flag of the rIORTLA tuple (rIORTLA.tbi_flag). When <i>Good Data</i> is 1, the return value from TMisqueued() function call.
<i>Good Data</i> =	0 or 1.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 11 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The D-port is linked to the unassigned list but the service class taken from static data does not indicate that the port should be on unassigned list.
Possible Error Effect:	The rIORTLA is unavailable for call processing.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to the proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Owner of rIORTLA tuple.
<i>Good Data</i> =	Service class of port.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 12 Error Code: BADTSCOUNT

Error Code:	BADTSCOUNT

Error Description:	A SLC [®] 96 time slot was found to have more than one time assignment unit assigned to it.
Possible Error Effect:	Possible origination problems for some mode II SLC [®] 96 special lines.
Corrective Action Taken:	Reconstruct the rIS96T1USG tuple with one time assignment unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS96T1USG tuple.
<i>Bad Data</i> =	Count of T1TS lines for this time assignment unit.
<i>Good Data</i> =	Number of time slots allowed (1).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIS96T1USG tuple.

Table 13 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An rIDS0B tuple was used for hairpin or nailup connection but the rIDS0B.use field was not set to indicate that.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDS0B tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	rIDS0B tuple.
<i>Dump 2</i> =	rIPORTLA tuple.

Table 14 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The rIPORTLA tuple is on the wrong list. The port could be stuck on an out-of-service list when it should be in-service or on an is-service list when it can not be used because of the parent circuit. There are two possible errors: (1) The port is not OOS-FE, but the parents circuits are OOS. (2) The port is OOS-FE, but the parents circuits are not OOS. If the port is on a IDCU TR08 the audit will dump the shelf/bank that was checked. For analog ports on an IDCU TR303, one DS1 will be dumped but all DS1's on the RT303 will be checked.
Possible Error Effect:	Either the port will be used for a call which cannot complete because of out-of-service equipment or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPORTLA tuple.
<i>Bad Data</i> =	List owner of rIPORTLA tuple.
<i>Good Data</i> =	Logical port type.
<i>Logical Index</i> =	Logical key of rIPORTLA tuple.
<i>Dump 1</i> =	rIPORTLA tuple.
<i>Dump 2</i> =	rSMEST tuple with port as key.
<i>Dump 3</i> =	rSMEST tuple for PH if available.
<i>Dump 4</i> =	rSMEST tuple for PSU if available.

Table 15 Error Code: BDLINK

Error Code:	BDLINK

Error Description:	The rPORTLA was found linked to the rIADB tuple when static data (rIFC_PORTTYP) did not have the shared analog bit set.
Possible Error Effect:	Call processing cannot access the port correctly.
Corrective Action Taken:	The rPORTLA tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPORTLA tuple.
<i>Bad Data</i> =	Linkage field to the rIADB tuple.
<i>Good Data</i> =	Shared analog bit in rIFC_PORTTYP tuple.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 16 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The packet switching modem pool port is not bound (rTEDB tuple to rTEILA tuple).
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEDB tuple.
<i>Bad Data</i> =	Key to rTEILA tuple (0 means no rTEILA tuple).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEDB tuple.
<i>Dump</i> =	rTEDB tuple.

Table 17 Error Code: BINDWRONG

Error Code:	BINDWRONG
Error Description:	The packet switching modem pool port is not bound to the correct rTEDB tuple.
Possible Error Effect:	The digital terminal will not be able to originate/terminate modem pool calls. Call processing will fail.
Corrective Action Taken:	The analog port will be recovered to a valid state for the digital port state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA tuple.
<i>Bad Data</i> =	TEDB key from rTEILA tuple.
<i>Good Data</i> =	Key of rTEDB tuple (logical port).
<i>Logical Index</i> =	Logical key of rTEILA tuple.
<i>Dump</i> =	rTEILA tuple.

Table 18 Error Code: BLINK

Error Code:	BLINK
Error Description:	The rPORTLA service class was of the datalink type but the datalink field was not set to DBYES.
Possible Error Effect:	The line or trunk would not be treated as a datalink for removes/restores.
Corrective Action Taken:	The datalink bit will be set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPORTLA tuple.
<i>Bad Data</i> =	datalink field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 19 Error Code: BTRANCK

Error Code:	BTRANCK
Error Description:	The associated D-port is out-of-service and this B-port is not on a list to indicate the unavailability of the D-port. This is considered to be valid for a short transient period. This error is not reported for at least 5 minutes.
Possible Error Effect:	Call processing cannot complete service on this port.

Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	D-port owner.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	D-channel rIORTLA tuple.
<i>Dump 3</i> =	rIPRISTAT tuple.
<i>Dump 4</i> =	when the lp_type is SVC, rIDPB tuple.

Table 20 Error Code: CADNPORT

Error Code:	CADNPORT
Error Description:	The port is either on the CADN list when the <i>cadn</i> field in FCPORTTYP is set to TMIIS or the port is not on the CADN list when the <i>cadn</i> field in FCPORTTYP is not TMIIS. For both of these cases, there error is only reported if the port is not owned.
Possible Error Effect:	The port may not be able to be used by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	<i>cadn</i> field in rIFC_PORTTYP tuple.
<i>Good Data</i> =	<i>list</i> field in rIORTLA tuple.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 21 Error Code: CKTDATA (1)

Error Code:	CKTDATA (1)
Error Description:	The PORTLA audit is attempting to read a nonexistent rICKTDATA tuple. The key to the rICKTDATA tuple is derived from the rIFC_PORTTYP tuple.
Possible Error Effect:	Call processing could be affected.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 22 Error Code: CKTDATA (2)

Error Code:	CKTDATA (2)
Error Description:	The rICKTDATA tuple is missing from the database. The key used to read the tuple was derived from the path2_ds0.ds1name in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rICKTDATA tuple.
<i>Dump 1</i> =	First rINAILUP tuple
<i>Dump 2</i> =	Second rINAILUP tuple

Table 23 Error Code: CUTOVER

Error Code:	CUTOVER
Error Description:	A port is on the wrong list based on the office's or this ports cutover status.
Possible Error Effect:	Pre-cut ports could incorrectly be receiving service from this office or post-cut (active) ports could be denied service.
Corrective Action Taken:	Port is conditioned and returned to correct list (maintenance status).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	List owner.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIORTLA tuple.
<i>Dump =</i>	rIORTLA tuple.

Table 24 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit failed a read of the RLDSLEQUIP or RLISMPATH relation associated with this RLPORTLA.
Possible Error Effect:	There could be an ODD inconsistency.
Corrective Action Taken:	The RLPORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	If the error address is null, check the RLDSLEQUIP relation. If the error address is not null, check the RLISMPATH relation. The key used is printed in <i>Bad Data</i> .
Dump Description:	
<i>Error Address =</i>	NA if the failed read is rIDSLEQUIP. If it is rIISMPATH, the port status supplementary information (rIORTLA.suppl_info).
<i>Bad Data =</i>	If the failed read is rIDSLEQUIP, DSL GROUP. If it is rIISMPATH, the index to the port (rIFC PORTTYP.port indx).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name (rIORTLA.port).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 25 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The CALEA bridge process (CABR) in-progress (rIORTLA.calbrgip) attribute in the rIORTLA tuple is set to indicate that there is a CABR bridge inprogress, but the CABR process number (rIORTLA.cabr_procn) and the CABR process uniqueness (rIORTLA.uniq) point to a dead process.
Possible Error Effect:	May cause asserts or interrupts when the CABR process releases the bridge associated with the port under audit.
Corrective Action Taken:	The CABR in-progress (rIORTLA.calbrgip) attribute will be set to indicate that there is no an active CABR bridge on the call and the CABR process number (rIORTLA.cabr_procn) will be set to null (OSnullpid.procn).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	CABR in-progress data (rIORTLA.calbrgip).
<i>Good Data =</i>	CABR process number (rIORTLA.cabr_procn).
<i>Logical Index =</i>	Logical key (rIORTLA.port) or the rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIOSPCB tuple.

Table 26 Error Code: DLI_FE

Error Code:	DLI FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the DLI status or NLI status on the major ONTC side. The inter-SM port indicated out-of-service family of equipment but the DLI or all NLIs on the major ONTC side is not out-of-service.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to

	proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMqcmhw stat).
<i>Good Data</i> =	The DLI status in data delivery.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 27 Error Code: DLOSTTUP

Error Code:	DLOSTTUP
Error Description:	The mate D-port tuple does not exist for this rIORTLA. A major problem but recovery will be completed in the PORTTYP audit.
Possible Error Effect:	Unable to complete packet switching calls.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	portkey of packet rIORTLA tuple.
<i>Good Data</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 28 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	An rIDPB does not exist for the packet switching port for this analog modem pool port.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate (DPB/PORTTYP) audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of packet switch rIORTLA tuple.
<i>Bad Data</i> =	Key of analog rIORTLA.
<i>Good Data</i> =	Key of rIDPB.
<i>Logical Index</i> =	Key of packet switch rIORTLA tuple.
<i>Dump</i> =	Packet switch rIORTLA tuple.

Table 29 Error Code: DSL_GRP2

Error Code:	DSL_GRP2
Error Description:	The rIDSLEQUIP tuple read failed.
Possible Error Effect:	The rIORTLA tuple has been garbaged, which caused the reading of the wrong rIDSLEQUIP tuple or the rIDSLEQUIP tuple has been garbaged which would give an incorrect state for the rIDSLEQUIP tuple.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	Check ODD to determine if the port is really a member of a DSL. If it is not, verify that the lp_type and class of service fields in the RLPORTLA relation are correct, and do not indicate that the port belongs to a DSL. If none of these errors is found, and the other audits do not clean up the problem, seek help.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 30 Error Code: DSLGROUP

Error Code:	DSLGROUP
Error Description:	The packet switching port list does not match the status of its DSL group. There

	are two possible errors:
	(1) The port is not OOS-FE, but the DSL group is OOS.
	(2) The port is OOS-FE, but the DSL group is not OOS.
Possible Error Effect:	Either the port will be used for a call which cannot complete because the DSL group is out-of-service or the port is out-of-service when it could be used for a call.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	DSL group.
<i>Good Data</i> =	Logical port type.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rSMEST tuple for the PH (if available).
<i>Dump 3</i> =	rSMEST tuple for the PSU (if available).

Table 31 Error Code: DS0B_LINK

Error Code:	DS0B_LINK
Error Description:	The field rIDS0B.use was set to PCD7SEC but the rIDS0B.portname field was not equal to rIORTLA.port field.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIORTLA.port field.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 32 Error Code: DS0B_LOST

Error Code:	DS0B_LOST
Error Description:	The audit tried to read the rIDS0B tuple for a given rIORTLA.ds0b field but the tuple was missing from the database.
Possible Error Effect:	The call processing could not use that DS0B time slot.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the rIDS0B tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIDS0B tuple.
<i>Dump 1</i> =	First rINAILUP tuple.
<i>Dump 2</i> =	Second rINAILUP tuple.

Table 33 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rIORTLA exists for a port whose service class indicates that a rIORTLA should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 34 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The grid group member number derived from the rIGSDB tuple does not agree with the equivalent number (grpname) from the rIORTLA tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The grpname in the rIORTLA tuple will be set to the group member number derived from the rIGSDB tuple in an attempt to recover consistency.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIORTLA unit type.
<i>Bad Data</i> =	rIORTLA tuple grpname.
<i>Good Data</i> =	grid group member from rIGSDB.
<i>Logical Index</i> =	Logical key of the rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 35 Error Code: GPORT

Error Code:	GPORT
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbitable[] but the monitoring bit in the rIORTLA tuple is not set.
Possible Error Effect:	Calls on this port may not be monitored.
Corrective Action Taken:	The monitoring bit is set to indicate that monitoring is active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].
<i>Bad Data</i> =	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	rIORTLA tuple involved in monitor B&I.

Table 36 Error Code: GRP2INFO

Error Code:	GRP2INFO
Error Description:	A port was found with an invalid trunk group number (<i>grpnum</i>).
Possible Error Effect:	This port may be unusable for call traffic.
Corrective Action Taken:	The <i>grpnum</i> field corrected the value as indicated in the corresponding FC_TRUNK tuple, or changed to a null value when no corresponding FC_TRUNK tuple could be found.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad rIORTLA tuple.
<i>Bad Data</i> =	Invalid group number (rIORTLA.grpnum).
<i>Good Data</i> =	Good value of <i>grpnum</i> (rIFC_TRUNK.tgroup) or null value.
<i>Logical Index</i> =	Port name involved.
<i>Dump 1</i> =	rIORTLA tuple involved.
<i>Dump 2</i> =	rIFC_TRUNK tuple involved if present.

Table 37 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The rIPRISTAT tuple cannot be found from the rIORTLA tuple.
Possible Error Effect:	The B-port cannot be checked against the D-port.
Corrective Action Taken:	Report the error and return no corrective action taken.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Key of rIORTLA tuple.
<i>Bad Data</i> =	Pointer to the rIPRISTAT tuple.
<i>Good Data</i> =	The logical port type.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	The rIORTLA tuple.

Table 38 Error Code: INT_DATA_BAD

Error Code:	INT_DATA_BAD
Error Description:	The value of rIORTLA.dlci is not as expected when <i>Good Data</i> is 1, the logical port type is a SVC trunk and it should match the value obtained when the PS macro is called when <i>Good Data</i> is 2, the value of rIORTLA.dlci should be zero.
Possible Error Effect:	The port may be sending messages to the wrong frame relay.
Corrective Action Taken:	Reset the rIORTLA.dlci as expected.
Manual Action Required:	None
Dump Description:	
<i>Error Address</i> =	When <i>Good Data</i> is 1, the expected value of the rIORTLA.dlci when <i>Good Data</i> is 2, the logical port type (rIFC_PORTTYP.lp_type).
<i>Bad Data</i> =	the rIORTLA.dlci.
<i>Good Data</i> =	1 or 2.
<i>Logical Index</i> =	Key of rIORTLA tuple (rIORTLA.port).
<i>Dump</i> =	rIORTLA tuple.

Table 39 Error Code: INV

Error Code:	INV
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data</i> =	Value found in that address.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump</i> =	The whole array AUs96inhp[].

Table 40 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The status of the inter-SM nailup in the rIORTLA is not valid.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status supplementary information.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 41 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	This rIORTLA is linked to an invalid list based on the logical port type.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The PORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.

<i>Bad Data</i> =	rIPORTLA list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIPORTLA.
<i>Dump</i> =	rIPORTLA tuple.

Table 42 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The rIPORTLA tuple is linked to an invalid list. The tuple is associated with a D-channel, and the logical port type indicates remote protocol test access (rIFC_PORTTYP.lp_type == PCRDTA_D or PCRDTA_UD), but the port is on the level 2 or level 3 error port status list (rIPORTLA.list == TMLTLV2ERR or TMLTLV3ERR). This is an illegal condition.
Possible Error Effect:	Remote digital test access (RDTA) actions and operations can not take place.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The port status list (rIPORTLA.list) which was found.
<i>Bad Data</i> =	The key to the rIPRISTAT tuple (rIPRISTAT.prigrp)
<i>Good Data</i> =	Return code from validation of the PRI D-port status.
<i>Logical Index</i> =	The key of the rIPORTLA tuple (port).
<i>Dump 1</i> =	The contents of the rIPORTLA tuple.
<i>Dump 2</i> =	If this dump exists, it holds the contents of the D-channel's rIPORTLA tuple.

Table 43 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	An entry in TMbitable[] is not empty but the module attribute in bigport is invalid and indicates garbage in the table.
Possible Error Effect:	Invalid entry is in TMbitable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbitable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbitable[].
<i>Bad Data</i> =	Module in TMbitable[].bigport.module.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	None.

Table 44 Error Code: ISDNTUP

Error Code:	ISDNTUP
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rIACT_SPR tuple is not in the database.
Possible Error Effect:	Many asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rIACT_SPR tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIACT_SPR tuple.
<i>Dump</i> =	NA.

Table 45 Error Code: LIST

Error Code:	LIST
Error Description:	The list to which the portla should be linked does not agree with the list field used by the call processing and terminal maintenance programs.

Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes in error.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for late idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA list.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 46 Error Code: LNK_FE

Error Code:	LNK_FE
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMS or NCT link status.
Possible Error Effect:	The port is OOS and not available for call processing.
Corrective Action Taken:	The rIORTLA tuple is recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The temporary status of links on the major ONTC (TMcmhw_stat).
<i>Bad Data</i> =	The summary status of links on the major ONTC (TMgcmhw_stat).
<i>Good Data</i> =	The TMS link status in data delivery.
<i>Logical Index</i> =	Logical key to rIORTLA (port name).
<i>Dump</i> =	The contents of the rIORTLA tuple.

Table 47 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The rCKTDATA tuple is missing from the database. The key used to read the tuple was derived from the path1 ds0.ds1name in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that circuit.
Corrective Action Taken:	The nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rINAILUP tuple.
<i>Bad Data</i> =	Loop number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rCKTDATA tuple.
<i>Dump 1</i> =	First rINAILUP tuple
<i>Dump 2</i> =	Second rINAILUP tuple

Table 48 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	An rIORTLA tuple is missing from the database. The port used to read the tuple was taken from the rIORTLA.port1 or rIORTLA.port3_member field in the rINAILUP tuple.
Possible Error Effect:	Call processing will not be able to use that port.
Corrective Action Taken:	The hairpin or nailup connection will be idled out and all the ports involved in the connection will be recovered to the to-be-idled list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The key used to read the missing rIORTLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	First rINAILUP tuple.
<i>Dump 2</i> =	Second rINAILUP tuple.

Table 49 Error Code: LOSTTUP

Error Code:	LOSTTUP

Error Description:	The packet switching modem pool port does not have a mate rTEDB or rTEILA tuple.
Possible Error Effect:	The packet portion of the modem pool call cannot complete.
Corrective Action Taken:	None. The appropriate audit will insert the missing tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA.
<i>Bad Data</i> =	Key to missing tuple (logical port - rTEDB or party - rTEILA).
<i>Good Data</i> =	Relation ID of missing tuple (rTEDB or rTEILA).
<i>Logical Index</i> =	Key of tuple with missing link.
<i>Dump</i> =	rIORTLA tuple (if missing rTEDB) or rTEDB tuple (if missing rTEILA).

Table 50 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The status of the inter-SM nailup in the portla does not agree with the "pstat" field in the TMdlt_data[] array.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 51 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The status of the inter-SM nailup in the portla does not agree with the TMdlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	"master" field in TMdlt_data[].
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 52 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	PORTLA audit is attempting to read a nonexistent tuple. There is a bad linkage between rIORTLA and rTKGSILC.
Possible Error Effect:	None. Call processing could be affected.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Port owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 53 Error Code: NOIDLELIST

Error Code:	NOIDLELIST

Error Description:	The rIORTLA is on an invalid list.
Possible Error Effect:	The port may be lost for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	rIORTLA owner.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rIORTLA tuple.

Table 54 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	An rIORTLA contains an invalid telephone status monitored value (rIORTLA.tsmcd). If the port is a trunk (rIFC_PORTTYPE.porttype = TM_TRK), either the port is marked as being monitored (rIORTLA.tsmcd != DBNO) or status is not idle (rIORTLA.tsm_stat != TSM_IDLE). If the port is a line, the monitored status is inconsistent with the ODD (no rIFC_LINE tuple exists and rIORTLA.tsmcd != DBNO or the rIFC_LINE tuple exists and rIORTLA.tsmcd != rIFC_LINE.tsmcd).
Possible Error Effect:	If the port is a trunk, asserts from telephone status monitoring code are likely. If the port is a line, telephone status monitoring may not work correctly for an end user.
Corrective Action Taken:	The audit corrects the value(s) in the rIORTLA tuple (rIORTLA.tsmcd = DBNO and rIORTLA.tsm_stat = TSM_IDLE for a trunk or rIORTLA.tsmcd = rIFC_LINE.tsmcd).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The port's telephone monitor status (rIORTLA.tsm_stat).
<i>Bad Data</i> =	The port's monitored status value (rIORTLA.tsmcd).
<i>Good Data</i> =	Value indicating whether the rIFC_LINE tuple exists. GLSUCCESS implies that it does. DBNO MATCH implies that it does not.
<i>Logical Index</i> =	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump 1</i> =	The rIORTLA tuple.
<i>Dump 2</i> =	The rIFC_PORTTYPE tuple.

Table 55 Error Code: NONEPROC

Error Code:	NONEPROC
Error Description:	The rIORTLA.ds0b field is a none zero value and this implies that the rIDS0B.use field should be set to PCDHP or PCD7SEC. It was found to be set to a different value.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The rIDS0B.portname field.
<i>Good Data</i> =	The rIDS0B.use field.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 56 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A holding circuit was found to have fields in the rIORTLA tuple that were not null.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA tuple is deleted.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 57 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	A port that is being monitored using the monitor B&I feature has been entered into the TMbtable[] but no rIORTLA tuple with the same port key can be found.
Possible Error Effect:	Invalid entry is in TMbtable[] and monitor B&I status is unknown.
Corrective Action Taken:	The TMbtable[] entry is cleared and a message is sent to the AM monitor process to indicate that the entry has been cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TMbtable[].
<i>Bad Data</i> =	TM process in AM controlling the port involved in monitor B&I.
<i>Good Data</i> =	Test position involved in monitor B&I.
<i>Logical Index</i> =	Port name involved in monitor B&I.
<i>Dump</i> =	None.

Table 58 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	This is an OSPS B-port that is in-service idle.
Possible Error Effect:	OSPS calls cannot be processed to this BST.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Logical port type (rIFC_PORTTYP.lp_type).
<i>Good Data</i> =	Service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 59 Error Code: NOTNU

Error Code:	NOTNU
Error Description:	An rIORTLA tuple was found to be used for hairpin or nailup connection but the rIORTLA.nailup field in it was not set to DBYES.
Possible Error Effect:	The nailup or hairpin connection will not be validly linked.
Corrective Action Taken:	The rIORTLA tuple will be placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIDS0B tuple.

Table 60 Error Code: ODD_DATA (1)

Error Code:	ODD_DATA (1)
Error Description:	Copies of ODD for this port (hunt and porttype) kept in the rIORTLA are corrupted.
Possible Error Effect:	Hunt bit incorrectly set to "hunted" - IDLPRT audit will attempt to idle a non-hunted port in AM routing data. Hunt bit incorrectly set to "non-hunted" - IDLPRT audit will not insure AM routing data for port agrees with module

	busy/idle status of port porttype bit set to wrong porttype.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Not used (see dump of rIORTLA).
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 61 Error Code: ODD_DATA (2)

Error Code:	ODD_DATA (2)
Error Description:	The circuit information (unit, unit type, circuit type, ckt_idx) contained in the rIORTLA tuple does not agree with the circuit information contained in the rICKTDATA tuple. The key to the rICKTDATA tuple is derived from the rIFC_PORTTYP tuple.
Possible Error Effect:	The circuit cannot be used by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	1.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rICKTDATA tuple.

Table 62 Error Code: ODD_DATA (3)

Error Code:	ODD_DATA (3)
Error Description:	The CCS7 information (protocol, cic, pci) contained in the rIORTLA tuple does not agree with the information contained in the rIFC_TRUNK tuple.
Possible Error Effect:	Incorrect port information.
Corrective Action Taken:	The information is updated and the rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	The value of <i>cic</i> in rIORTLA tuple.
<i>Good Data</i> =	The value of <i>pci</i> in rIORTLA tuple.
<i>Logical Index</i> =	Logical key for the rIORTLA tuple.
<i>Dump 1</i> =	rIORTLA tuple.
<i>Dump 2</i> =	rIFC_TRUNK tuple.

Table 63 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	There is a mismatch between static data relations for this rIORTLA tuple.
Possible Error Effect:	There is no way for the switch to figure out if the port should have this service.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Office personnel should determine which relation is missing and insert it using ODBE. If the port is really supposed to be nailed up, office personnel must populate the rINAILUP tuple for this port using ODBE. If the port should not be nailed up, office personnel should use ODBE to clear the nail-up data from the rIPRTTYP tuple for this port. If this is a modem pool port, the missing relations should be inserted for this rIORTLA. If the rIFC_PORTTYP is dumped, then there is a missing rIFC_TRUNK or rIPP_MEM tuple which needs to be inserted. (If the <i>Good Data</i> is the relation Id for rIPP_MEM, insert that tuple.)
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple or, when <i>Good Data</i> is the relation ID for RLPP_MEM, the key of the missing rIPP_MEM tuple.

<i>Bad Data</i> =	Relation ID for the missing tuple. • RLEQUIPDSL. • RLNAILUP. • RLPORT_GROUP. • RLPSGP_PRT. • RLPSX25LN. Otherwise, the <i>servclass</i> field in rIFC_PORTTYP or, when <i>Good Data</i> is the relation ID for RLPP_MEM, the rIFC_PORTTYP.lp_type.
<i>Good Data</i> =	NA or, if missing rIFC_TRUNK, then this is <i>lp_type</i> in rIFC_PORTTYP or the relation ID RLPP_MEM when missing that tuple.
<i>Logical Index</i> =	Logical key of rIPORTLA tuple (port name).
<i>Dump 1</i> =	rIPORTLA tuple.
<i>Dump 2</i> =	rIFC_PORTTYP tuple if missing rIFC_TRUNK.

Table 64 Error Code: OOS_MATE

Error Code:	OOS_MATE
Error Description:	An OSPS Autoquote DSL incorrectly has a different basic port status from its associated mate analog line.
Possible Error Effect:	If the Autoquote DSL is incorrectly in-service, it will be hunted but it will not be possible to get control of the mate analog line. Autoquote call failures will result whenever this pair is hunted. If the Autoquote DSL is incorrectly out-of-service, the pair will never be hunted and therefore will never be used for Autoquote calls.
Corrective Action Taken:	The Autoquote DSL is recovered to the same basic state as its mate analog line.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the Autoquote DSL rIPORTLA tuple.
<i>Bad Data</i> =	Port status basic state for the Autoquote DSL.
<i>Good Data</i> =	Port status basic state for the Autoquote mate analog line.
<i>Logical Index</i> =	Port name of the Autoquote DSL.
<i>Dump 1</i> =	rIPORTLA tuple for the Autoquote DSL.
<i>Dump 2</i> =	rIPORTLA tuple for the Autoquote mate analog line.

Table 65 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	This rIPORTLA is linked to an invalid list.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its association with processes, in error.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPORTLA tuple.
<i>Bad Data</i> =	The rIPORTLA owner linkage proc_cplr.st_headk.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key (port name) of the rIPORTLA tuple.
<i>Dump</i> =	The contents of the rIPORTLA tuple.

Table 66 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Something is wrong with the logical key of the owner of the rIPORTLA tuple.
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIPORTLA tuple is placed on the to-be-idled list for later idling and

	restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	The port list of the rIORTLA tuple (rIORTLA.list).
<i>Good Data =</i>	The owner link of the rIORTLA tuple (rIORTLA.prtlst.st_ownk - DBSFPRTLST).
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 67 Error Code: PARENT

Error Code:	PARENT
Error Description:	The PORTLA audit is attempting to determine the facility circuit associated with the port. However, the SMIM conversion macro has returned a value of null circuit.
Possible Error Effect:	The port may be unusable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return value from SMIM function.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key to the rI SMEST tuple.
<i>Dump =</i>	The rI CKTDATA tuple.

Table 68 Error Code: P_UNKSTAT1

Error Code:	P_UNKSTAT1
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the port status contained by the terminal process system.
Possible Error Effect:	Packet switching inter-SM routing could be failing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Port status.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rIORTLA.
<i>Dump =</i>	rIORTLA tuple.

Table 69 Error Code: PHNOOS

Error Code:	PHNOOS
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the PH status.
Possible Error Effect:	Call processing could be affected because the port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	PH status.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 70 Error Code: PICB

Error Code:	PICB
Error Description:	Circuits have been found in the RLCKTDATA relation associated with the PICB specified in the logical index field of this dump, but the rPICB tuple is not in the

	database.
Possible Error Effect:	Manual asserts and single process purges may be taken due to the inconsistency in static data.
Corrective Action Taken:	None.
Manual Action Required:	The rPICB tuple specified by the logical index of the dump should be inserted into the database.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rPICB tuple.
<i>Dump</i> =	NA.

Table 71 Error Code: PORT_TM

Error Code:	PORT_TM
Error Description:	The terminal maintenance list fields do not agree with the rPORTLA list fields.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The port is placed on the to-be-idled for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPORTLA tuple.
<i>Bad Data</i> =	rPORTLA list.
<i>Good Data</i> =	rPORTLA owner.
<i>Logical Index</i> =	Logical key of rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 72 Error Code: PORT

Error Code:	PORT
Error Description:	The PORTLA audit attempted to convert a port name to a circuit name, but the PC utility macro returned a null value indicating that the circuit is not equipped.
Possible Error Effect:	The port will not be usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA (0)
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	The key to the rPORTLA tuple.
<i>Dump</i> =	The rPORTLA tuple.

Table 73 Error Code: PORTNAME

Error Code:	PORTNAME
Error Description:	An attempt to convert a port name to a port index has resulted in an indication that the port does not exist (null index).
Possible Error Effect:	The Port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Port name.
<i>Good Data</i> =	NA (0)
<i>Logical Index</i> =	Port index.
<i>Dump</i> =	NA.

Table 74 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The port status in the rPORTLA implies the port should be on a different list than it is actually on.
Possible Error Effect:	Attempts to change a port's mtce status will result in asserts.

Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	If port had been manually removed from service, it must be removed from service again.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Port status.
<i>Good Data</i> =	rIORTLA list.
<i>Logical Index</i> =	Logical key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

Table 75 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve an rIGPFTIMER from the key stored in one of the rIORTLA tuple's timer key fields.
Possible Error Effect:	Varies according to conditions including which timer field was being accessed.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data</i> =	The rIGPFTIMER key returned from the read attempt.
<i>Good Data</i> =	The rIGPFTIMER key from the timer key field.
<i>Logical Index</i> =	The rIORTLA tuple logical key (rIORTLA.port).
<i>Dump 1</i> =	The rIORTLA tuple.
<i>Dump 2</i> =	The rIGPFTIMER tuple buffer resulting from the read attempt.

Table 76 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The audit calls macro PSSVCPDLICI() to get the dcli for a frame relay SVC trunk, and PSBADDLICI is returned because the rIFC_PORTTYP or rIFRBC_SVC tuple is not found, or the port is not found within the rIFRBC_SVC tuple.
Possible Error Effect:	The audit may roll in errors and the circuit may not be recovered properly.
Corrective Action Taken:	None -- see Manual Action.
Manual Action Required:	1 = Check for, and if missing, insert the missing rIFC_PORTTYP tuple with rIFC_PORTTYP.port.member set to rIORTLA.port and rIFC_PORTTYP.port.module set to Glpcr_id. 2 = Check for, and if missing, insert the missing rIFRBC_SVC tuple with rIFRBC_SVC.gport set to rIFC_PORTTYP.sleevid. 3 = Check for, and if missing, add this port (rIORTLA.port) to an empty rIFRBC_SVC.mport[] in the rIFRBC_SVC tuple keyed by rIFC_PORTTYP.sleevid.
Dump Description:	
<i>Error Address</i> =	Address of the rIORTLA tuple.
<i>Bad Data</i> =	Local "dcli" returned from PS macro call.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIORTLA tuple (rIORTLA.port).
<i>Dump</i> =	rIORTLA tuple.

Table 77 Error Code: SHELF_SLC

Error Code:	SHELF_SLC
Error Description:	An attempt to determine the DCLU unit number from the S96Shelf circuit name has resulted in a return value of null unit.
Possible Error Effect:	The TAU line hardware path will not be torn down and re-established. The TAU line to TSI path will remain connected.

Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA (0)
<i>Good Data =</i>	Shelf name.
<i>Logical Index =</i>	Return value from SMIM function.
<i>Dump =</i>	NA.

Table 78 Error Code: SHELFREAD

Error Code:	SHELFREAD
Error Description:	Cannot read rIS96SHELF tuple.
Possible Error Effect:	No checking can be done on this port.
Corrective Action Taken:	None.
Manual Action Required:	Check ODD information for this relation.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	RLS96SHELF relation ID.
<i>Logical Index =</i>	Bank name.
<i>Dump =</i>	NA.

Table 79 Error Code: SLAVE_LINK

Error Code:	SLAVE_LINK
Error Description:	The status of the inter-SM nailup in the rIORTLA does not agree with the TMDlt_data[] array.
Possible Error Effect:	The port is OOS when it should not be.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	"master" field in TMDlt_data[].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 80 Error Code: SLC_PORT

Error Code:	SLC_PORT
Error Description:	The PORTLA audit attempted to convert a port name into the corresponding shelf or bank name. The return from the PC function indicated that the bank or shelf does not exist.
Possible Error Effect:	The port is not usable by call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Port name.
<i>Good Data =</i>	NA (0).
<i>Logical Index =</i>	Return value from the PC function.
<i>Dump =</i>	NA.

Table 81 Error Code: SLCLOSTTUP

Error Code:	SLCLOSTTUP
Error Description:	Cannot read the rIS96T1USG tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Skip auditing this tuple and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIS96T1USG tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	rIS96T1USG tuple relation ID.

<i>Logical Index</i> =	Bank name.
<i>Dump</i> =	NA.

Table 82 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple for this rPORTLA cannot be found.
Possible Error Effect:	This rPORTLA cannot be checked for correct FE state.
Corrective Action Taken:	Related audits will recover this port if required.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA (0).
<i>Bad Data</i> =	NA (0).
<i>Good Data</i> =	NA (0).
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	rPORTLA tuple.

Table 83 Error Code: STATE

Error Code:	STATE
Error Description:	The state field contains an invalid value.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	Contents of state.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 84 Error Code: STATE_IDL

Error Code:	STATE IDL
Error Description:	The state field indicates idle linkage, but the linkage does not indicate that the rPORTLA is linked to its owner (as indicated by the contents of the list field).
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	Contents of owner link.
<i>Good Data</i> =	Contents of list.
<i>Logical Index</i> =	Key of rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 85 Error Code: STATE_LNK

Error Code:	STATE LNK
Error Description:	The state field indicates that linkage exists but the link fields are empty.
Possible Error Effect:	The routing and terminal allocation subsystem will not correctly update the linkage when the port changes its associations with processes.
Corrective Action Taken:	The rPORTLA is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rPORTLA tuple.
<i>Bad Data</i> =	Contents of the bad link field.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of rPORTLA tuple.
<i>Dump</i> =	rPORTLA tuple.

Table 86 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A rIORTLA tuple was found linked to an unknown service list. The rIORTLA.list field must be marked as one of the legal service lists (either TMIS for in-service, or one a variety of possible out of service lists) as specified by DMLIST. This error may occur if a new service list has been added by terminal maintenance, but the PORTLA audit was never modified to recognize it.
Possible Error Effect:	The port will be unavailable to call processing and terminal maintenance activities since the associated rIORTLA tuple is in an unknown state.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to the proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIORTLA tuple.
<i>Bad Data =</i>	The rIORTLA.list field found by the audit.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIORTLA tuple (port name).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 87 Error Code: STATUS

Error Code:	STATUS
Error Description:	The port status supplementary information in the rIORTLA is invalid for the relationship between this SM and its mate (rISMPATH.mate_sm). If rIORTLA.suppl_info is <i>TMDLI</i> , <i>TMSNLI</i> , or <i>TMTMSCONN</i> , the two SMs must not be in an HSM-RSM relationship or an RSM-RSM relationship under the same HSM.
Possible Error Effect:	The rIORTLA is OOS when it should not be and unavailable for call processing.
Corrective Action Taken:	The port is placed on the to-be-idled list for later idling. All port maintenance status will be cleared out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The port status supplementary information (rIORTLA.suppl_info).
<i>Bad Data =</i>	The relationship between the SM and its mate.
<i>Good Data =</i>	The mate SM number which is served by the inter-SM path (rISMPATH.mate_sm).
<i>Logical Index =</i>	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump =</i>	The contents of the rIORTLA tuple.

Table 88 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	An rIORTLA tuple contains an invalid telephone status monitoring value (rIORTLA.tsm_stat). The status is inconsistent with a value derived from the associated rIPCBLA tuple or the rIPCBLA tuple was not accessible.
Possible Error Effect:	Telephone status monitoring may not work correctly for an end user.
Corrective Action Taken:	If the correct status was accessible, the audit corrects the value in the rIORTLA tuple (rIORTLA.tsm_stat). Otherwise the port is reinitialized and put on the to-be-idled list to be recovered by terminal maintenance.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	A boolean indicating if the associated status data was accessible.
<i>Bad Data =</i>	The rIORTLA tuple's telephone monitoring status (rIORTLA.tsm_stat).
<i>Good Data =</i>	The correct value of the port's monitor status value (rIORTLA.tsm_stat).
<i>Logical Index =</i>	Logical key of rIORTLA tuple (rIORTLA.port).
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIFC_PORTTYPE tuple.
<i>Dump 3 =</i>	The associated rIPCBLA tuple if available.

Table 89 Error Code: STUCK

Error Code:	STUCK
Error Description:	The multipoint analog and digital ports have list fields which indicate in-service but the CPE is out-of-service.
Possible Error Effect:	Multipoint modem pool ports are on an out-of-service CPE. If this CPE should be in-service, the CPE should be power cycled.
Corrective Action Taken:	The analog port is idled for recovery to valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of analog rIORTLA tuple.
<i>Bad Data =</i>	CPE state.
<i>Good Data =</i>	Analog rIORTLA list.
<i>Logical Index =</i>	Key of analog rIORTLA tuple.
<i>Dump 1 =</i>	Analog rIORTLA tuple.
<i>Dump 2 =</i>	Digital rIORTLA tuple.
<i>Dump 3 =</i>	rTEILA tuple.

Table 90 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	The analog modem pool port is in service but the corresponding digital one is not.
Possible Error Effect:	Modem pool call could be failing.
Corrective Action Taken:	Both ports are recovered by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	List the analog port is on.
<i>Good Data =</i>	List the digital port is on.
<i>Logical Index =</i>	Analog port name.
<i>Dump 1 =</i>	Analog rIORTLA tuple.
<i>Dump 2 =</i>	The corresponding digital rIORTLA tuple.

Table 91 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The D-port is on the TMLTWAIT list but the T321 index is not set (rIPRISTAT.t321_idx = TMTNULLIDX).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The list from rIORTLA tuple (rIORTLA.list).
<i>Bad Data =</i>	The T321 index from the rIPRISTAT tuple (rIPRISTAT.t321_idx).
<i>Good Data =</i>	The key of the rIPRISTAT tuple (rIPRISTAT.prigrp).
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump 1 =</i>	The rIORTLA tuple.
<i>Dump 2 =</i>	The rIPRISTAT tuple.

Table 92 Error Code: TPWAIT

Error Code:	TPWAIT
Error Description:	When the state of the rIORTLA tuple is DBPSIDLE, the wait for restoration bit should be cleared (rIORTLA.rstawait = DBNO).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The <i>rstawait</i> attribute is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The waiting for service acknowledge message flag of the rIORTLA tuple (rIORTLA.srvawait).
<i>Bad Data =</i>	The waiting for restoration flag of the rIORTLA tuple (rIORTLA.rstawait).
<i>Good Data =</i>	The logical port type.
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 93 Error Code: TRANCK

Error Code:	TRANCK
Error Description:	The B-port has been in a transient state for too long (longer than audit cycle).
Possible Error Effect:	This tuple is in a stuck OOS state and not available for call processing.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	port_owner.
<i>Good Data =</i>	d_owner.
<i>Logical Index =</i>	Logical key of rIORTLA tuple.
<i>Dump 1 =</i>	rIORTLA tuple.
<i>Dump 2 =</i>	rIORTLA tuple for D-port.

Table 94 Error Code: TRUNK_LINE

Error Code:	TRUNK_LINE
Error Description:	The port type for this port claims that it is a trunk but the rCKTDATA tuple that should exist for the trunk is missing.
Possible Error Effect:	There is no way for the audit to recover this port without the rCKTDATA information.
Corrective Action Taken:	None. The switch can't tell which data to believe.
Manual Action Required:	Determine whether the port is supposed to be a trunk. If it is, the add the rCKTDATA tuple using ODBE, or delete the rPORTTYP tuple. If a line, the rPORTTYP tuple needs to be updated to indicate that the port is a line.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Service class of rIORTLA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 95 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The logical key of the owner of the rIORTLA tuple points to an unassigned port, but the service class is not unassigned (rIFC_PORTTYP.servclass != RC_PORTUNA).
Possible Error Effect:	The port will remain unusable by call processing.
Corrective Action Taken:	The rIORTLA tuple is placed on the to-be-idled list for later idling and restoration to proper port status.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIORTLA tuple.
<i>Bad Data =</i>	The owner link of the rIORTLA tuple (rIORTLA.prtlst.st_ownk - DBSFPRTLST).
<i>Good Data =</i>	The service class of the port (rIFC_PORTTYP.servclass).
<i>Logical Index =</i>	Key of rIORTLA tuple.
<i>Dump =</i>	The rIORTLA tuple.

Table 96 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	One of two possible errors: (1) A CCS trunk is found on a CCSINIT or a trunk maintenance list for non-CCS trunks. (2) A non-CCS trunk is found on the CCSINIT or the trunk maintenance continuity list for only CCS trunks.
Possible Error Effect:	Corresponds to Error Description as follows:

	(1) The routing and terminal allocation subsystem won't hunt this trunk. Trunk maintenance may have difficulty recovering this trunk. (2) Any continuity testing done on a non-CCS trunk will fail.
Corrective Action Taken:	The rIORTLA is placed on the to-be-idled list for later idling and restoration to proper port status. When the CCS trunk is put on the to-be-idled list, a reset message is sent to the far-end office.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Contents of list.
<i>Good Data</i> =	Contents of owner link.
<i>Logical Index</i> =	Key of rIORTLA tuple.
<i>Dump</i> =	rIORTLA tuple.

PORTTYP

Software Release: 5E10 and later

1. PORT NAME TO SERVICE CLASS (PORTTYP) AUDIT

The switching system continually monitors the usage of lines and trunks by checking the software status of the ports associated with them. When a port is in use, it is assigned a service class. The port name to service class audit (PORTTYP) checks the service class to determine whether the port is in use. The service class and other information about the port is kept in the static port name to service class relation (RLFC_PORTTYP). The port linkage area relation (RLPORTLA) also contains information about the port, its status, and its associations; this information includes the owner of the port, the processes that are waiting to access the port, the status of the port (precut, postcut, and so forth), and a preemption flag showing whether the port is being preempted. The PORTTYP audit verifies that every assigned port in the static (RLFC_PORTTYP) relation has an associated dynamic port linkage area tuple (rIPORTLA). The PORTTYP audit also identifies extra rIPORTLA tuples.

This audit is part of the feature control audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The PORTTYP audit uses information in the static RLFC_PORTTYP relation to verify information in the dynamic RLPORTLA relation. The RLFC_PORTTYP relation contains information such as the type of port, the port name, the logical port type, and the service class to which the port is assigned. Figure 1 shows the fields in RLFC_PORTTYP and RLPORTLA checked by the audit.

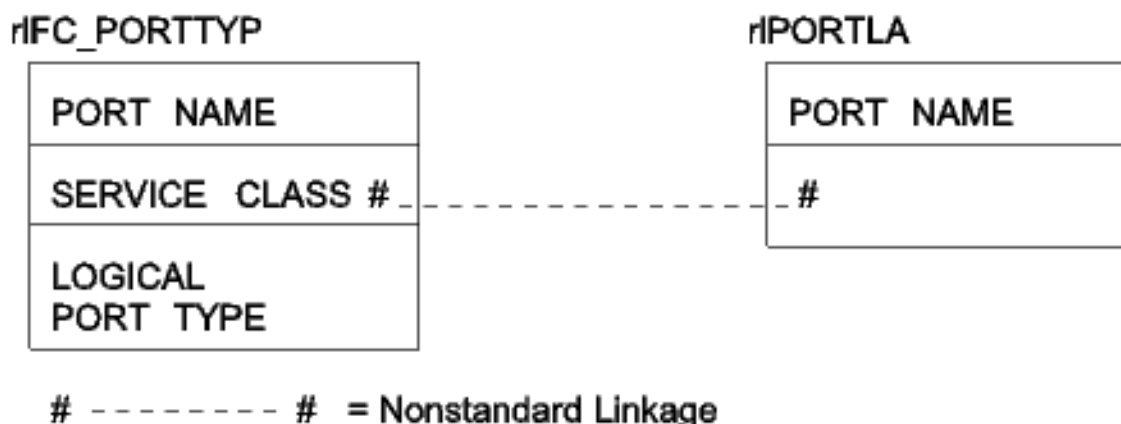


Figure 1 (PORTTYP-1) Data Structures of the PORTTYP Audit

3. AUDIT CHECKS

The PORTTYP audit invokes the static driver (DRSTATIC) to step through the tuples in RLFC_PORTTYP and makes these checks.

- (a) If the service class of the port is assigned (that is, it has a value other than null) and is a service class that should have a related rIPORTLA tuple, a tuple in rIPORTLA further describing the port should exist and have the same port name as the tuple in rLFC_PORTTYP. For tuples of rLFC_PORTTYP associated with assigned ports and an appropriate service class, the audit ensures that a tuple for the port member exists. If a corresponding tuple does not exist in rIPORTLA, one is created.
- (b) If the service class of the port is assigned but should not have a related rIPORTLA tuple, the audit ensures that no tuple for it exists in rIPORTLA. If a corresponding tuple exists in rIPORTLA, the audit deletes it.

- (c) If the service class of the port indicates that a rIDPB should exist, the audit will verify its existence and insert the rIDPB if missing. Extra rIDPB tuples will be found by the DPB audit.
- (d) The audit then reads the tuples of rIORTLA and ensures that a tuple exists in rIFC_PORTTYP. If no tuple exists in rIFC_PORTTYP, the tuple in rIORTLA is deleted.

4. ERROR HANDLING

If an rIORTLA tuple that should exist is missing, the audit calls a terminal maintenance function to insert that tuple into the RLPORTLA relation. Excess rIORTLA tuples are deleted.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An rIORTLA tuple not exist for a port whose service class indicates that an rIORTLA tuple should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rIORTLA tuple is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA tuple.
<i>Bad Data =</i>	Service class of the port.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	rIORTLA tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rIORTLA does not exist for a port whose service class indicates that an rIORTLA should exist.
Possible Error Effect:	This will cause lost resources.
Corrective Action Taken:	The rIORTLA tuple will be inserted into the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIORTLA.
<i>Bad Data =</i>	Service class of the port.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Port name.
<i>Dump =</i>	NA.

Table 3 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	An rIDPB does not exist for a D channel whose service class indicates that an rIDPB should exist.
Possible Error Effect:	Resources are not available for this port for call processing.
Corrective Action Taken:	The rIDPB tuple will be inserted into the database by the terminal maintenance subsystem if possible. A DB failure or an ODD problem could cause this rIDPB tuple to not be inserted.
Manual Action Required:	Check the following if this error rolls: (a) Valid servclass in rIORTTYP for port type. (b) Existence of rIDSLEQUIP tuple for the equipped port found in the

	Logical Key dump.
	(c) Valid dserv in rIDSLEQUIP for that port type.
Dump Description:	
<i>Error Address</i> =	None.
<i>Bad Data</i> =	Port type.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Port name.
<i>Dump</i> =	The rIFC_PORTTYP tuple.

POSACD

Software Release: 5E10 and later

1. AUTOMATIC CALL DISTRIBUTOR POSTION TO PORT (POSACD) AUDIT

The automatic call distributor management information system (ACD/MIS) is an integrated services digital network (ISDN) feature that is similar to the operator services position system (OSPS). This feature provides a full set of agent and supervisor service capabilities to commercial applications and internal service provider applications. In the ACD/MIS system, queued calls are presented to logged in agents at ISDN positions. Each agent is part of a serving team that is capable of handling certain type of calls. All agent activities are reported to the management information system (MIS), and each agent's call can be monitored by a supervisor. The RLACD_POS relation contains information that maps an ISDN position representing an ACD agent to a local port name and station identifier (that is, party). The RLACD_POS relation resides in the communications module processor (CMP) and is used by ACD software to route calls to the proper switching module (SM) that services the agent who will handle the call. Tuples in the RLACD_POS relation only represent ACD/MIS agents. Each RLACD_POS tuple has a corresponding dynamic tuple in the relation RLSTQDB (serving team queued data block). The POSACD audit runs in the operating kernel process (OKP) in the CMP in segmented mode. It is part of the OPAMBACD software package.

2. DATA STRUCTURES

The RLACD_POS static relation contains information that maps an ISDN ACD agent position to a particular port on an SM. This information is used for routing purposes for ACD hunted calls. The RLACD_POS is an indexed compaction access relation that is keyed by the **posnum** field which represents the ISDN position number. Other fields of interest in these static tuples are the port field which contains the logical port name, the party field which contains the station identifier, and the module field which indicates which SM contains the port.

The purpose of the POSACD audit is to insure that there is a one for one correspondence between the static RLACD_POS tuples and the dynamic RLSTQDB tuples in the CMP.

3. AUDIT CHECKS

The POSACD audit uses the STATIC audit driver to traverse through each tuple in the RLACD_POS relation. For each tuple, the POSACD audit will make the following checks:

If the POSACD audit is entered:

- LOSTTUP = Check to make sure that the RLACD_POS tuple has a corresponding RLSTQDB tuple.
- CURLNK = Check to make sure the pointer to the owning RLHCSERVMT tuple for RLSTQDB tuples on the unoccupied list is not corrupt once we have found a missing RLSTQDB tuple.
- MATELINK = Check to make sure that we can add a missing RLSTQDB tuple to the unoccupied list defined in the RLHCSERVMT relation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CURLNK

Error Code:	CURLNK
Error Description:	The audit was unable to get a pointer to the rHCSEVTM tuple that represents the unoccupied list of rISTQDB tuples when recovering a lost rISTQDB tuple.
Possible Error Effect:	This may mean a possible loss of rISTQDB tuple resources.
Corrective Action Taken:	None. The error is reported and the related HCSEVTM audit is scheduled to clean up the rHCSEVTM tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACD_POS tuple.
<i>Bad Data =</i>	The bad pointer obtained for the rHCSEVTM tuple.
<i>Good Data =</i>	The logical key of the unoccupied list rHCSEVTM tuple.
<i>Logical Index =</i>	The logical key of the rIACD_POS tuple.
<i>Dump 1 =</i>	The rIACD_POS tuple.
<i>Dump 2 =</i>	The rISTQDB tuple that was recovered.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a static rIACD_POS tuple that did not have a corresponding dynamic rISTQDB tuple.
Possible Error Effect:	This may prevent any hunted calls from being presented to the ACD agent position represented by the rIACD_POS tuple.
Corrective Action Taken:	The audit will try to insert the missing rISTQDB tuple into the unoccupied list defined off the RLHCSEVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACD_POS tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key of the rIACD_POS tuple.
<i>Dump =</i>	The rIACD_POS tuple.

Table 3 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	While recovering a lost rISTQDB tuple, the audit was unable to link this rISTQDB tuple to the unoccupied list defined in by an rHCSEVTM tuple.
Possible Error Effect:	This may result in the possible loss of rISTQDB tuple resources.
Corrective Action Taken:	None. However, the related STQDB and HCSEVTM audits are scheduled to clean up the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIACD_POS tuple.
<i>Bad Data =</i>	The return code from the call to DAatsast used to link the rISTQDB tuple to the unoccupied list. NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key of the rIACD_POS tuple.
<i>Dump 1 =</i>	The rIACD_POS tuple.
<i>Dump 2 =</i>	The rISTQDB tuple that was to be inserted in the unoccupied list.

POSDB

Software Release: 5E10 and later

1. POSITION DATA BLOCK (POSDB) AUDIT

Design alternatives for operator actions data included using the stack or creating a new area or areas for them and linking them from the position data block. The rIPOSDB relation defines all the dynamic data associated with a position or operator login identification.

This audit is part of the Operator Services Position System (OSPS) audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It operates in the OSDS-M environment in the segmented mode only.

2. DATA STRUCTURES

The data structure checked by the POSDB audit is the position data block (RLPOSDB). The audit makes its checks based on information in the position-to-local port mapping relation (RLLPRT_POS), a static relation in the SMs. It makes a check on the operator login ID relation (RLOLOGINID). The RLPOSDB relation has owner-member linkages with the associated process block relation (RLAPB) and the loop data block relation (RLLOOPDB).

For details on the fields contained in the RLPOSDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

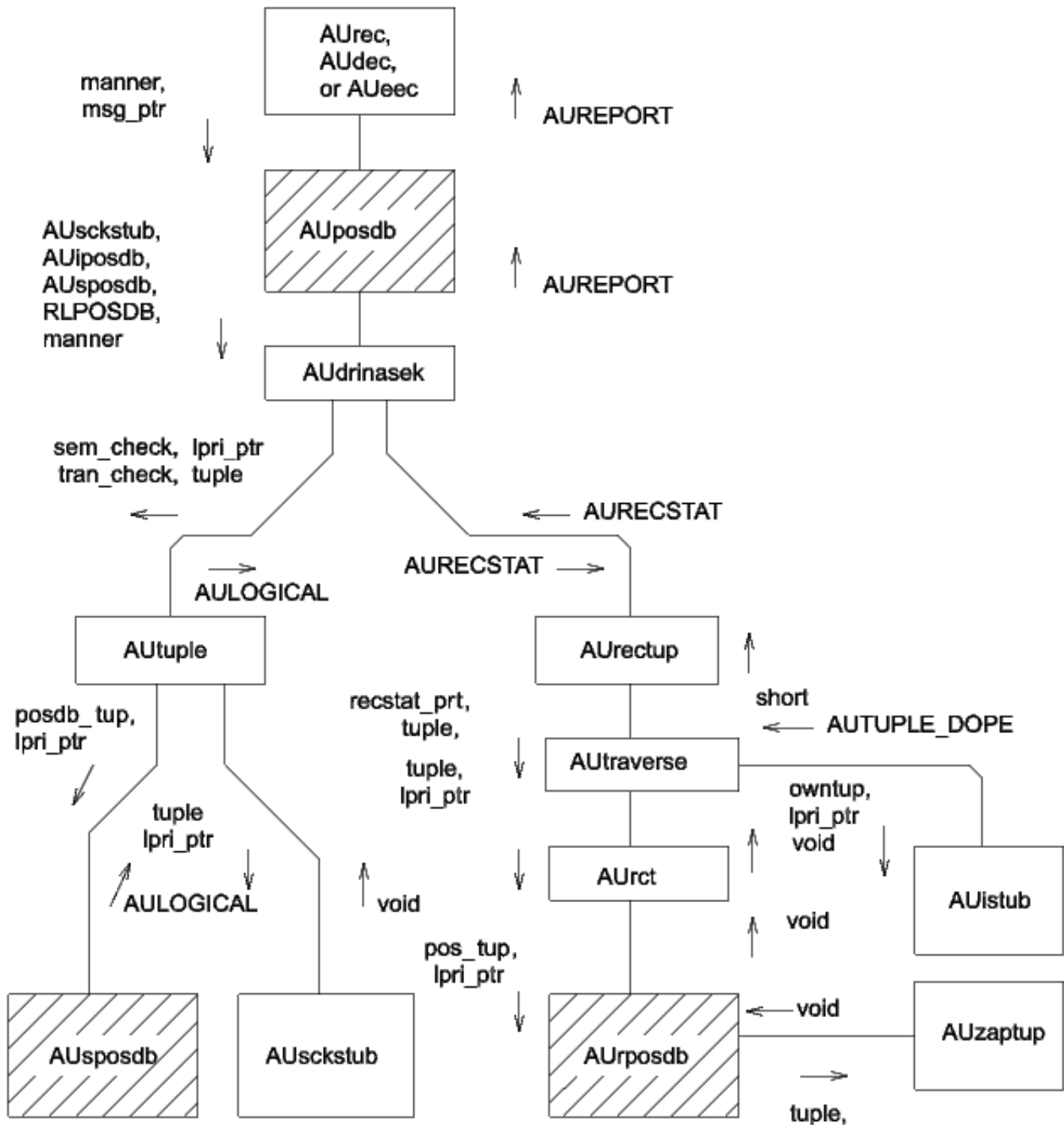


Figure 1 (POSDB-1) POSDB Audit Structure Chart

3. AUDIT CHECKS

The POSDB audit uses the static driver (DRSTATIC) to step through the RLPOSDb relation, as shown in Figure 1 .

- (a) Initially, the audit calls a dynamic-to-static existence check function for each tuple in RLPOSDb. The existence check function verifies that a corresponding tuple exists in the static position to local port mapping relation (RLLPRT_POS). If not, the rIPOSDb tuple is considered to be an extra tuple and is deleted.

(b) The audit calls a semantic check function to perform the following checks.

- (1) The audit verifies that the count (loopdb_ct) item in the rIPOSDB is consistent with the number of rILOOPDBs linked to the rIPOSDB. It also checks that no more than one LOOPDB is linked to a POSDB.

NOTE: Checks 2-4 are only done when an operator is logged in (that is, when the position state field (pos_state) is not unoccupied).

- (2) The audit verifies that the operator ID (op_id) item in the RLPOSDB is consistent with the operator ID item in the RLOLOGINID relation.
- (3) The audit verifies that the position number (pos_num) item in the rIPOSDB is consistent with the position number item in the rILPRT_POS relation.
- (4) The audit verifies that the internal operator ID (intopid) item and the serv_tm attribute in rIPOSDB is consistent with the internal operator ID and the serv_tm in the rILOGINID relation. The POSDB audit calls the AUdrinasek() driver to step through each rIPOSDB tuple. The driver calls AUTuple() to perform the linkage and quarantine checks. If the syntactic checks pass, AUTuple calls the semantic check function, AUSposdb, to perform the semantic checks.

4. ERROR HANDLING

All errors result in output messages. The corresponding position terminal process will traverse the call and purge it from the system. Also, an automatic call distribution (ACD) primitive is called to display logoff information to the AM.

The audit schedules the loop data block (LOOPDB) and associated process block (APB) audits as related audits.

The audit generates these error codes and those produced by the DRSTATIC driver.

Upon detection of errors in either syntactic or semantic data, AUrectup() is called to recover the bad tuple. The recovery is done by AUrposdb(). Whenever an error is found in a POSDB relation, the RLPOSDB contents will be printed and the related audits will be scheduled to run.

When an error is detected in a RLPOSDB tuple, the audit will recover the bad tuple and send a message to the ACD/AM to notify it that the position has been taken out of service. The ACD primitive ACpos_stat() is called to take the appropriate action to log off the operator. The audit will also schedule the LOOPDB audit to be run in the elevated mode.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra tuple was found.
Possible Error Effect:	Garbage was left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The extra rIPOSDB tuple that was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPOSDB.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rIPOSDB.
<i>Dump</i> =	rIPOSDB tuple.

Table 2 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The ODD for this port in rIPOSDB is corrupted; it is compared against the static rILOGINID.
Possible Error Effect:	The operator ID, (op_id), the internal operator ID, (intopid), and the serving team (serv_team) call the ACSIfcrrct function, which will call an assert. The function will abort, causing a halt to handling the message.
Corrective Action Taken:	The rIPOSDB will be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPOSDB tuple.
<i>Bad Data</i> =	NA (see dump of rIPOSDB).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIPOSDB.
<i>Dump 1</i> =	rIPOSDB.
<i>Dump 2</i> =	rILOGINID.

Table 3 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	A static tuple exists for rILPRT_POS but the static tuple for rILOGINID does not exist.
Possible Error Effect:	rIPOSDB will be updated from the rILPRT_POS, but the attributes related to rILOGINID will not be updated.
Corrective Action Taken:	The rILOGINID must be created and initialized.
Manual Action Required:	Action must be taken to create and initialize the rILOGINID relation.
Dump Description:	
<i>Error Address</i> =	Address of rIPOSDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIPOSDB.
<i>Dump</i> =	rIPOSDB.

Table 4 Error Code: PORT

Error Code:	PORT
Error Description:	A static tuple exists for rILOGINID but the static tuple for rILPRT_POS does not exist.
Possible Error Effect:	rIPOSDB will be updated from the rILOGINID, but the attributes related to rILPRT_POS will not be updated.
Corrective Action Taken:	The rILPRT_POS must be created and initialized.
Manual Action Required:	Action must be taken to create and initialize rILPRT_POS.
Dump Description:	
<i>Error Address</i> =	Address of rIPOSDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIPOSDB.
<i>Dump</i> =	rILPRT_POS.

Table 5 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The status of LOOPDB_CT linkages to the rIPOSDB is invalid.
Possible Error Effect:	This may attempt to serve calls that do not exist. It may also cause calls not to be served that do exist.
Corrective Action Taken:	The corresponding position terminal process will traverse the call and purge it from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPOSDB.

<i>Bad Data</i> =	LOOPDB_CT value.
<i>Good Data</i> =	Correct number of rLOOPDBs linked to the rIPOSDB.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rIPOSDB.

Table 6 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The ODD for this port kept in the rIPOSDB is corrupted. The position number is compared with static rLPRT_POS.
Possible Error Effect:	If the position number is occupied and should not be, it will want to access a call. If the position number is unoccupied and should be, it will not access the call.
Corrective Action Taken:	The rIPOSDB will be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIPOSDB tuple.
<i>Bad Data</i> =	NA (see dump of rIPOSDB).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIPOSDB.
<i>Dump</i> =	rIPOSDB.

POSPORT

Software Release: 5E10 and later

1. POSITION TO PORT MAPPING (POSPORT) AUDIT

Features of the Operator Service Position System (OSPS) include basic automatic call distribution (ACD) and call handling. These features contain the call processing software needed for an operator to log onto a basic services terminal (BDT) and handle requests for operator services. The ACD device routes calls to available positions. Audits are needed to maintain the integrity of the data structures needed to perform these OSPS functions. The POSPORT audit validates data related to equipped operator positions.

The POSPORT audit is part of the OSPS audits subgroup. The audit runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The audit uses two relations, the dynamic serving team queue data block relation (RLSTQDB) and the static port position relation (RLPOS_PORT).

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) to perform a static-to-dynamic existence check between RLPOS_PORT and RLSTQDB.

4. ERROR HANDLING

When the audit detects a lost tuple, it creates a tuple in RLSTQDB and initializes it from the information in RLPOS_PORT.

The POSPORT audits schedules the STQDB audit as a related audit when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A r1STQDB does not exist for a position when it should.
Possible Error Effect:	This condition results in lost resources.
Corrective Action Taken:	The r1STQDB tuple will be inserted into the data base and added to the unoccupied list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of r1STQDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of r1STQDB.
<i>Dump =</i>	NA.

26. PP - PT

PPDB-A

Software Release: 5E10 - 5E11

1. PACKET PIPE DATA BLOCK (PPDB) AUDIT

The frame relay protocol handler (FRPH) is a component of the communication path between the cell site and the switch. The FRPH provides an interface for wireless' code division multiple access (CDMA) voice and signaling packets from the external transmission facility into the packet switching unit (PSU) of the switching module (SM).

Special data for the FRPH is kept in three arrays in the FRPH, PSppdb[], PSppcb[], and PSspts[]. PSppdb[], the packet pipe data block, stores the "static" per packet pipe information required to define a packet pipe (PP) and is indexed by the packet pipe channel group member number. Also, the PSppdb[] is write-protected.

PSppcb[], the packet pipe control block, stores the "dynamic" per packet pipe information required to define a Packet Pipe and is also indexed by the packet pipe channel group member number. The PSppcb[] is not write-protected.

PSspts[], the packet pipe time slot assignment table, stores the PP member number for the packet pipe and is indexed by the packet pipe's starting timeslot. The PSspts[] is write-protected.

The PPDB audit runs in segmented mode and is in the frame relay PH (the FRPH is a PH4).

2. DATA STRUCTURES

The PPDB audit validates the following data:

- In PSppdb[]:
 - isdata* = The entry's valid data bit.
 - channel* = The entry's index into the DCDB.
- In PSppcb[]:
 - state* = The entry's state, the state of the packet pipe.
- In PSdcdcb[]:
 - isdata* = The DCDB entry's valid data bit.
 - member* = The DCDB entry's semantic link to the packet pipe.
 - phctype* = The DCDB entry's PH channel type.
- In PSspts[]:
 - member* = The entry's semantic link to the packet pipe.
 - avail_cnt* = The entry's count of available, contiguous timeslots.

3. AUDIT CHECKS

The following checks are performed by the PPDB audit:

- If PSppdb[].isdata is DBNO,
 - BADSTATE = Verify that PSppcb[].state is PSPPOFF.
 - INV_TIMER = Verify that no timers are up and running.
- If PSppdb[].isdata is DBYES,
 - INV_STATE = Verify that PSppdb[].state is not PSPPOFF and is in range.
 - DCHAN = Verify that PSppdb[].channel is in range.
 - BADOWNER = Validate PH channel type, member, and isdata in the PSdcbd[].
 - MEMBAD = Validate member in the corresponding PSpts[] entry.
 - INV_TIMER = Verify that any necessary timers are up and running.
- For each entry in PSpts[],
 - OUTOFRANGE = Verify that PSpts[].member is in range.
 - PPKEY = Validate the channel in the corresponding PSppdb[] entry.
 - INVLIST = Verify that all timeslots in the PP have the same PSpts[].member.
 - AVAIL_CNT = Validate the PSpts[].avail_cnt.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT
Error Description:	There are two possible errors: (1) The count of available timeslots (PSpts[].avail_cnt) for each timeslot should be one greater than the count in the next PSpts[] entry when there is no packet pipe associated with the timeslot (PSpts[].member = PSPPPERCG), and is not. (2) It should be zero when there is a valid packet pipe (PSpts[].member is greater than or equal to 0 and less than the value of PSPPPERCG), and is not.
Possible Error Effect:	If the count of available timeslots is less than is truly available, we may not be able to set up new packet pipes on that PH which would affect call capacity. If the count of available timeslots is greater than is available, setting up new

	packet pipes on the PH4 could cause an overwrite onto existing PP's, which would cause both PP's to be removed.
Corrective Action Taken:	The audit will report the error and reset all the counts of available timeslots in the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The count of available timeslots that should be in the current entry.
<i>Bad Data =</i>	The count of available timeslots according to the current entry (PSpts[].avail cnt).
<i>Good Data =</i>	The semantic link from PSpts[] to PSppdb[] (PSpts[].member).
<i>Logical Index =</i>	The index into PSpts[].
<i>Dump =</i>	The contents of the PSpts[] entry.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	Either the point-to, point-back checks failed between PPDB and DCDB, or the channel is turned off in the DCDB (PSdcbd[].isdata = DBNO) when the PPDB is turned on (PSppdb[].isdata = DBYES), or the PH channel type is wrong (PSdcbd[].phctype != PSPPCCHAN).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause Packet Pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The PH channel type of the DCDB (PSdcbd[].phctype).
<i>Bad Data =</i>	The semantic link from the DCDB to the packet pipe (PSdcbd[].member).
<i>Good Data =</i>	The valid data attribute of the DCDB (PSdcbd[].isdata).
<i>Logical Index =</i>	The index into both PSppdb[] and PSppcb[].
<i>Dump 1 =</i>	The contents of the PSppdb[] entry.
<i>Dump 2 =</i>	The contents of the PSppcb[] entry.

Table 3 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The packet pipe is turned off (PSppdb[].isdata = DBNO), but the PPCB is in an invalid state (PSppcb[].state != PSPPOFF).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit sets the state of the PPCB to off (PSppcb[].state = PSPPOFF).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index into the PSppdb[] and PSppcb[].
<i>Bad Data =</i>	The state of the packet pipe (PSppcb[].state).
<i>Good Data =</i>	The valid data attribute of the packet pipe (PSppdb[].isdata).
<i>Logical Index =</i>	The index into both PSppdb[] and PSppcb[].
<i>Dump 1 =</i>	The contents of the PSppdb[] entry.
<i>Dump 2 =</i>	The contents of the PSppcb[] entry.

Table 4 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The semantic link to the DCDB is out of range (PSppdb[].channel > PSMAXCHN).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The index into the PSppdb[] and PSppcb[].
<i>Bad Data</i> =	The semantic link to the DCDB (PSppdb[].channel).
<i>Good Data</i> =	The valid data attribute of the packet pipe (PSppdb[].isdata).
<i>Logical Index</i> =	The index into both PSppdb[] and PSppcb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.
<i>Dump 2</i> =	The contents of the PSppcb[] entry.

Table 5 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The packet pipe is turned on (PSppdb[].isdata = DBYES), but the PPCB is in an invalid state (PSppcb[].state = PSPPOFF) or the state is out of range.
Possible Error Effect:	A packet pipe in the wrong state will most likely assert when any activity is attempted. The PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into both PSppdb[] and PSppcb[].
<i>Bad Data</i> =	The state of the packet pipe (PSppcb[].state).
<i>Good Data</i> =	The valid data attribute of the packet pipe (PSppdb[].isdata).
<i>Logical Index</i> =	The index into both PSppdb[] and PSppcb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.
<i>Dump 2</i> =	The contents of the PSppcb[] entry.

Table 6 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	Either the packet pipe is inactive (PSppdb[].isdata = DBNO) and there are still timers set in the PSppcb[], or the packet pipe is active (PSppdb[].isdata = DBYES) and timers are running (or not running) incorrectly for the state of the packet pipe.
Possible Error Effect:	Either unnecessary timers are running, or necessary timers are not running. When unnecessary timers expire, the packet pipe will attempt some invalid action and probably assert. If necessary timers are not there, the packet pipe may remain forever in some transitional state.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The type of the timer (PSppcb[].tmr_type).
<i>Bad Data</i> =	This field is of size long and dumps two values. In the upper most two bytes(right) it dumps the index of the active, idle or loop timers; on the remaining two bytes it dumps the index to the start timer. The algorithm used is, PSppcb[].tmr_idx << 16 PSppcb[].st_tmr_idx.
<i>Good Data</i> =	The state of the packet pipe (PSppcb[].state). If the state is PSPPOFF, unnecessary timers are running. Otherwise, either case may be true.
<i>Logical Index</i> =	The index into both PSppdb[] and PSppcb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.
<i>Dump 2</i> =	The contents of the PSppcb[] entry.

Table 7 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The PSppdb[] entry is pointing to an entry in PSspts[] that does not totally point back to it. PSppdb[].channel is used to access the PSspts[] entry and the first timeslot has the correct PSspts[].member. Each of the PSspts[].member attributes should point to the same PSppdb[], but at least one does not.
Possible Error Effect:	The bandwidth is incorrect, therefore information in the frame is probably being lost.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and

	restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The number of timeslots that should have the same member in PSspts[].
<i>Bad Data =</i>	The semantic link from PSspts[] to PSppdb[] (PSspts[].member).
<i>Good Data =</i>	The value that should be in PSspts[].member.
<i>Logical Index =</i>	The index into PSspts[].
<i>Dump 1 =</i>	The contents of the PSspts[] entry.
<i>Dump 2 =</i>	The contents of the PSdcdb[] entry.
<i>Dump 3 =</i>	The contents of the PSppdb[] entry.

Table 8 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The PSppdb[] entry is pointing to an entry in PSspts[] that does not point back to it. PSppdb[].channel is used to access the PSspts[] entry, but PSspts[].member should map into the PSppdb[] entry that took us to PSspts[], and it does not.
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources. Also, the packet pipe may be losing frames.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index into PSspts[] (timeslot).
<i>Bad Data =</i>	The invalid value for PP in the PSspts[] (PSspts[].member).
<i>Good Data =</i>	The correct value for PP.
<i>Logical Index =</i>	The index into PSppdb[].
<i>Dump 1 =</i>	The contents of the PSppdb[] entry.
<i>Dump 2 =</i>	The contents of the PSspts[] entry.

Table 9 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The index into the PSppdb[] entry is out of range (PSspts[].member > PSPPPERCG).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index into PSspts[].
<i>Bad Data =</i>	The semantic link from PSspts[] to PSppdb[] (PSspts[].member).
<i>Good Data =</i>	The count of available timeslots in the current entry (PSspts[].avail_cnt).
<i>Logical Index =</i>	The index into PSspts[].
<i>Dump =</i>	The contents of the PSspts[] entry.

Table 10 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	The PSspts[] entry is pointing to an entry in PSppdb[] that does not point back to it. PSspts[].member is used to access the PSppdb[] entry, but PSppdb[].channel should map into the same PSspts[], and it does not.
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The index into PSppts[].
<i>Bad Data</i> =	The semantic link from PSppts[] to PSppdb[] (PSppts[].member).
<i>Good Data</i> =	The semantic link from PSppdb[] to PSppts[] (PSppdb[].channel).
<i>Logical Index</i> =	The index into PSppts[].
<i>Dump 1</i> =	The contents of the PSppts[] entry.
<i>Dump 2</i> =	The contents of the PSppdb[] entry.

PPDB-B

Software Release: 5E12 and later

1. PACKET PIPE DATA BLOCK (PPDB) AUDIT

The frame relay protocol handler (FRPH) is a component of the communication path between the cell site and the 5ESS[®] digital cellular switch (5E-DCS). The FRPH provides an interface for wireless' code division multiple access (CDMA) voice and signaling packets from the external transmission facility into the packet switching unit (PSU) of the switching module (SM).

Special data for the FRPH is kept in three arrays in the FRPH, *PSppdb[]*, *PSppcb[]*, and *PSppts[]*. *PSppdb[]*, the packet pipe data block, stores the "static" per packet pipe information required to define a packet pipe (PP) and is indexed by the packet pipe channel group member number. Also, the *PSppdb[]* is write-protected.

PSppcb[], the packet pipe control block, stores the "dynamic" per packet pipe information required to define a packet pipe and is also indexed by the packet pipe channel group member number. The *PSppcb[]* is not write-protected.

PSppts[], the packet pipe time slot assignment table, stores the PP member number for the packet pipe and is indexed by the packet pipe's starting timeslot. The *PSppts[]* is write-protected.

The PPDB audit runs in segmented mode and is in the frame relay PH (the FRPH is a PH4).

2. DATA STRUCTURES

For details on the fields contained in these structures, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: AVAIL_CNT

Error Code:	AVAIL_CNT
Error Description:	The count of available timeslots (<i>PSppts[].avail_cnt</i>) for each timeslot should be one greater than the count in the next <i>PSppts[]</i> entry when there is no packet pipe associated with the timeslot (<i>PSppts[].member = PSPPPERCG</i>), and is not, or it should be zero when there is a valid packet pipe (<i>PSppts[].member</i> is greater than or equal to 0 and less than the value of <i>PSPPPERCG</i>), and is not.
Possible Error Effect:	If the count of available timeslots is less than is truly available, we may not be able to set up new packet pipes on that protocol handler (PH) which would affect call capacity. If the count of available timeslots is greater than is available, setting up new packet pipes on the PH4 could cause an overwrite onto existing PP's, which would cause both PP's to be removed.
Corrective Action Taken:	The audit will report the error and reset all the counts of available timeslots in the array.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The count of available timeslots that should be in the current entry.
<i>Bad Data</i> =	The count of available timeslots according to the current entry (PSppts[].avail cnt).
<i>Good Data</i> =	The semantic link from PSppts[] to PSppdb[] (PSppts[].member).
<i>Logical Index</i> =	The index into PSppts[].
<i>Dump</i> =	The contents of the PSppts[] entry.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	Either the point-to, point-back checks failed between PPDB and DCDB, or the channel is turned off in the DCDB (PSdcbd[].isdata = DBNO) when the PPDB is turned on (PSppdb[].isdata = DBYES), or the PH channel type is wrong (PSdcbd[].phctype != PSPPCHAN).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The PH channel type of the DCDB (PSdcbd[].phctype).
<i>Bad Data</i> =	The semantic link from the DCDB to the packet pipe (PSdcbd[].member).
<i>Good Data</i> =	The valid data attribute of the DCDB (PSdcbd[].isdata).
<i>Logical Index</i> =	The index into both PSppdb[] and PSppcb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.
<i>Dump 2</i> =	The contents of the PSppcb[] entry.

Table 3 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The packet pipe is turned off (PSppdb[].isdata = DBNO), but the PPCB is in an invalid state (PSppcb[].state != PSPPOFF).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit sets the state of the PPCB to off (PSppcb[].state = PSPPOFF).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into the PSppdb[] and PSppcb[].
<i>Bad Data</i> =	The state of the packet pipe (PSppcb[].state).
<i>Good Data</i> =	The valid data attribute of the packet pipe (PSppdb[].isdata).
<i>Logical Index</i> =	The index into both PSppdb[] and PSppcb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.
<i>Dump 2</i> =	The contents of the PSppcb[] entry.

Table 4 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The semantic link to the DCDB is out of range (PSppdb[].channel > PSMAXCHN).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into the PSppdb[] and PSppcb[].
<i>Bad Data</i> =	The semantic link to the DCDB (PSppdb[].channel).
<i>Good Data</i> =	The valid data attribute of the packet pipe (PSppdb[].isdata).
<i>Logical Index</i> =	The index into both PSppdb[] and PSppcb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.

<i>Dump 2 =</i>	The contents of the PSppcb[] entry.

Table 5 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The packet pipe is turned on (PSppdb[].isdata = DBYES), but the PPCB is in an invalid state (PSppcb[].state = PSPPOFF) or the state is out of range.
Possible Error Effect:	A packet pipe in the wrong state will most likely assert when any activity is attempted. The PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index into both PSppdb[] and PSppcb[].
<i>Bad Data =</i>	The state of the packet pipe (PSppcb[].state).
<i>Good Data =</i>	The valid data attribute of the packet pipe (PSppdb[].isdata).
<i>Logical Index =</i>	The index into both PSppdb[] and PSppcb[].
<i>Dump 1 =</i>	The contents of the PSppdb[] entry.
<i>Dump 2 =</i>	The contents of the PSppcb[] entry.

Table 6 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	Either the packet pipe is inactive (PSppdb[].isdata = DBNO) and there are still timers set in the PSppcb[], or the packet pipe is active (PSppdb[].isdata = DBYES) and timers are running (or not running) incorrectly for the state of the packet pipe.
Possible Error Effect:	Either unnecessary timers are running, or necessary timers are not running. When unnecessary timers expire, the packet pipe will attempt some invalid action and probably assert. If necessary timers are not there, the packet pipe may remain forever in some transitional state.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The type of the timer (PSppcb[].tmr_type).
<i>Bad Data =</i>	This field is of size long and dumps two values. In the upper most two bytes(right) it dumps the index of the active, idle or loop timers; on the remaining two bytes it dumps the index to the start timer. The algorithm used is, PSppcb[].tmr_idx << 16 PSppcb[].st_tmr_idx.
<i>Good Data =</i>	The state of the Packet Pipe (PSppcb[].state). If the state is PSPPOFF, unnecessary timers are running. Otherwise, either case may be true.
<i>Logical Index =</i>	The index into both PSppdb[] and PSppcb[].
<i>Dump 1 =</i>	The contents of the PSppdb[] entry.
<i>Dump 2 =</i>	The contents of the PSppcb[] entry.

Table 7 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	The PSppdb[] entry is pointing to an entry in PSspts[] that does not totally point back to it. PSppdb[].channel is used to access the PSspts[] entry and the first timeslot has the correct PSspts[].member. Each of the PSspts[].member attributes should point to the same PSppdb[], but at least one does not.
Possible Error Effect:	The bandwidth is incorrect, therefore information in the frame is probably being lost.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The number of timeslots that should have the same member in PSspts[].
<i>Bad Data =</i>	The semantic link from PSspts[] to PSppdb[] (PSspts[].member).

<i>Good Data</i> =	The value that should be in PSspts[].member.
<i>Logical Index</i> =	The index into PSspts[].
<i>Dump 1</i> =	The contents of the PSspts[] entry.
<i>Dump 2</i> =	The contents of the PSdcbd[] entry.
<i>Dump 3</i> =	The contents of the PSppdb[] entry.

Table 8 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The PSppdb[] entry is pointing to an entry in PSspts[] that does not point back to it. PSppdb[].channel is used to access the PSspts[] entry, but PSspts[].member should map into the PSppdb[] entry that took us to PSspts[], and it does not.
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources. Also, the packet pipe may be losing frames.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into PSspts[] (timeslot).
<i>Bad Data</i> =	The invalid value for PP in the PSspts[] (PSspts[].member).
<i>Good Data</i> =	The correct value for PP.
<i>Logical Index</i> =	The index into PSppdb[].
<i>Dump 1</i> =	The contents of the PSppdb[] entry.
<i>Dump 2</i> =	The contents of the PSspts[] entry.

Table 9 Error Code: MEMBER_BAD

Error Code:	MEMBER_BAD
Error Description:	The frame relay mapping table (FRMAP) is used and audited when the DCDB phctype is an FRBC channel and the data bit is set. This is a semantic check function and verifies that the dlci value stored in the FRMAP is equal to the header index into the table.
Possible Error Effect:	The FRMAP contains information used by the ISDN frame relay IOP software. Frames could be lost if the dlci is incorrect.
Corrective Action Taken:	The audit will report the error and reset the dlci to the header index in the table.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The contents of PSfrmap[].header.dest.
<i>Bad Data</i> =	The contents of PSfrmap[].header.dlci.
<i>Good Data</i> =	The header index (&PSfrmap[].header).
<i>Logical Key</i> =	The index into PSfrmap[].
<i>Dump</i> =	None.

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The index into the PSppdb[] entry is out of range (PSspts[].member > PSPPPERCG).
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into PSspts[].
<i>Bad Data</i> =	The semantic link from PSspts[] to PSppdb[] (PSspts[].member).
<i>Good Data</i> =	The count of available timeslots in the current entry (PSspts[].avail_cnt).
<i>Logical Index</i> =	The index into PSspts[].
<i>Dump</i> =	The contents of the PSspts[] entry.

Table 11 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	The PSspts[] entry is pointing to an entry in PSppdb[] that does not point back to it. PSspts[].member is used to access the PSppdb[] entry, but PSppdb[].channel should map into the same PSspts[], and it does not.
Possible Error Effect:	If there is a packet pipe associated with the PPDB, it is not able to transmit or receive frames. Otherwise, the PPDB and PPCB may be lost resources, which could cause packet pipe set up to fail due to a lack of resources.
Corrective Action Taken:	The audit will report the error and cause the packet pipe to be removed and restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into PSspts[].
<i>Bad Data</i> =	The semantic link from PSspts[] to PSppdb[] (PSspts[].member).
<i>Good Data</i> =	The semantic link from PSppdb[] to PSspts[] (PSppdb[].channel).
<i>Logical Index</i> =	The index into PSspts[].
<i>Dump 1</i> =	The contents of the PSspts[] entry.
<i>Dump 2</i> =	The contents of the PSppdb[] entry.

PPLLNST

Software Release: 5E10 and later

1. PRECEDENCE AND PREEMPTION LARGE LINE STATUS (PPLLNST) AUDIT

The precedence and preemption (PP) feature of the automatic voice network (AUTOVON) can be tailored for applications through individualized dialing plans and feature-specific data. Precedence and preemption features are assigned to members of terminal groups; each user in a precedence and preemption group may or may not have access to the AUTOVON network. There are five levels of precedence. Routine calls occur at precedence level 4, the lowest precedence level. Precedence calls are calls with precedence levels 0 (flash override), 1 (flash), 2 (immediate), 3 (priority), and 4 (routine), where 0 is the highest precedence level.

The precedence and preemption feature is a modular feature designed within the framework used by the business and residence custom services (BRCS) feature. It executes under the control of the feature execution (FEX) system, which supports the concurrent execution of multiple subprocesses within a single terminal process. A record of the precedence level must be maintained for each port active on a call. Per-call dynamic data, including the precedence level, is stored in the process control block linkage area relation (RLPCBLA). A routing and terminal allocation (RTA) primitive also maintains the record of the precedence level of each hunted port. Four dynamic relations keep track of precedence and preemption: the precedence and preemption line status relation (RLPPLNST), the precedence and preemption large line status relation (RLPPLLNST), the precedence and preemption trunk status relation (RLPPTKST), and the precedence and preemption large trunk status relation (RLPPLTKST). Four audits check these relations; the audits are named for the relations they check.

The PPLLNST audit is part of the RTA audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The data structures the PPLLNST audit checks are the dynamic RLPPLLNST and the static multi-line hunt group relation (RLRT_MHG). The fields checked include the key field, the hunt type, and the processor (module) number.

3. AUDIT CHECKS

The PPLLNST audit calls the static driver (DRSTATIC) to perform a dynamic-to-static existence check. The DRSTATIC driver steps through RLPPLLNST and retrieves tuples sequentially. For each tuple retrieved, the semantic check function does the following.

- (a) The audit calculates the correct key of the tuple of rIPPLLNST and compares it with the key value retrieved by the data administration (DA) primitive DAGtppln.
- (b) Using the correct key, the audit reads rIRT_MHG and verifies that the hunt type is regular (RTH_PPLMHG).
- (c) If the hunt type is not correct, the audit checks whether rIRT_MHG is being updated. If it is not being updated, there is an error. If it is being updated, control returns to DRSTATIC and another tuple is checked.
- (d) If the hunt type is correct, the audit checks the module number of the tuple in rIRT_MHG. In the administrative module (AM), only tuples from global hunted groups are valid. In the remote switching module (RSM), only tuples from the local hunted groups are valid except when the RSM is in the stand-alone mode. In the stand-alone mode, tuples from global hunted groups are also valid. In the SM and in the RSM only tuples from local hunt group ports that exist in that SM or RSM are valid. In the SM, designated as the hunting SM, only tuples from global and local hunt group ports, belonging to that hunting SM, are valid. During stand-alone operation, global hunt groups are allowed in an RSM.

4. ERROR HANDLING

If the key values, calculated by the audit and retrieved by the DA primitive, are not the same, the value of the key retrieved by the DA primitive is corrected to agree with the audit value.

Tuples, in the RLPPLLNST that are erroneous because they have no corresponding tuple in the RLRT_MHG, are in the wrong processor, or have the wrong hunt type, are deleted.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	grpnum of this tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	grpnum of this tuple.
<i>Dump =</i>	NA.

Table 2 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The group is of the wrong hung type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in rIPPLLNST (that is, RTH PPLMHG) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPPLLNST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	grpnum of this tuple.
<i>Dump =</i>	NA.

Table 3 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the SM and in the RSM only tuples from local hunt group ports that exist in that SM or RSM are valid. In the SM designated as hunting SM, only tuples from global and local hunt group ports belonging to that hunting SM are valid. During stand-alone operation, global hunt groups are allowed in an RSM.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPPLLNST tuple.
<i>Bad Data =</i>	Module number for local group, 0 if global.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	grpnum of this tuple.
<i>Dump =</i>	NA.

Table 4 Error Code: PPLKEY

Error Code:	PPLKEY
Error Description:	An rIPPLLNST tuple has bad data in the key (grpnum) field for a multiline hunt

	group.
Possible Error Effect:	Unable to preempt the lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rIPPLLNST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rIPPLLNST tuple.
<i>Good Data</i> =	Group number of the MLH group.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	rIPPLLNST tuple.

PPLNST

Software Release: 5E10 and later

1. PRECEDENCE AND PREEMPTION LINE STATUS (PPLNST) AUDIT

The precedence and preemption (PP) feature of the automatic voice network (AUTOVON) can be tailored for applications through individualized dialing plans and feature-specific data. Precedence and preemption features are assigned to members of terminal groups; each user in a precedence and preemption group may or may not have access to the AUTOVON network. There the lowest precedence level. Precedence calls are calls having precedence levels of 0 (flash override), 1 (flash), 2 (immediate), 3 (priority), and 4 (routine), where 0 is the highest precedence level.

The precedence and preemption feature is a modular feature designed within the framework used by the business and residence custom services (BRCS) feature. It executes under the control of the feature execution (FEX) system, which supports the concurrent execution of multiple subprocesses within a single terminal process. A record of the precedence level must be maintained for each port active on a call. Per-call dynamic data, including the precedence level, is stored in the process control block linkage area relation (RLPCBLA). A routing and terminal allocation primitive also maintains the record of the precedence level of each hunted port. Four dynamic relations keep track of precedence and preemption: the precedence and preemption line status relation (RLPPLNST), the precedence and preemption large line status relation (RLPPLLNST), the precedence and preemption trunk status relation (RLPPTKST), and the precedence and preemption large trunk status relation (RLPPLTKST). Four audits check these relations; the audits are named for the relations they check.

The PPLNST audit is part of the routing and terminal allocation audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environments in the segmented modes.

2. DATA STRUCTURES

The data structures the PPLNST audit checks are the dynamic rPPLNST and the static multi-line hunt group relation (RLRT_MHG). The fields checked include the key field, the hunt type, and the processor (module) number.

3. AUDIT CHECKS

The PPLNST audit calls the static driver (DRSTATIC) to perform a dynamic-to-static existence check. The DRSTATIC driver steps through RLPPLNST and retrieves successive tuples sequentially. For each tuple retrieved, the semantic check function does the following.

- (a) Obtains the key of the tuple of rPPLNST by using the audit get index primitive (AUgetindx), and compares the key value to the key value retrieved by the data administration (DA) primitive DAGtpplnst.
- (b) Uses the correct key to read rLRT_MHG, and verify that the hunt type is regular (RTH_PPSMHG).
- (c) Checks whether RLRT_MHG is being updated if the hunt type is not correct. If RLRT_MHG is not being updated, there is an error; the audit reports the error. If it is being updated, control returns to DRSTATIC and another tuple is checked.
- (d) Checks the module number of the tuple in RLRT_MHG if the hunt type is correct. In the administrative module (AM), only tuples from global hunted groups are valid. In the remote switching module (RSM), only tuples from local hunted groups are valid except when the RSM is in the stand-alone mode. In the stand-alone mode, tuples from global hunted groups are also valid.
- (e) In the switching module (SM) and in the RSM, only tuples from local hunt group ports that exist in that SM or RSM are valid. In the SM, designated as the hunting SM, only tuples from global and local hunt group ports belonging to that hunting SM are valid. During stand-alone operation, global hunt groups are allowed in a RSM.

4. ERROR HANDLING

If the key values retrieved by the audit and the DA primitive are not the same, the value of the key value retrieved by the DA primitive is corrected to agree with the audit-obtained value.

Tuples in RLPLNST that are erroneous because they have no corresponding tuple in RLRT_MHG, are in the wrong processor, or have the wrong hunt type, are deleted.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	riPPLNST tuple.

Table 2 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The group is of the wrong hung type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in riPPLNST (that is, RTH PPSMHG) and the tuple could not have retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the riPPLNST tuple.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	riPPLNST tuple.

Table 3 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the SM and in the RSM only tuples from local hunt group ports that exist in that SM or RSM are valid. In the SM designated as hunting SM, only tuples from global and local hunt group ports belonging to that hunting SM are valid. During stand-alone operation, global hunt groups are allowed in an RSM.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of riPPLNST tuple.
<i>Bad Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	riPPLNST tuple.

Table 4 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	An rIPPLNST tuple has bad data in the key (grpnum) field for a multi-line hunt group.
Possible Error Effect:	Unable to preempt the lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rIPPLNST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rIPPLNST tuple.
<i>Good Data</i> =	Group number of the multi-line hunt group.
<i>Logical Index</i> =	Group number of the multi-line hunt group.
<i>Dump</i> =	rIPPLNST tuple.

PPLTKST**Software Release:** 5E10 and later**1. PRECEDENCE AND PREEMPTION LARGE TRUNK STATUS (PPLTKST) AUDIT**

The precedence and preemption (PP) feature of the automatic voice network (AUTOVON) can be tailored for applications through individualized dialing plans and feature-specific data. Precedence and preemption features are assigned to members of terminal groups; each user in a precedence and preemption group may or may not have access to the AUTOVON network. There are five levels of precedence. Routine calls occur at precedence level 4, the lowest precedence level. Precedence calls are calls having precedence levels 0 (flash override), 1 (flash), 2 (immediate), 3 (priority), and 4 (routine), where 0 is the highest precedence level.

The precedence and preemption feature is a modular feature designed within the framework used by the business and residence custom services (BRCS) feature. It executes under the control of the feature execution (FEX) system, which supports the concurrent execution of multiple subprocesses within a single terminal process. A record of the precedence level must be maintained for each port active on a call. Per-call dynamic data, including the precedence level, is stored in the process control block linkage area relation (RLPCBLA). A routing and terminal allocation primitive also maintains the record of the precedence level of each hunted port. Four dynamic relations keep track of precedence and preemption: the precedence and preemption line status relation (RLPPLNST), the precedence and preemption large line status relation (RLPPLLNST), the precedence and preemption trunk status relation (RLPPTKST), and the precedence and preemption large trunk status relation (RLPPLTKST). Four audits check these relations; the audits are named for the relations they check.

The PPLTKST audit is part of the routing and terminal allocation audits subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The data structures the PPLTKST audit checks are the dynamic RLPPLTKST and the static trunk group relation (RLRT_TRKG). The fields checked include the key field, the hunt type, and the processor (module) number.

3. AUDIT CHECKS

The PPLTKST audit calls the static driver (DRSTATIC) to perform a dynamic-to-static existence check. The DRSTATIC driver steps through RLPPLTKST and retrieves tuples sequentially. For each tuple retrieved, the semantic check function does the following.

- (a) The audit obtains the correct key of the tuple of RLPPLTKST by using the audit get index primitive (AUgetindx) and compares the key value with the key value retrieved by the data administrator (DA) primitive DAgtppltk. If they do not match, an error is reported.
- (b) The audit uses the correct key to read rIRT_TRKG and verify that the hunt type is linear (RTH_PLFLIN or RTH_PLBLIN) or first-in, first-out (RTH_PLFIFO).
- (c) If the hunt type is not correct, the audit checks whether rIRT_TRKG is being updated. If it is not being updated, there is an error; the audit reports the error. If it is being updated, control returns to DRSTATIC and another tuple is checked.
- (d) If the hunt type is correct, the audit checks the module number of the tuple in rIRT_TRKG. In the administrative module (AM), only tuples from global hunted groups are valid. In the remote switching module (RSM), only tuples from local hunted groups are valid except when the RSM is in the stand-alone mode. In the stand-alone mode, tuples from global hunted groups are also valid.

4. ERROR HANDLING

If the key values retrieved by the audit and the DA primitive are not the same, the value of the key retrieved by the DA primitive is corrected to agree with the audit-obtained value.

Tuples in RLPPLTKST that are erroneous because they have no corresponding tuple in RLRT_TRKG, are in the wrong processor, or have the wrong hunt type are deleted.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	riPPLTKST tuple.

Table 2 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the AM, only tuples from global trunk groups should be present. In an RSM, only tuple from local trunk groups on that RSM should be present except during stand-alone operation when tuples from global trunk groups are allowed.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of riPPLTKST tuple.
<i>Bad Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	riPPLTKST tuple.

Table 3 Error Code: PPLKEY

Error Code:	PPLKEY
Error Description:	An riPPLTKST tuple has bad data in the key (grpnum) field for a large hunted port group.
Possible Error Effect:	Unable to preempt the lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the riPPLTKST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the riPPLTKST tuple.
<i>Good Data</i> =	Group number of the hunted port group.
<i>Logical Index</i> =	Group number of the hunted port group.
<i>Dump</i> =	riPPLTKST tuple.

Table 4 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in riPPLTKST (that is, RTH_PLFLIN, RTH_PLBLIN, or RTH_PLFIFO) and the tuple could not

	be retrieved.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rIPPLTKST tuple.

PPSAST

Software Release: 5E10 and later

1. PRECEDENCE AND PREEMPTION SUBADDRESS STATUS (PPSAST) AUDIT

The purpose of the audit is to perform dynamic-to-static existence checks between RLPPSAST and RLIS_LINE. For each rPPSAST tuple, there must be an rLIS_LINE tuple. If an rLIS_LINE tuple does not exist, but a rPPSAST tuple exists, then a tuple should be deleted from RLPPSAST. In addition to tuple existence checks, the semantic check will be made on the RLPPSAST relation. This audit also performs the key check and validates the RLPPSAST data.

The PPSAST audit runs in the segmented mode and operates in the OSDS-M environment.

2. DATA STRUCTURES

PPSAST audit uses the RLPPSAST relation.

3. AUDIT CHECKS

The PPSAST audit uses the static driver (DRSTATIC) to step through the RLPPSAST relation and sequentially retrieve tuples. The DRSTATIC driver invokes a semantic check function that performs the following checks:

- (1) The audit performs the existence check between the dynamic rPPSAST and rLIS_LINE tuples.
- (2) The audit performs the following checks for precedence flags:
 - (a) A busy subaddress is checked to verify that only one or no precedence flag is set in the rPPSAST.
 - (b) An idle subaddress is checked to verify that no precedence flag is set in the rPPSAST.

4. ERROR HANDLING

When the PPSAST audit encounters any errors, an error message is printed and an attempt is made to recover from the error. Audit control schedules the related audit LTMSTAT to run in the elevated mode. An assert will be fired if the database delete fails.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DUPTUP

Error Code:	DUPTUP
Error Description:	The pp_stat feature bit of rLIS_LINE static tuple, used to indicate whether PP feature is on or not, is found to be set to DBNO by audit (is_line_buf.pp_stat == DBNO). This means an extra rPPSAST tuple exists.
Possible Error Effect:	Waste of resources.
Corrective Action Taken:	The extra rPPSAST tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the extra rPPSAST tuple in rPPSAST.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rPPSAST tuple.
<i>Dump =</i>	rPPSAST tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rPPSAST tuple was detected.
Possible Error Effect:	ASSERT failures when failure to insert tuple in the database.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the extra tuple in rPPSAST.
<i>Bad Data =</i>	Valid values are: 1 = If the PP feature bit is set to DBNO. 0 = If the rIIS_LINE tuple is missing.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rPPSAST tuple.
<i>Dump =</i>	rPPSAST tuple.

Table 3 Error Code: INVPPFLG

Error Code:	INVPPFLG
Error Description:	A PP flag is marked busy when it is supposed to be idle or too many PP flags are marked BUSY in the rPPSAST tuple for a requested Sub address.
Possible Error Effect:	This will cause lost resources and deny preemption from happening.
Corrective Action Taken:	An error will be reported by the PPSAST audit and a recovery action will be performed to correct this error. To correct this error, PP flags in rPPSAST will be initialized to idle for a requested Sub address.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rPPSAST tuple.
<i>Bad Data =</i>	Sub address number.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of PPSAST tuple name.
<i>Dump =</i>	rPPSAST tuple.

Table 4 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The corresponding rLTMSTAT dynamic tuple does not exist.
Possible Error Effect:	Inconsistency between rPPSAST and rLTMSTAT and possible ASSERT failures. This also leads to call denied.
Corrective Action Taken:	Audit control will schedule LTMSTAT audit as a related audit to correct this problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of missing tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rPPSAST tuple.
<i>Dump =</i>	rPPSAST tuple.

Table 5 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	An Invalid key value if found in the rPPSAST tuple.
Possible Error Effect:	This will deny call from terminating on requested DN and stop preemption from happening. Assert failures to report key error.
Corrective Action Taken:	A key error will be reported and corrected by the PPSAST audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rPPSAST tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Correct key value.
<i>Logical Index =</i>	Key of rPPSAST tuple name.

<i>Dump</i> = riPPSAST tuple.

PPTKST

Software Release: 5E10 and later

1. PRECEDENCE AND PREEMPTION TRUNK STATUS (PPTKST) AUDIT

The precedence and preemption (PP) feature of the automatic voice network (AUTOVON) can be tailored for applications through individualized dialing plans and feature-specific data. Precedence and preemption features are assigned to members of terminal groups; each user in a precedence and preemption group may or may not have access to the AUTOVON network. There are five levels of precedence. Routine calls occur at precedence level 4, the lowest precedence level. Precedence calls are calls having precedence levels 0 (flash override), 1 (flash), 2 (immediate), 3 (priority), and 4 (routine), where 0 is the highest precedence level.

The precedence and preemption feature is a modular feature designed within the framework used by the business and residence custom services (BRCS) feature. It executes under the control of the feature execution (FEX) system, which supports the concurrent execution of multiple subprocesses within a single terminal process. A record of the precedence level must be maintained for each port active on a call. Per-call dynamic data, including the precedence level, is stored in the process control block linkage area relation (RLPCBLA). A routing and terminal allocation primitive also maintains the a record of the precedence level of each hunted port. Four dynamic relations keep track of precedence and preemption: the precedence and preemption line status relation (RLPPLNST), the precedence and preemption large line status relation (RLPPLLNST), the precedence and preemption trunk status relation (RLPPTKST), and the precedence and preemption large trunk status relation (RLPPLTKST). Four audits check these relations; the audits are named for the relations they check.

The PPTKST audit is part of the routing and terminal allocation audits subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The data structures the PPTKST audit checks are the dynamic RLPPTKST and the static trunk group relation (RLRT_TRKG). The fields checked include the key field, the hunt type, and the processor (module) number.

3. AUDIT CHECKS

The PPTKST audit calls the static driver (DRSTATIC) to perform a dynamic-to-static existence check. The DRSTATIC driver steps through RLPPTKST and retrieves tuples sequentially. For each tuple retrieved, the semantic check function does the following.

- (a) The audit obtains the correct key of the tuple of RLPPTKST by using the audit get index primitive (AUgetindx) and compares the key value with the key value retrieved by the data administrator (DA) primitive DAgtptpst. If they do not match, an error is reported.
- (b) The audit uses the correct key to read rLRT_TRKG and verify that the hunt type is linear (RTH_PRFLIN or RTH_PRBLIN) or first-in, first-out (RTH_PRFIFO).
- (c) If the hunt type is not correct, the audit checks whether rLRT_TRKG is being updated. If it is not being updated, there is an error; the audit reports the error. If it is being updated, control returns to DRSTATIC and another tuple is checked.
- (d) If the hunt type is correct, the audit checks the module number of the tuple in RLRT_TRKG. In the administrative module (AM), only tuples from global hunted groups are valid. In the remote switching module (RSM), only tuples from local hunted groups are valid except when the RSM is in the stand-alone mode. In the stand-alone mode, tuples from global hunted groups are also valid.

4. ERROR HANDLING

If the key values retrieved by the audit and the DA primitive are not the same, the value of the key retrieved by the DA primitive is corrected to agree with the audit-obtained value.

Tuples in RLPPTKST that are erroneous because they have no corresponding tuple in RLRT_TRKG, are in the wrong processor, or have the wrong hunt type are deleted.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	rlPPTKST tuple.

Table 2 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the AM, only tuples from global trunk groups should be present. In an RSM, only tuples from local trunk groups on that RSM should be present except during stand-alone operation when tuples from global trunk groups are allowed.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rlPPTKST tuple.
<i>Good Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rlPPTKST tuple.

Table 3 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	An rlPPTKST tuple has bad data in the key (grpnum) field for a regular hunted port group.
Possible Error Effect:	Unable to preempt the lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rlPPTKST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rlPPTKST tuple.
<i>Good Data</i> =	Group number of the port group.
<i>Logical Index</i> =	Group number of port group.
<i>Dump</i> =	rlPPTKST tuple.

Table 4 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in rlPPTKST (that is, RTH_PRFLIN, RTH_PRBLIN, or RTH_PRFIFO) and the tuple could not

	be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rIPPTKST tuple.

PRIASGN-A

Software Release: 5E10 - 5E11

1. PRIMARY RATE INTERFACE D-CHANNELS ASSIGNED TO SWITCHING MODULE (PRIASGN) AUDIT

TMpri_assigned[] is an array on the operational kernel process (OKP) that tells which switching modules (SMs) have D-channel primary rate interfaces (PRIs) on them, not including operator services position system (OSPS) D-channels. The data is audited by the PRI assigned (PRIASGN) audit.

The PRIASGN audit is part of the terminal maintenance audits subgroup, and runs in the OKP environment in the segmented mode.

2. DATA STRUCTURES

in RLSM_DCHN and TMpri_assigned[]. Other data referenced includes a global flag AUpri_flag which is set by the trigger function.

Information for all SM D-channels is maintained in the administrative module in TMpri_assigned[]. This information is audited by the PRIASGN audit, which compares each entry in TMpri_assigned[] with data in RLSM_DCHN relation.

The TMpri_assigned[] array contains six longs, which represent whether each of the 192 SMs have at least one D-channel PRI. Note that the SMs range from 1 to 192, although the bits range from 0 to 191 (representing SM 1 in bit 0, and so forth).

3. AUDIT CHECKS

The PRIASGN audit uses the DRSTATIC audit driver and checks the relation RLSM_DCHN. Reading each rISM_DCHN tuple found in the relation, the audit sets a bit in a local array corresponding to the SM number. When all tuples have been read, the local array is compared with TMpri_assigned[]; if any of the six longs is incorrect, they are changed and the audit error UNEQUAL is reported.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	A long in the array TMpri_assigned[] is incorrect, according to the local array AUpri_work[]. Note that the SMs range from 1 to 192, although the bits range from 0 to 191 (representing SM 1 in bit 0, etc).
Possible Error Effect:	When the array is incorrect, there may be failures in the functions accessing it.
Corrective Action Taken:	The long in the array TMpri_assigned[] is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TMpri_assigned[0] long where the error was found.
<i>Bad Data =</i>	The incorrect value in the long from the TMpri_assigned[0] array.
<i>Good Data =</i>	The correct value in the long from the temporary work array.
<i>Logical Index =</i>	The number of the incorrect long in the TMpri_assigned[0] array.
<i>Dump =</i>	NA.

PRIASGN-B

Software Release: 5E12 - 5E13

1. PRIMARY RATE INTERFACE D-CHANNELS ASSIGNED TO SWITCHING MODULE (PRIASGN) AUDIT

TMpridstatus[] is an array on the operational kernel process (OKP) that tells which primary rate interface groups (PRIs) have D-channels provisioned. The data is audited by the PRI assigned (PRIASGN) audit against the static relation RLPRI_SM when the entry in the array says either primary or secondary D-channel port is provisioned.

The PRIASGN audit is part of the terminal maintenance audits subgroup, and runs in the OKP environment in the segmented mode.

2. DATA STRUCTURES

There are no data structures used by this audit, just a TM array.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	Either primary or secondary D-channel port in the array TMpridstatus[] is listed as provisioned, however no rPRI_SM tuple exists for the associated PRI group.
Possible Error Effect:	PRI groups that do not exist may appear to exist.
Corrective Action Taken:	The entry in the array TMpridstatus[] is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the array entry in TMpridstatus[] where the error was found.
<i>Bad Data</i> =	If 1, the primary D-channel port is listed as being provisioned. Otherwise, it is not.
<i>Good Data</i> =	If 1, the secondary D-channel port is listed as being provisioned. Otherwise, it is not.
<i>Logical Index</i> =	The index into the TMpridstatus[] array (the PRI group).
<i>Dump</i> =	The value in the array entry in TMpridstatus[].

Table 2 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index to the TMpridstatus[].
<i>Dump</i> =	NA.

PRIASSIGNED

Software Release: 5E10 and later

1. PRIMARY RATE INTERFACE D-CHANNELS ASSIGNED TO SWITCHING MODULE (PRIASSIGNED) AUDIT

TMpri_assigned[] is an array on the operational kernel process (OKP) that tells which switching modules (SMs) have D-channel primary rate interface (PRIs) on them, not including OSPS D-channels. The data is audited by the PRI assigned (PRIASSIGNED) audit.

The PRIASSIGNED audit is part of the Terminal Maintenance audits subgroup, and runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The PRIASSIGNED audit references the SM D-channel information in RLISM_DCHN and TMpri_assigned[]. Other data referenced includes a global flag AUpri_flag which is set by the trigger function.

Information for all SM D-channels is maintained in the administrative module in TMpri_assigned[]. This information is audited by the PRIASSIGNED audit which compares each entry in TMpri_assigned[] with data in RLISM_DCHN relation.

The TMpri_assigned[] array contains six longs, which represent whether each of the 192 SMs have at least one D-channel PRI. Note that the SMs range from 1 to 192, although the bits range from 0 to 191 (representing SM 1 in bit 0, and so forth).

3. AUDIT CHECKS

The PRIASSIGNED audit uses the DRSTATIC audit driver and checks the relation RLISM_DCHN. Reading each rISM_DCHN tuple found in the relation, the audit sets a bit in a local array corresponding to the SM number. When all tuples have been read, the local array is compared with TMpri_assigned[]; if any of the six longs are incorrect, they are changed and the audit error UNEQUAL is reported.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s) and those generated by the DRSTATIC driver.

Table 1 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	A long in the array TMpri_assigned[] is incorrect, according to the local array AUpri_work[]. Note that the SMs range from 1 to 192, although the bits range from 0 to 191 (representing SM 1 in bit 0, and so forth).
Possible Error Effect:	When the array is incorrect, there may be failures in the functions accessing it.
Corrective Action Taken:	The long in the array TMpri_assigned[] is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TMpri_assigned[0] long where the error was found.
<i>Bad Data =</i>	The incorrect value in the long from the TMpri_assigned[0] array.
<i>Good Data =</i>	The correct value in the long from the temporary work array.
<i>Logical Index =</i>	The number of the incorrect long in the TMpri_assigned[0] array.
<i>Dump =</i>	NA.

PRIDSTAT

Software Release: 5E13 and later

1. PRIMARY RATE INTERFACE PRI D-CHANNEL PROVISIONING STATUSES (PRIDSTAT) AUDIT

The PRIDSTAT audit is developed for the D-Channel Expansion (DCEX) feature (99-5e-4373). The AM dynamic relation RLPRIDSTAT contains information on PRI D-channel provisioning and IS/OOS statuses. The data is audited by the PRIDSTAT audit.

The PRIDSTAT audit is part of the terminal maintenance audits subgroup, and runs in the OKP environment in the segmented mode.

2. DATA STRUCTURES

For details on the fields contained in the relations, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	If the primary D-channel is provisioned (rlPRI_SM.dprim_port is non-null), the dynamic relation should agree (rlPRIDSTAT.dprim_prov should be DBYES). If the primary D-channel is not provisioned (rlPRI_SM.dprim_port is null), the dynamic relation should agree (rlPRIDSTAT.dprim_prov should be DBNO).
Possible Error Effect:	The information displayed on the MCC terminal may not be correct, assert may be fired.
Corrective Action Taken:	The audit will correct the primary D-channel provisioning status (rlPRIDSTAT.dprim_prov).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rlPRIDSTAT tuple.
<i>Bad Data =</i>	The dprim_prov field of rlPRIDSTAT tuple (rlPRIDSTAT.dprim_prov).
<i>Good Data =</i>	The dprim_port field of static rlPRI_SM tuple (rlPRI_SM.dprim_port).
<i>Logical Index =</i>	The rlPRIDSTAT tuple logical key (rlPRIDSTAT.prigrp).
<i>Dump =</i>	The rlPRIDSTAT tuple being audited.

Table 2 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The SM number in the dynamic relation (rlPRIDSTAT.sm) did not match the SM number in the static relation (rlPRI_SM.sm).
Possible Error Effect:	The information displayed on the MCC terminal may not be correct, assert may be fired.
Corrective Action Taken:	The audit will correct the sm field of the rlPRIDSTAT tuple using the data in sm field of static rlPRI_SM tuple.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the rIPRIDSTAT tuple (rIPRIDSTAT.prigrp).
<i>Bad Data</i> =	The SM of rIPRIDSTAT tuple (rIPRIDSTAT.sm).
<i>Good Data</i> =	The SM of rIPRI SM tuple (rIPRI SM.sm).
<i>Logical Index</i> =	The rIPRIDSTAT tuple logical key (rIPRIDSTAT.prigrp).
<i>Dump</i> =	The rIPRIDSTAT tuple being audited.

Table 3 Error Code: RDATA_BAD

Error Code:	RDATA_BAD
Error Description:	If the secondary D-channel is provisioned (rIPRI_SM.dsec_port is non-null), the dynamic relation should agree (rIPRIDSTAT.dsec_prov should be DBYES). If the secondary D-channel is not provisioned (rIPRI_SM.dsec_port is null), the dynamic relation should agree (rIPRIDSTAT.dsec_prov should be DBNO).
Possible Error Effect:	The information displayed on the MCC terminal may not be correct, assert may be fired.
Corrective Action Taken:	The audit will correct the primary D-channel provisioning status (rIPRIDSTAT.dsec_prov).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of rIPRIDSTAT tuple.
<i>Bad Data</i> =	The dsec_prov field of rIPRIDSTAT tuple (rIPRIDSTAT.dsec_prov).
<i>Good Data</i> =	The dsec_port field of static rIPRI SM tuple (rIPRI_SM.dsec_port).
<i>Logical Index</i> =	The rIPRIDSTAT tuple logical key (rIPRIDSTAT.prigrp).
<i>Dump</i> =	The rIPRIDSTAT tuple being audited.

PRIGRP

Software Release: 5E10 and later

1. PRIMARY RATE INTERFACE GROUP (PRIGRP) AUDIT

The primary rate interface group (PRIGRP) audit audits the RLPRIGRP relation, introduced by the non-facility associated signaling (NFAS) and D-channel backup (DCBU) features. This relation is used by all non operator services position system (OSPS) primary rate interface (PRI).

An NFAS provides a Bellcore compatible PRI defined as $nB + D$, where $n=1-479$. PRI is a collection of B-channels and a signaling D-channel. All B-channels are controlled by that D-channel. There are two types of PRI. In the classic definition, n is a maximum of 23 and all B-channels are to be on the same digital signal level 1 (DS1) facility known as digital facilities interface (DFI). This is called a custom PRI. In a standard PRI definition, n is a maximum of 479 and the B-channels can spread over multiple facilities with maximum up to 20 facilities within a single SM. The D-channels must reside on the 24th channel of a DFI and must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

DCBU is a Bellcore compatible PRI with $mB + 2D$, where $m = 1 - 478$, and there are 2 D-channels. One D-channel is designated as "active" and the other D-channel is designated as "standby". Should the "active" D-channel fail due to hardware, layer 2 or layer 3, then the "standby" D-channel should become the "active" D-channel. Both D-channels must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

Both NFAS and DCBU apply only to PRIs that can only interface to customer premise equipments (CPEs) (includes the 5ESS[®] switch "User side PRIs") which are compatible with Bellcore's TR-1268. The existing $23B + D$ PRIs (also known as, custom PRI), will not support NFAS and DCBU.

The PRIGRP audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode and it checks for a missing rIPRISTAT tuple. If one is found, it is inserted and the quarantine bit of the rIPRISTAT is set to initialize both the primary and secondary D ports and also the signaling port.

2. DATA STRUCTURES

The primary data structure for this audit is the RLPRIGRP relation. It is a static, access type of hashed relation keyed by *prigrp*. The relation defines the general data that compose a PRI group. It gives the signaling type, D-channel ports, the total number of B-channels assigned to this PRI group and all the possible facilities.

An rIPRIGRP tuples has a semantic one to one linkage to the rIPRISTAT tuples through *prigrp* field. There must be one dynamic rIPRISTAT tuple per rIPRIGRP tuple.

3. AUDIT CHECKS

The PRIGRP audit invokes the DRSTATIC driver to make the static to dynamic checks for each rIPRIGRP tuple. It also uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given.

LOSTTUP = Check for missing rIPRISTAT tuple.

RL_MISMATCH = Check the rIPRISTAT tuple is the same that was just inserted.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Audit tried to read the dynamic rIPRISTAT tuple with its key, <i>prigrp</i> , equal to the key of the static rIPRIGRP tuple, <i>prigrp</i> and failed.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	The missing dynamic rIPRISTAT tuple is inserted through a TM interface.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRIGRP tuple under audit.
<i>Bad Data</i> =	The return value from the read of the rIPRISTAT tuple that failed.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIPRIGRP tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRIGRP tuple being audited.

Table 2 Error Code: RL_MISMATCH

Error Code:	RL_MISMATCH
Error Description:	Audit found a missing dynamic rIPRISTAT tuple and called a TM interface to insert it. Then it tried to access the inserted tuple, but failed.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	This error will be reported and audit will return. But this will cause the PRISTAT audit to run as a related audit and correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRIGRP tuple being audited.
<i>Bad Data</i> =	The <i>prigrp</i> field of rIPRISTAT tuple that audit failed to access.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIPRIGRP tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRIGRP tuple being audited.
<i>Dump 2</i> =	The rIPRISTAT tuple being audited.

PRISM

Software Release: 5E13 and later

1. PRIMARY RATE INTERFACE TO SM MAPPING (PRISM) AUDIT

The PRISM audit is developed for the D-Channel Expansion (DCEX) feature (99-5e-4373).

The AM static relation RLPRI_SM contains primary rate interface to SM mapping information.

The PRISM audit is part of the terminal maintenance audits subgroup, and runs in the OKP environment in the segmented mode.

2. DATA STRUCTURES

For details on the fields contained in the relations, refer to the 5ESS[®] Translations Data Manual (235-600-1xx) and the 5ESS[®] Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	For every static tuple in RLPRI_SM relation, there should exist a corresponding dynamic tuple in the RLPRIDSTAT relation. The dynamic rIPRIDSTAT tuple is missing.
Possible Error Effect:	The expected dynamic data tuple is missing, assert may be fired.
Corrective Action Taken:	The audit will insert the missing rIPRIDSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	N/A.
<i>Bad Data =</i>	RLPRIDSTAT
<i>Good Data =</i>	The rIPRI_SM tuple logical key field (rIPRI_SM.prigrp).
<i>Logical Index =</i>	The logical key of the missing rIPRIDSTAT tuple(rIPRI_SM.prigrp).
<i>Dump =</i>	The rIPRI_SM tuple being audited.

Table 2 Error Code: RDATA_BAD

Error Code:	RDATA_BAD
Error Description:	The rIPRIDSTAT tuple was not inserted successfully.
Possible Error Effect:	The expected dynamic data tuple is missing, assert may be fired.
Corrective Action Taken:	The audit will report the error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0
<i>Bad Data =</i>	RLPRIDSTAT
<i>Good Data =</i>	The rIPRI_SM tuple logical key field (rIPRI_SM.prigrp).
<i>Logical Index =</i>	The rIPRI_SM tuple logical key field (rIPRI_SM.prigrp).
<i>Dump =</i>	The rIPRI_SM tuple being audited.

PRISTAT-A

Software Release: 5E10 - 5E11

1. PRIMARY RATE INTERFACE GROUP STATUS (PRISTAT) AUDIT

The primary rate interface Group status (PRISTAT) audit audits the RLPRISTAT relation, introduced by the non-facility associated signaling (NFAS) and D-channel backup (DCBU) features. This relation is used by all non operator services position system (OSPS) primary rate interfaces (PRIs).

An NFAS provides a Bellcore compatible PRI defined as $nB + D$, where $n=1-479$. PRI is a collection of B-channels and a signaling D-channel. All B-channels are controlled by that D-channel. There are two types of non-OSPS PRI. In the classic definition, n is a maximum of 23 and all B-channels are to be on the same digital signal level 1 (DS1) facility known as digital facilities interface (DFI). This is called a custom PRI. In a standard PRI definition, n is a maximum of 479 and the B-channels can spread over multiple facilities with maximum up to 20 facilities within a single SM. The D-channels must reside on the 24th channel of a DFI and must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-Channels are in the same PRI group).

DCBU is a Bellcore compatible PRI with $mB + 2D$, where $m = 1 - 478$, and there are 2 D channels. One D-channel is designated as "active" and the other D channel is designated as "standby". Should the "active" D channel fail due to hardware, layer 2 or layer 3, then the "standby" D-channel should become the "active" D-channel. Both D-channels must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

Both NFAS and DCBU apply only to PRIs that can only interface to customer premise interfaces (CPEs) (includes the 5ESS[®] switch "User side PRIs") which are compatible with Bellcore's TR-1268. The existing $23B + D$ PRIs (also known as custom PRI) will not support NFAS and DCBU.

The PRISTAT audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

The primary data structure for this audit is the RLPRISTAT relation. It is a dynamic, indexed non-sequential relation keyed by *prigrp*. There must be one rIPRISTAT tuple per rIPRIGRP tuple which is a static relation. The rIPRISTAT tuple is created through a trigger whenever a new rIPRIGRP tuple is inserted. The data is centralized per PRI group and needs to reside in a dynamic relation since it is run-time dependent and updated frequently. Along with storing the possible two D-ports, there is the current active D-channel, information on assigning call reference values (CRVs), timers, multi-frame establish and release job_ids, other process IDs, boolean fields, and so forth.

All rIPRISTAT tuples has a owner-member linkages to the rIDALB tuples representing all calls on the PRI group. Refer to Figure 1 .

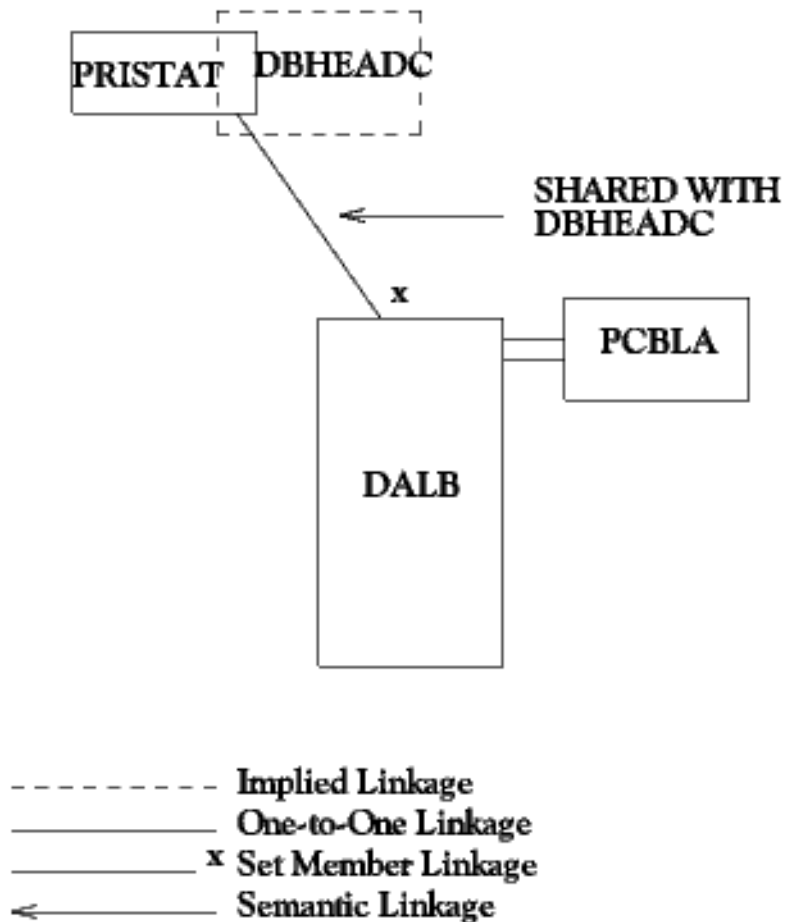


Figure 1 (PRISTAT-1) RLPRISTAT Linkages

3. AUDIT CHECKS

The PRISTAT audit invokes the DRINDEX driver to make the dynamic to static, linkages, semantic and special selective initialization checks for each rIPRISTAT tuple. It also uses the following error codes. This listing gives a basic indication of the order the checks are made. For details regarding each error code see the comment blocks given.

EXTRATUP	= Check for extra rIPRISTAT tuple.
R_PORT	= Check dprim_port field against the same field in static rIPRIGRP tuple.
GPORT	= Check dsec_port field against the same field in static rIPRIGRP tuple.
INV	= Check pri_type field against the same field in static rIPRIGRP tuple.
INV_IDLE	= Check dprim_int_id field against the interf_id field in static rIDCHNPRI tuple.
NOPORTLA	= Check that audit got correct rPORTLA tuple for the primary D-port.
SGNODD	= Failure to read an rIDCHNPRI tuple for the primary D-port.
INV_STATE	= Check dsec_int_id field against the interf_id field in static rIDCHNPRI tuple.

LOST_PORTLA	= Check that audit got correct rIORTLA tuple for the secondary D-port.
ODDLOSTUP	= Failure to read an rIDCHNPRI tuple for the secondary D-port.
MKNULL	= Check for null signaling D-port and block_call field to be set when both the D-ports are null.
CALLFLAG_BAD	= Check for block_call field to be set when the signaling D-port is null.
RL_MISMATCH	= Check for module field of the signaling D-port to be equal to the same field of either primary or secondary D-port depending upon which D-port equals the signaling D-port.
UNEQUAL	= Check for sig_int_id field to be equal to the interf_id field of rIDCHNPRI tuple of either primary or secondary D-port depending upon which D-port equals the signaling D-port.
MKBUSY	= Check for state field of rIORTLA tuple of either primary or secondary D-port depending upon which D-port equals the signaling D-port when the block_call bit is set to DBYES.
NONNULL	= Checks for sig_port field of rIORTLA tuple for either NFAS or DCBU case.
BADIS_DTE	= Check is_dte field against the same field in static rIDSLEQUIP tuple.
PORT	= Failure to read an rIDSLEQUIP tuple for the secondary D-port.
INV_TIMER	= Checks to verify t309_idx, t321_idx and tservice_idx timer fields.
INV_TIMER	= Checks to verify t321_idx and tservice_idx timer fields have null value when there is only one D Channel, that is NFAS case.
TIMERLST	= Verify that at least one of the D-port is on the WAIT list when t321_idx timer is running.
BADUSE	= Verify the t321_job_id field by making it sure that there is an ISPA job on the queue.
NOTZERO	= Verify the t321_idx field to be null when there is a valid t321_job_id field.
OUTOFRANGE	= Verify that the t321_job_id, mfe1_job_id, mfe2_job_id, mfr1_job_id, mfr2_job_id and bchan_job_id fields have a valid value with allowable range.
RELQKEY	= Verify the mfe1_job_id, mfe2_job_id, mfr1_job_id, mfr2_job_id and bchan_job_id fields by making it sure that there is an ISPA job on the queue. valid.
DEADPID	= Verify that the send_pid and hdlb_pid process id fields are
CDPRIM	= Verify that when dprimresp field is set to DBYES, the primary D-port is owned.
CDSEC	= Verify that when dsecresp field is set to DBYES, the secondary D-port is owned.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIS_DTE

Error Code:	BADIS_DTE
Error Description:	The <i>is_dte</i> field of the rIPRISTAT tuple did not match the same field of the static

	rIDSLEQUIP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>is_dte</i> field will be updated with the value found in the same field found in the rIDSLEQUIP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The key field of the rIDSLEQUIP tuple, <i>dsl</i> , which is also one of the D-ports of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>is_dte</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>is_dte</i> field of rIDSLEQUIP tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The <i>t321_job_id</i> field of the rIPRISTAT tuple has a valid ISPA job_id value indicating that there is a valid ISPA job on the queue ready to be run. But when audit checked this fact through a TM interface, using first the member field of the primary D-port and then the member field of the secondary D-port, it received failure both the times from the TM interface (other than GLSUCCESS).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>t321_job_id</i> field will be updated with the null value, TMNULLJOBID.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>port</i> field of the rIPORTLA tuple of the primary D-port.
<i>Bad Data</i> =	The <i>t321_job_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>port</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 3 Error Code: CALLFLAG_BAD

Error Code:	CALLFLAG_BAD
Error Description:	Audit found the error condition where the signaling D-port field, <i>sig_port</i> , is null, neither of the D-ports, primary or secondary, are owned by any process and yet the <i>block_call</i> bit is not set.
Possible Error Effect:	Attempt to start a PRI call may be made on this PRI group.
Corrective Action Taken:	The audit will set the <i>block_call</i> bit to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>block_call</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The member field of the <i>sig_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 4 Error Code: CDPRIM

Error Code:	CDPRIM
Error Description:	Audit found an error condition where the <i>dprimresp</i> field of the rIPRISTAT tuple is set to DBYES, the primary D-port is not null and it is not owned. The primary D-port must be owned, that is the <i>state</i> field of the rIPORTLA tuple of the primary D-port must not be DBPSIDLE, when the <i>dprimresp</i> field is set.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprimresp</i> field will be updated to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>member</i> field of the primary D-port of rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>dprimresp</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>state</i> field of the rIPORTLA tuple of the primary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.

<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.

Table 5 Error Code: CDSEC

Error Code:	CDSEC
Error Description:	Audit found an error condition where the <i>dsecresp</i> field of the rIPRISTAT tuple is set to DBYES, the secondary D-port is not null and it is not owned. The secondary D-port must be owned, that is the <i>state</i> field of the rIPORTLA tuple of the secondary D-port must not be DBPSIDLE, when the <i>dsecresp</i> field is set.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dsecresp</i> field will be updated to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>member</i> field of the secondary D-port of rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>dsecresp</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>state</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the secondary D-port.

Table 6 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	There are two field in the rIPRISTAT tuple that represent a process_id(pid)s, namely, <i>send_pid</i> and <i>hdlb_pid</i> . During auditing of these fields, audit could not verify these process ids, that is, the audit could not find these processes. As to which of these field is in error, refer to the good data field for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The process_id field in error will be updated with the null value, OSnullpid. As to which field is nulled, refer to the good data field.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>procno</i> field of the process ID field to be in error.
<i>Bad Data</i> =	The <i>pcrid</i> field of the process ID field to be in error.
<i>Good Data</i> =	Valid values are: 1 = The audit found error with the <i>send_pid</i> field of the rIPRISTAT tuple. 2 = The audit found error with the <i>hdlb_pid</i> field of the rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 7 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rIPRISTAT dynamic tuple without a corresponding static tuple where the rIPRISTAT does not have the static rIPRIGRP tuple with rIPRISTAT.prigrp = rIPRIGRP.prigrp.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	The rIPRISTAT tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPRISTAT tuple found to be extra.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 8 Error Code: GPORT

Error Code:	GPORT

Error Description:	The <i>dsec_port</i> field of the rIPRISTAT tuple did not match the same field in the static rIPRIGRP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dsec_port</i> will be updated with the value from the same field from static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The member field of the <i>dsec_port</i> field of rIPRIGRP tuple.
<i>Bad Data</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dsec_port</i> field of rIPRIGRP tuple.
<i>Logical Index</i> =	The module field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 9 Error Code: INV

Error Code:	INV
Error Description:	The <i>pri_type</i> field of the rIPRISTAT tuple did not match the same field in the static rIPRIGRP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>pri_type</i> will be updated with the value from the same field from static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>pri_type</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>pri_type</i> field of rIPRIGRP tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 10 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The <i>dprim_int_id</i> field of the rIPRISTAT tuple did not match the <i>interf_id</i> field of the static rIDCHNPRI tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprim_int_id</i> field will be updated with the value found in the <i>interf_id</i> field of the rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>dprim_int_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>interf_id</i> field of the rIDCHNPRI tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 11 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The <i>dsec_int_id</i> field of the rIPRISTAT tuple did not match the <i>interf_id</i> field of the static rIDCHNPRI tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dsec_int_id</i> field will be updated with the value found in the <i>interf_id</i> field of the rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>dsec_int_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>interf_id</i> field of the rIDCHNPRI tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 12 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	There are three timer index fields in the rIPRISTAT tuple, namely <i>t309_idx</i> ,

	<i>t321_idx</i> and <i>tservice_idx</i> that are audited. Whenever any of the field has a non null value, audit will make sure that there indeed is a timer pending. This is done through a TM interface which returns the key of the rIPRISTAT tuple and that key is checked against the key of the rIPRISTAT tuple being audited. If the keys do not match, this error is reported. As to which timer index triggered the error, refer to the address field for explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The timer index field in error will be null out(TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = The <i>t309_idx</i> field was in error. 2 = The <i>t321_idx</i> field was in error. 3 = The <i>tservice_idx</i> field was in error.
<i>Bad Data</i> =	The timer index field of rIPRISTAT tuple that was in error. Refer to the address field value and its explanation to determine which timer index field this is.
<i>Good Data</i> =	The pristat key returned by the TM interface and checked against the key of the rIPRISTAT tuple being audited.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 13 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	The <i>dsec_port</i> field of the rIPRISTAT tuple has been validated from the static rIPRGRP tuple. And yet audit could not find a valid rIPORTLA tuple for this D-port.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and return. This will cause the PORTLA audit to run as a related audit to PRISTAT audit and should correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>port</i> field of the rIPORTLA tuple that audit tried to access.
<i>Good Data</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple that was used as a key to access the rIPORTLA tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple that was unsuccessfully accessed.

Table 14 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	Audit found an error condition where the <i>block_call</i> bit was set but the D-port that equaled the signaling D-port was not owned by a process. Audit will report this error in either of the two cases where the <i>sig_port</i> is equal to the primary or the secondary D-port. As to which D-port it matched, refer to the <i>Good Data</i> field.
Possible Error Effect:	The call originations on this PRI group will be stopped till the <i>block_call</i> bit is cleared.
Corrective Action Taken:	Since audit does not know whether the <i>block_call</i> bit was set correctly or not, it will report this error, null out the member field of the <i>sig_port</i> , set the <i>block_call</i> bit to DBYES and will start recovery of both the D-ports. Recovery code will make correct decisions.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>block_call</i> bit of the rIPRISTAT tuple.

<i>Bad Data</i> =	The <i>state</i> field of rIPORTLA tuple of the D-port that equaled the signaling D-port, either primary or secondary. Refer to the good data field to decide the D-port.
<i>Good Data</i> =	Valid values are: 1 = The <i>sig_port</i> field was equal to the primary D-port and was not owned by any process. 2 = The <i>sig_port</i> field was equal to the secondary D-port and was not owned by any process.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the D-port, either primary or secondary. Refer to the good data to decide.

Table 15 Error Code: MKNUL

Error Code:	MKNUL
Error Description:	Audit found an error condition where both the D-ports of the rIPRISTAT tuple were null, that is TMNULL_PORT, and either the signaling D-port, <i>sig_port</i> , was not null or the <i>block_call</i> bit was not set.
Possible Error Effect:	Calls may originate on this PRI group without any D-port in service.
Corrective Action Taken:	The audit will report the error, null out the <i>sig_port</i> field, set the <i>block_call</i> bit to DBYES and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>block_call</i> field of the rIPRISTAT tuple.
<i>Bad Data</i> =	The member field of the <i>sig_port</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 16 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	This error represents various error conditions related to the signaling port, <i>sig_port</i> , field of the rIPRISTAT tuple. To determine what error condition, caused this error, find out the value in the good data field and compare it with the explanation given in this field for each value.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The member field of the <i>sig_port</i> field will be set to null value, the <i>block_call</i> bit will be set to DBYES, and both the D-ports will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: - GLFAIL or -1 means there was only one D-port available to this PRI group or in other words, this is an NFAS case. - If it is not GLFAIL (-1), then it is the <i>list</i> field of the primary D-port's rIPORTLA tuple.
<i>Bad Data</i> =	Valid values are: - If the address field above has a value of GLFAIL (-1), then this is <i>list</i> of the rIPORTLA tuple of the only D-port that this rIPRISTAT tuple has. - If the address field above does not have a value of GLFAIL (-1), then this is the <i>list</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Good Data</i> =	Valid values are:

	1 = The audit found an error where the <i>sig_port</i> field has a null value (TMNULL_PORT), neither primary nor the secondary D-port is owned by a process and either primary or secondary D-port is on the in service list (the <i>list</i> field of the rPORTLA is TMLTIS). 2 = The audit found an error where the <i>sig_port</i> field is equal to the primary D-port but the primary D-port is not on the in service list, that is the <i>list</i> field of the rPORTLA of the primary D-port is not TMLTIS. 3 = The audit found an error where the <i>sig_port</i> field is equal to the secondary D-port but the secondary D-port is not on the in service list, that is the <i>list</i> field of the rPORTLA of the secondary D-port is not TMLTIS. 4 = The audit found an error where the <i>sig_port</i> field has non null value and yet it is neither equal to the primary nor the secondary D-port.
<i>Logical Index</i> =	Key of the rPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rPRISTAT tuple being audited.
<i>Dump 2</i> =	If the address field has a value of GLFAIL (-1), then this is the dump of the rPORTLA tuple of the only D channel this rPRISTAT has, either primary or secondary.
<i>Dump 3</i> =	Otherwise, this is the dump of the rPORTLA tuple of the primary D-port. This dump will be there only in the DCBU case, that is the address field does not have a GLFAIL (-1) value. It is the dump of the rPORTLA tuple of the secondary D-port.

Table 17 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	The <i>dprim_port</i> field of the rPRISTAT tuple has been validated from the static rPRIGRP tuple. And yet audit could not find a valid rPORTLA tuple for this D-port.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and return. This will cause the PORTLA audit to run as a related audit to PRISTAT audit and should correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>port</i> field of the rPORTLA tuple that audit tried to access.
<i>Good Data</i> =	The member field of the <i>dprim_port</i> field of rPRISTAT tuple that was used as a key to access the rPORTLA tuple.
<i>Logical Index</i> =	Key of the rPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rPRISTAT tuple being audited.
<i>Dump 2</i> =	The rPORTLA tuple that was unsuccessfully accessed.

Table 18 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The <i>t321_job_id</i> field of the rPRISTAT tuple has a valid ISPA job_id and it has been verified. Meaning the t321_job has been queued. This job is queued when a t321 timer expires. And yet the <i>t321_idx</i> field is set meaning a t321 timer is running.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will clear the t321 timer and set the <i>t321_idx</i> field to a null value, TMTNULLIDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>list</i> field of the rPORTLA tuple of the primary D-port.
<i>Bad Data</i> =	The <i>t321_idx</i> field of the rPRISTAT tuple.

<i>Good Data</i> =	The <i>list</i> field of the rPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 19 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	Using the <i>dsec_port</i> of the rIPRISTAT tuple being audited as the key, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Refer to the Manual Action Required field for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Refer to the dump of the rIPRISTAT tuple from this error code. Find out the value of <i>dsec_port</i> field and try to insert an rIDCHNPRI tuple to static ODD. As for the other fields, consult other areas including terminal maintenance and triggers.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 20 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	There are six different fields in the rIPRISTAT tuple that represents an ISPA job_id when an ISPA job is queued to be scheduled. These fields are, <i>t321_job_id</i> , <i>mfe1_job_id</i> , <i>mfe2_job_id</i> , <i>mfr1_job_id</i> , <i>mfr2_job_id</i> , and <i>bchan_job_id</i> . When audit tried to audit one of these fields, the job_id field was found to have neither a valid value within allowable range which is 0 through less than TMNULLJOBID nor it is null (TMNULLJOBID). As to which field is in error, refer to the address field for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The job_id field that was in error will be reset to a null value, TMNULLJOBID.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = The <i>t321_job_id</i> field was in error. 2 = The <i>mfe1_job_id</i> field was in error. 3 = The <i>mfe2_job_id</i> field was in error. 4 = The <i>mfr1_job_id</i> field was in error. 5 = The <i>mfr2_job_id</i> field was in error. 6 = The <i>bchan_job_id</i> field was in error.
<i>Bad Data</i> =	The job_id field of the of rIPRISTAT tuple tuple that was in error Refer to the address field value and its explanation as to which field was in error.
<i>Good Data</i> =	The null value of the ISPA job_id, TMNULLJOBID.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 21 Error Code: PORT

Error Code:	PORT
Error Description:	Using the non-null D-port field of the rIPRISTAT tuple as the key, either the

	<i>dprim_port</i> or <i>dsec_port</i> , audit tried to read the static rIDSLEQUIP tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Refer to the Manual Action Required field for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Refer to the dump of the rIPRISTAT tuple from this error code. Find out the value of member field of <i>dprim_port</i> field. If this field was null, find out the member field of the <i>dsec_port</i> field. Using the member field as a key, try to insert an rIDSLEQUIP tuple to static ODD. As for the other fields, consult other areas including Triggers.
Dump Description:	
<i>Error Address</i> =	The D-port used to access the rIDSLEQUIP tuple.
<i>Bad Data</i> =	The <i>is_dte</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The <i>dsl</i> field(key) of the rIDSLEQUIP tuple. This should be same as the value in Address field in normal cases.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 22 Error Code: RELQKEY

Error Code:	RELQKEY
Error Description:	This error refers to an error related to one of the five fields of the rIPRISTAT tuple, namely, <i>mfe1_job_id</i> , <i>mfe2_job_id</i> , <i>mfr1_job_id</i> , <i>mfr2_job_id</i> and <i>bchan_job_id</i> . All these field represent an ISPA job that has been queued to be processed. When audit finds these fields having a valid job_id, it verifies that the job is indeed queued through a TM interface. Audit reports this error when it receives a return value of anything other than GLSUCCESS. To determine which field was in error, refer to the good data field for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The job_id field that was found in error will be updated with the null value, TMNULLJOBID. Refer to good data field for the job_id field found in error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The return value from the TM interface.
<i>Bad Data</i> =	The job_id field of rIPRISTAT tuple that was in error. Refer to the good data field to determine which field is this.
<i>Good Data</i> =	Valid values are: 1 = The <i>mfe1_job_id</i> field was in error. 2 = The <i>mfe2_job_id</i> field was in error. 3 = The <i>mfr1_job_id</i> field was in error. 4 = The <i>mfr2_job_id</i> field was in error. 5 = The <i>bchan_job_id</i> field was in error.
<i>Logical Index</i> =	This field has five different values depending upon the value found in the good data field above and it has following meanings: - If the good data field is 1, then this field equals the member field of the <i>dprim_port</i> field of rIPRISTAT tuple. - If the good data field is 2, then this field equals the member field of the <i>dsec_port</i> field of rIPRISTAT tuple. - If the good data field is 3, then this field equals the member field of the <i>dprim_port</i> field of rIPRISTAT tuple.

	- If the good data field is 4, then this field equals the member field of the <i>dsec_port</i> field of rIPRISTAT tuple. - If the good data field is 5, then this field equals the member field of the <i>dprim_port</i> field of rIPRISTAT tuple if it is not null, otherwise it is the member field of the <i>dsec_port</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rPORTLA tuple of the primary D-port if it is not null else it is the rPORTLA tuple of the secondary D-port.

Table 23 Error Code: RL_MISMATCH

Error Code:	RL_MISMATCH
Error Description:	The module field of the <i>sig_port</i> field of the rIPRISTAT tuple did not match the same field of either the primary D-port or the secondary D-port. As to which D-port it should have matched, refer to the address field.
Possible Error Effect:	The PRI calls may fail.
Corrective Action Taken:	The module field of the <i>sig_port</i> field will be updated with the module field of either of the D-port as applicable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = The member field of the <i>sig_port</i> field was equal to the member field of the primary D-port and the module field of <i>sig_port</i> does not match the module field of the primary D-port. 2 = The member field of the <i>sig_port</i> field was equal to the member field of the secondary D-port and the module field of <i>sig_port</i> does not match the module field of the secondary D-port.
<i>Bad Data</i> =	The module field of the <i>sig_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the D-port equaling signaling D-port. Refer to the Address field to decide.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 24 Error Code: R_PORT

Error Code:	R_PORT
Error Description:	The <i>dprim_port</i> field of the rIPRISTAT tuple did not match the same field in the static rIPRIGRP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprim_port</i> will be updated with the value from the same field from static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The member field of the <i>dprim_port</i> field of rIPRIGRP tuple.
<i>Bad Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dprim_port</i> field of rIPRIGRP tuple.
<i>Logical Index</i> =	The module field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 25 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Using the <i>dprim_port</i> of the rIPRISTAT tuple being audited as the key, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Refer to the Manual Action Required field for more information.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Refer to the dump of the rIPRISTAT tuple from this error code. Find out the value of <i>dprim_port</i> field and try to insert an rIDCHNPRI tuple to static ODD. As for the other fields, consult other areas including terminal maintenance and triggers.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 26 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The <i>t321_idx</i> field of the rIPRISTAT tuple indicates whether a T321 timer is running or not and one of the D-channel is in IS_WAIT state. The <i>tserve_idx</i> field also indicates whether a T321 timer is running or not, but the D-channel is in IS state. Both timers are applicable only in the D channel Back up case and should have a null value in all other cases. But audit found either one of the timer to have non-null value when this PRI did not have a D-channel back up. As to which timer has non-null value, refer to the address field.
Possible Error Effect:	The D-channels may get re-initialized.
Corrective Action Taken:	The <i>t321_idx</i> field or <i>tserve_idx</i> field will be reset to its null value, TMTNULLIDX, depending upon which ever was non-null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = The <i>t321_idx</i> field has a non-null value. 2 = The <i>tserve_idx</i> field has a non-null value.
<i>Bad Data</i> =	The <i>t321_idx</i> or <i>tserve_idx</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The null value of the timer index, TMTNULLIDX.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 27 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The <i>t321_idx</i> field of the rIPRISTAT tuple indicates whether a T321 timer is running or not and is applicable only in the D-channel back up case. When this field is set, the timer is running and one of the two D-ports must be in on the WAIT list. When audit made list check, it did not find any one of the two D-ports on the TMLTWAIT list when the <i>t321_idx</i> field was set.
Possible Error Effect:	The D-channels may get re-initialized.
Corrective Action Taken:	The T321 timer will be stopped and the <i>t321_idx</i> field will be reset to its null value, TMTNULLIDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The list field of the rIPORTLA tuple of the primary D-channel of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>t321_idx</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The list field of the rIPORTLA tuple of the secondary D-channel of the rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.

<i>Dump 3</i> =	The rPORTLA tuple of the secondary D-port.

Table 28 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The <i>sig_int_id</i> field of the rIPRISTAT tuple did not match either the <i>interface_id</i> (DFI) of primary D-port or <i>interface_id</i> of secondary D-port. As to which <i>interface_id</i> it should have matched, refer to the address field.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>sig_int_id</i> field will be updated with the <i>interface_id</i> of either of the D-port as applicable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Valid values are: 1 = The <i>sig_port</i> field was equal to the primary D-port and the <i>sig_int_id</i> does not match the <i>interface_id</i> of primary D-port. 2 = The <i>sig_port</i> field was equal to the secondary D-port and the <i>sig_int_id</i> does not match the <i>interface_id</i> of secondary D-port.
<i>Bad Data</i> =	The <i>sig_int_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>interface_id</i> of the D-port equaling signaling D-port. Refer to the Address field to decide.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

PRISTAT-B

Software Release: 5E12 only

1. PRIMARY RATE INTERFACE GROUP STATUS (PRISTAT) AUDIT

The primary rate interface group status (PRISTAT) audit audits the RLPRISTAT relation, introduced by the non-facility associated signaling (NFAS) and D-channel backup (DCBU) features available on the 5ESS® switch. This relation is used by all non operator services position system (OSPS) primary rate interface (PRI).

An NFAS provides a Bellcore compatible PRI defined as $nB + D$, where $n=1-479$. PRI is a collection of B-channels and a signaling D-channel. All B-channels are controlled by that D-channel. Now there are two types of non-OSPS PRI. In the classic definition, n is a maximum of 23 and all B-channels are to be on the same digital signal 1 (DS1) facility known as digital facilities interface (DFI). This is called a custom PRI. In a standard PRI definition, n is a maximum of 479 and the B-channels can spread over multiple facilities with maximum up to 20 facilities within a single SM. The D-channels must reside on the 24th channel of a DFI and must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

DCBU is a Bellcore compatible PRI with $mB + 2D$, where $m = 1 - 478$, and there are 2 D-channels. One D-channel is designated as "active" and the other D-channel is designated as "standby". Should the "active" D-channel fail due to hardware, layer 2 or layer 3, then the "standby" D-channel should become the "active" D channel. Both D-channels must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

Both NFAS and DCBU apply only to PRIs that can only interface to CPEs (includes the 5ESS® switch "User side PRIs") which are compatible with Bellcore's TR-1268. The existing $23B + D$ PRIs (also known as custom PRI) will not support NFAS and DCBU.

The PRISTAT audit runs in the OSDS-M environment in the segmented mode.

2. DATA STRUCTURES

For details on the fields contained in the rIPRISTAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIS_DTE

Error Code:	BADIS_DTE
Error Description:	The <i>is_dte</i> field of the rIPRISTAT tuple did not match the same field of the static rIDSLEQUIP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>is_dte</i> field will be updated with the value found in the same field found in the rIDSLEQUIP tuple.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The key field of the rIDSLEQUIP tuple, <i>dsi</i> , which is also one of the D-ports of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>is_dte</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>is_dte</i> field of rIDSLEQUIP tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 2 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The <i>std_nsf_ie</i> field in the rIPRISTAT tuple does not match the same field in the static rIPRIGRP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The field in the rIPRISTAT tuple is updated with the value in the static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0 (N/A).
<i>Bad Data</i> =	The <i>std_nsf_ie</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>std_nsf_ie</i> field of rIPRIGRP tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPRIGRP tuple.

Table 3 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The <i>t321_job_id</i> field of the rIPRISTAT tuple has a valid ISPA job_id value indicating that there is a valid ISPA job on the queue ready to be run. But when audit checked this fact through a TM interface, using first the member field of the primary D-port and then the member field of the secondary D-port, it received failure both the times from the TM interface (other than GLSUCCESS).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>t321_job_id</i> field will be updated with the null value, TMNULLJOBID.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>port</i> field of the rIPORTLA tuple of the primary D-port.
<i>Bad Data</i> =	The <i>t321_job_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>port</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 4 Error Code: BDPID

Error Code:	BDPID
Error Description:	An rIPRISTAT tuple contains an invalid restart process number (rIPRISTAT.restart_pid).
Possible Error Effect:	Restart procedures may not be applied correctly to this PRI.
Corrective Action Taken:	The audit clears the process field (rIPRISTAT.restart_pid = OSnullpid).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIPRISTAT.restart_pid.procno.
<i>Bad Data</i> =	rIPRISTAT.restart_pid.pcrd.
<i>Good Data</i> =	rIPRISTAT.pcrd.uniq.
<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump</i> =	The rIPRISTAT tuple.

Table 5 Error Code: CADNPORT

Error Code:	CADNPORT
Error Description:	An rIPRISTAT tuple contains an invalid CADN status for either the primary or secondary D-channel port (rIPRISTAT.dchstat). When the error address is 1 or

	2, it is checking the primary port 3 or 4, the secondary port. When the error address is 1 or 3, the primary or secondary port should be OOS CADN but is not when it is 2 or 4, the primary or secondary port should not be CADN.
Possible Error Effect:	CADN may be reported to the customer when it's not really there, or a port may be reported as in-service that is really CADN.
Corrective Action Taken:	The audit sets the CADN and status attributes appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1-4.
<i>Bad Data =</i>	The CADN bit in rIPRISTAT.dchstat.
<i>Good Data =</i>	When error address is 1 or 3, the status bit in rIPRISTAT.dchstat. When error address is 2, the list from the primary rIORTLA tuple. When error address is 4, the list from the secondary rIORTLA tuple.
<i>Logical Index =</i>	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1 =</i>	The rIPRISTAT tuple.
<i>Dump 2 =</i>	When error address is 1 or 2, the rIORTLA tuple of the primary D-channel port. When error address is 3 or 4, the rIORTLA tuple of the secondary D-channel port.

Table 6 Error Code: CALLFLAG_BAD

Error Code:	CALLFLAG_BAD
Error Description:	Audit found the error condition where the signaling D-port field, <i>sig_port</i> , is null, neither of the D-ports, primary or secondary, are owned by any process and yet the <i>block_call</i> bit is not set.
Possible Error Effect:	Attempt to start a PRI call may be made on this PRI group.
Corrective Action Taken:	The audit will set the <i>block_call</i> bit to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>block_call</i> field of the rIPRISTAT tuple.
<i>Good Data =</i>	The member field of the <i>sig_port</i> field of rIPRISTAT tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 7 Error Code: CDPRIM

Error Code:	CDPRIM
Error Description:	Audit found an error condition where the <i>dprimresp</i> field of the rIPRISTAT tuple is set to DBYES, the primary D-port is not null and it is not owned. The primary D-port must be owned, that is the <i>state</i> field of the rIORTLA tuple of the primary D-port must not be DBPSIDLE, when the <i>dprimresp</i> field is set.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprimresp</i> field will be updated to DBNO.
Manual Action Required:	None.
Dump Description:	Dump Description:
<i>Error Address =</i>	The member field of the primary D-port of rIPRISTAT tuple.
<i>Bad Data =</i>	The <i>dprimresp</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>state</i> field of the rIORTLA tuple of the primary D-port.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1 =</i>	The rIPRISTAT tuple being audited.
<i>Dump 2 =</i>	The rIORTLA tuple of the primary D-port.

Table 8 Error Code: CDSEC

Error Code:	CDSEC
Error Description:	Audit found an error condition where the <i>dsecresp</i> field of the rIPRISTAT tuple is set to DBYES, the secondary D-port is not null and it is not owned. The secondary D-port must be owned, that is the <i>state</i> field of the rIORTLA tuple of the secondary D-port must not be DBPSIDLE, when the <i>dsecresp</i> field is set.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dsecresp</i> field will be updated to DBNO.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The <i>member</i> field of the secondary D-port of rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>dsecresp</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>state</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the secondary D-port.

Table 9 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	There are two field in the rIPRISTAT tuple that represent process IDs (PIDs), namely, <i>send_pid</i> and <i>hdlb_pid</i> . During the auditing of these fields, the audit could not verify these process IDs, that is, the audit could not find these processes. For further information, refer to the <i>Good Data</i> field for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>process_id</i> field in error will be updated with the null value, OSnullpid. As to which field is nulled, refer to the good data field below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>procno</i> field of the process ID field to be in error.
<i>Bad Data</i> =	The <i>pcrid</i> field of the process ID field to be in error.
<i>Good Data</i> =	This field can have two different values and it has following meaning: 1 = The audit found error with the <i>send_pid</i> field of the rIPRISTAT tuple. 2 = The audit found error with the <i>hdlb_pid</i> field of the rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 10 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rIPRISTAT dynamic tuple without a corresponding static tuple where the rIPRISTAT does not have the static rIPRIGRP tuple with rIPRISTAT.prigrp = rIPRIGRP.prigrp.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	The rIPRISTAT tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPRISTAT tuple found to be extra.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 11 Error Code: INV

Error Code:	INV
Error Description:	One of the following fields in the rIPRISTAT tuple does not match the same field in the static rIPRIGRP tuple. • dprim_port.member • dprim_port.module • dsec_port.member • dsec_port.module

	• <code>pri_type</code> • <code>bchan_serv</code>
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The fields in the rIPRISTAT tuple are updated with the values in the static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0 (N/A).
<i>Bad Data =</i>	The <i>pri_type</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>pri_type</i> field of rIPRIGRP tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1 =</i>	The rIPRISTAT tuple being audited.
<i>Dump 2 =</i>	The rIPRIGRP tuple.

Table 12 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The <i>dprim_int_id</i> field of the rIPRISTAT tuple did not match the <i>interf_id</i> field of the static rIDCHNPRI tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprim_int_id</i> field will be updated with the value found in the <i>interf_id</i> field of the rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>dprim_int_id</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>interf_id</i> field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 13 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The <i>dsec_int_id</i> field of the rIPRISTAT tuple did not match the <i>interf_id</i> field of the static rIDCHNPRI tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dsec_int_id</i> field will be updated with the value found in the <i>interf_id</i> field of the rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>dsec_int_id</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>interf_id</i> field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 14 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	The <i>dsec_port</i> field of the rIPRISTAT tuple has been validated from the static rIPRIGRP tuple. And yet audit could not find a valid rIPORTLA tuple for this D-port.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and return. This will cause the PORTLA audit to run as a related audit to PRISTAT audit and should correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>port</i> field of the rIPORTLA tuple that audit tried to access.
<i>Good Data =</i>	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple that was used as a key to access the rIPORTLA tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.

<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple that was unsuccessfully accessed.

Table 15 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	Audit found an error condition where the <i>block_call</i> bit was set but the D-port that equaled the signaling D-port was not owned by a process. Audit will report this error in either of the two cases where the <i>sig_port</i> is equal to the primary or the secondary D-port. For further information, refer to the <i>Good Data</i> field.
Possible Error Effect:	The call originations on this PRI group will be stopped till the <i>block_call</i> bit is cleared.
Corrective Action Taken:	Since audit does not know whether the <i>block_call</i> bit was set correctly or not, it will report this error, null out the member field of the <i>sig_port</i> , set the <i>block_call</i> bit to DBYES and will start recovery of both the D-ports. Recovery code will make correct decisions.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>block_call</i> bit of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>state</i> field of rIPORTLA tuple of the D-port that equaled the signaling D-port, either primary or secondary. Refer to the <i>Good Data</i> field to determine which D-port.
<i>Good Data</i> =	This field will have one of the two values, 1 or 2 and has following meaning: 1 = The <i>sig_port</i> field was equal to the primary D-port and was not owned by any process. 2 = The <i>sig_port</i> field was equal to the secondary D-port and was not owned by any process.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the D-port, either primary or secondary. Refer to <i>Good Data</i> for further information.

Table 16 Error Code: MKNULL

Error Code:	MKNULL
Error Description:	Audit found an error condition where both the D-ports of the rIPRISTAT tuple were null, that is TMNULL_PORT, and either the signaling D-port, <i>sig_port</i> , was not null or the <i>block_call</i> bit was not set.
Possible Error Effect:	Calls may originate on this PRI group without any D-port in service.
Corrective Action Taken:	The audit will report the error, null out the <i>sig_port</i> field, set the <i>block_call</i> bit to DBYES and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>block_call</i> field of the rIPRISTAT tuple.
<i>Bad Data</i> =	The member field of the <i>sig_port</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 17 Error Code: MODE

Error Code:	MODE
Error Description:	An rIPRISTAT tuple contains an invalid mode (either AUTO or MANUAL) for either the primary or secondary D-channel port in rIPRISTAT.dchstat. When the error address is 1 or 2, it is checking the primary port 3 or 4, the secondary port. When the error address is 1 or 3, the primary or secondary port should be MANUAL but is instead AUTO when it is 2 or 4, the primary or secondary port should be AUTO.

Possible Error Effect:	The mode of this port on the PRI group will be reported incorrectly.
Corrective Action Taken:	The audit sets the mode appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1-4.
<i>Bad Data =</i>	The mode from rIPRISTAT.dchstat.
<i>Good Data =</i>	The mode from the rIORTLA.
<i>Logical Index =</i>	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1 =</i>	The rIPRISTAT tuple being audited.
<i>Dump 2 =</i>	When error address is 1 or 2, the rIORTLA tuple of the primary D-channel port. When error address is 3 or 4, the rIORTLA tuple of the secondary D-channel port.

Table 18 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	An rIPRISTAT tuple is reporting either the primary or secondary D-channel port as not provisioned when it is. When the error address is 1, it is checking the primary port 2, the secondary port.
Possible Error Effect:	The provisioning of this port on the PRI group will be reported incorrectly to the customer.
Corrective Action Taken:	The audit sets the provisioning appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1-2.
<i>Bad Data =</i>	The provisioning from rIPRISTAT.dchstat.
<i>Good Data =</i>	When error address is 1, the primary port (rIPRISTAT.dprim_port) when error address is 2, the secondary port (rIPRISTAT.dsec_port).
<i>Logical Index =</i>	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 19 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	This error represents various error conditions related to the signaling port, <i>sig_port</i> , field of the rIPRISTAT tuple. To determine what error condition, caused this error, refer to the <i>Good Data</i> field and compare it with the explanation given in this field for each value.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The member field of the <i>sig_port</i> field will be set to null value, the <i>block_call</i> bit will be set to DBYES, and both the D-ports will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field will have one of the two values, and it has following meaning: - GLFAIL or -1 means there was only one D-port available to this PRI group or in other words, this is an NFAS case. - If it is not GLFAIL(-1), then it is the <i>list</i> field of the primary D-port's rIORTLA tuple.
<i>Bad Data =</i>	This field will have one of two values, and it has following meaning: - If the address field above has a value of GLFAIL(-1), then this is <i>list</i> of the rIORTLA tuple of the only D-port that this rIPRISTAT tuple has. - If the address field above does not have a value of GLFAIL(-1), then this is the <i>list</i> field of the rIORTLA tuple of the secondary D-port.
<i>Good Data =</i>	This field will has four different values, and it has following meaning: 1 = The audit found an error where the <i>sig_port</i> field has a null value

	(TMNULL_PORT), neither primary nor the secondary D-port is owned by a process and either primary or secondary D-port is on the in service list (the <i>list</i> field of the rIPORTLA is TMLTIS). 2 = The audit found an error where the <i>sig_port</i> field is equal to the primary D-port but the primary D-port is not on the in service list, that is the <i>list</i> field of the rIPORTLA of the primary D-port is not TMLTIS. 3 = The audit found an error where the <i>sig_port</i> field is equal to the secondary D-port but the secondary D-port is not on the in service list, that is the <i>list</i> field of the rIPORTLA of the secondary D-port is not TMLTIS. 4 = The audit found an error where the <i>sig_port</i> field has non null value and yet it is neither equal to the primary nor the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	If the address field has a value of GLFAIL(-1), then this is the dump of the rIPORTLA tuple of the only D-channel this rIPRISTAT has, either primary or secondary. Else, this is the dump of the rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	This dump will be there only in the DCBU case, that is the address field does not have a GLFAIL(-1) value. It is the dump of the rIPORTLA tuple of the secondary D-port.

Table 20 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	The <i>dprim_port</i> field of the rIPRISTAT tuple has been validated from the static rIPRIGRP tuple. And yet audit could not find a valid rIPORTLA tuple for this D-port.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and return. This will cause the PORTLA audit to run as a related audit to PRISTAT audit and should correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>port</i> field of the rIPORTLA tuple that audit tried to access.
<i>Good Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple that was used as a key to access the rIPORTLA tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple that was unsuccessfully accessed.

Table 21 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	An rIPRISTAT tuple is reporting either the primary or secondary D-channel port as provisioned when it is not. When the error address is 1, it is checking the primary port 2, the secondary port.
Possible Error Effect:	The provisioning of this port on the PRI group will be reported incorrectly to the customer.
Corrective Action Taken:	The audit sets the provisioning appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1-2.
<i>Bad Data</i> =	The provisioning from rIPRISTAT.dchstat.
<i>Good Data</i> =	When error address is 1, the primary port (rIPRISTAT.dprim_port) when error address is 2, the secondary port (rIPRISTAT.dsec_port).
<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 22 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The <i>t321_job_id</i> field of the rIPRISTAT tuple has a valid ISPA job_id and it has been verified. Meaning the t321_job has been queued. This job is queued when a t321 timer expires. And yet the <i>t321_idx</i> field is set meaning a t321 timer is running.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will clear the t321 timer and set the <i>t321_idx</i> field to a null value, TMNULLIDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>list</i> field of the rPORTLA tuple of the primary D-port.
<i>Bad Data</i> =	The <i>t321_idx</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The <i>list</i> field of the rPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 23 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Using the <i>dsec_port</i> of the rIPRISTAT tuple being audited as the key, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Look at the Manual Action Required field below for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Look at the dump of the rIPRISTAT tuple from this error code. Find out the value of <i>dsec_port</i> field and try to insert an rIDCHNPRI tuple to static ODD. As for the other fields, consult other areas including terminal maintenance and triggers.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 24 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	There are six different fields in the rIPRISTAT tuple that represents an ISPA job_id when an ISPA job is queued to be scheduled. These fields are, <i>t321_job_id</i> , <i>mfe1_job_id</i> , <i>mfe2_job_id</i> , <i>mfr1_job_id</i> , <i>mfr2_job_id</i> , and <i>bchan_job_id</i> . When audit tried to audit one of these fields, the job_id field was found to have neither a valid value within allowable range which is 0 through less than TMNULLJOBID nor it is null (TMNULLJOBID). As to which field is in error, refer to the address field below for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The job_id field that was in error will be reset to a null value, TMNULLJOBID.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field can have one of the six different values and it has following meanings: 1 = The <i>t321_job_id</i> field was in error. 2 = The <i>mfe1_job_id</i> field was in error. 3 = The <i>mfe2_job_id</i> field was in error. 4 = The <i>mfr1_job_id</i> field was in error. 5 = The <i>mfr2_job_id</i> field was in error.

	6 = The <i>bchan_job_id</i> field was in error.
<i>Bad Data</i> =	The <i>job_id</i> field of the of rIPRISTAT tuple tuple that was in error Refer to the address field value and its explanation as to which field was in error.
<i>Good Data</i> =	The null value of the ISPA <i>job_id</i> , TMNULLJOBID.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 25 Error Code: PORT

Error Code:	PORT
Error Description:	Using the non-null D-port field of the rIPRISTAT tuple as the key, either the <i>dprim_port</i> or <i>dsec_port</i> , audit tried to read the static rIDSLEQUIP tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Look at the Manual Action Required field below for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Look at the dump of the rIPRISTAT tuple from this error code. Find out the value of member field of <i>dprim_port</i> field. If this field was null, find out the member field of the <i>dsec_port</i> field. Using the member field as a key, try to insert an rIDSLEQUIP tuple to static ODD. As for the other fields, consult other areas including triggers.
Dump Description:	
<i>Error Address</i> =	The D-port used to access the rIDSLEQUIP tuple.
<i>Bad Data</i> =	The <i>is_dte</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The <i>dsl</i> field(key) of the rIDSLEQUIP tuple. This should be same as the value in <i>Error Address</i> in normal cases.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 26 Error Code: RELQKEY

Error Code:	RELQKEY
Error Description:	This error refers to an error related to one of the five fields of the rIPRISTAT tuple, namely, <i>mfe1_job_id</i> , <i>mfe2_job_id</i> , <i>mfr1_job_id</i> , <i>mfr2_job_id</i> and <i>bchan_job_id</i> . All these field represent an ISPA job that has been queued to be processed. When audit finds these fields having a valid <i>job_id</i> , it verifies that the job is indeed queued through a TM interface. Audit reports this error when it receives a return value of anything other than GLSUCCESS. As to which field was in error, refer to the good data field below for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>job_id</i> field that was found in error will be updated with the null value, TMNULLJOBID. Refer to good data field for the <i>job_id</i> field found in error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The return value from the TM interface.
<i>Bad Data</i> =	The <i>job_id</i> field of rIPRISTAT tuple that was in error. Refer to the good data field below to determine which field is this.
<i>Good Data</i> =	This field represents five different values of the <i>job_ids</i> and it has following meanings: 1 = The <i>mfe1_job_id</i> field was in error. 2 = The <i>mfe2_job_id</i> field was in error. 3 = The <i>mfr1_job_id</i> field was in error. 4 = The <i>mfr2_job_id</i> field was in error.

	5 = The <i>bchan_job_id</i> field was in error.
<i>Logical Index</i> =	This field has five different values depending upon the value found in the good data field above and it has following meanings: • If the value in the good data field is 1, then this field has a value of the member field of the <i>dprim_port</i> field of rIPRISTAT tuple. • If the value in the good data field is 2, then this field has a value of the member field of the <i>dsec_port</i> field of rIPRISTAT tuple. • If the value in the good data field is 3, then this field has a value of the member field of the <i>dprim_port</i> field of rIPRISTAT tuple. • If the value in the good data field is 4, then this field has a value of the member field of the <i>dsec_port</i> field of rIPRISTAT tuple. • If the value in the good data field is 5, then this field has a value of the member field of the <i>dprim_port</i> field of rIPRISTAT tuple if it is not null else it is the member field of the <i>dsec_port</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port if it is not null else it is the rIPORTLA tuple of the secondary D-port.

Table 27 Error Code: RL_MISMATCH

Error Code:	RL_MISMATCH
Error Description:	The module field of the <i>sig_port</i> field of the rIPRISTAT tuple did not match the same field of either the primary D-port or the secondary D-port. As to which D-port it should have matched, look at the address field below.
Possible Error Effect:	The PRI calls may fail.
Corrective Action Taken:	The module field of the <i>sig_port</i> field will be updated with the module field of either of the D-port as applicable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field will have one of the two following values: 1 = The member field of the <i>sig_port</i> field was equal to the member field of the primary D-port and the module field of <i>sig_port</i> does not match the module field of the primary D-port. 2 = The member field of the <i>sig_port</i> field was equal to the member field of the secondary D-port and the module field of <i>sig_port</i> does not match the module field of the secondary D-port.
<i>Bad Data</i> =	The module field of the <i>sig_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the D-port equaling signaling D-port. Refer to the <i>Error Address</i> field.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 28 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Using the <i>dprim_port</i> of the rIPRISTAT tuple being audited as the key, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Look at the Manual Action Required field below for more information.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Look at the dump of the rIPRISTAT tuple from this error code. Find out the value of <i>dprim_port</i> field and try to insert an rIDCHNPRI tuple to static ODD. As for the other fields, consult other areas including terminal maintenance and triggers.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 29 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	An rIPRISTAT tuple contains a bad status - either in-service (IS) or out-of-service (OOS) for either the primary or secondary D-channel port in rIPRISTAT.dchstat. When the error address is 1 or 2, it is checking the primary port 3 or 4, the secondary port. When the error address is 1 or 3, the primary or secondary port should be OOS but is instead IS when it is 2 or 4, the primary or secondary port should be IS.
Possible Error Effect:	The status of this port on the PRI group will be reported incorrectly to the customers.
Corrective Action Taken:	The audit sets the status appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1-4.
<i>Bad Data</i> =	The status from rIPRISTAT.dchstat.
<i>Good Data</i> =	The status from the rIPORTLA (rIPORTLA.bsc_state).
<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	When error address is 1 or 2, the rIPORTLA tuple of the primary D-channel port. When error address is 3 or 4, the rIPORTLA tuple of the secondary D-channel port.

Table 30 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The <i>t321_idx</i> field of the rIPRISTAT tuple indicates whether a T321 timer is running or not and one of the D-channel is in IS_WAIT state. The <i>tservice_idx</i> field also indicates whether a T321 timer is running or not, but the D-channel is in IS state. Both timers are applicable only in the D-channel Back up case and should have a null value in all other cases. But audit found either one of the timer to have non-null value when this PRI did not have a D-channel back up. As to which timer has non-null value, look at the address field below.
Possible Error Effect:	The D-channels may get re-initialized.
Corrective Action Taken:	The <i>t321_idx</i> field or <i>tservice_idx</i> field will be reset to its null value, TMTNULLIDX, depending upon which ever was non-null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field will have one of the following values: 1 = The <i>t321_idx</i> field has a non-null value. 2 = The <i>tservice_idx</i> field has a non-null value.
<i>Bad Data</i> =	The <i>t321_idx</i> or <i>tservice_idx</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The null value of the timer index, TMTNULLIDX.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 31 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The <i>t321_idx</i> field of the rIPRISTAT tuple indicates whether a T321 timer is running or not and is applicable only in the D-channel back up case. When this field is set, the timer is running and one of the two D-ports must be in on the WAIT list. When audit made list check, it did not find any one of the two D-ports on the TMLTWAIT list when the <i>t321_idx</i> field was set.
Possible Error Effect:	The D-channels may get re-initialized.
Corrective Action Taken:	The T321 timer will be stopped and the <i>t321_idx</i> field will be reset to its null value, TMTNULLIDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The list field of the rPORTLA tuple of the primary D-channel of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>t321_idx</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The list field of the rPORTLA tuple of the secondary D-channel of the rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rPORTLA tuple of the secondary D-port.

Table 32 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The <i>sig_int_id</i> field of the rIPRISTAT tuple did not match either the interface_id (DFI) of primary D-port or interface_id of secondary D-port. For further information, refer to the <i>Error Address</i> field.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>sig_int_id</i> field will be updated with the interface_id of either of the D-port as applicable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field will have one of the following values: 1 = The <i>sig_port</i> field was equal to the primary D-port and the <i>sig_int_id</i> does not match the interface_id of primary D-port. 2 = The <i>sig_port</i> field was equal to the secondary D-port and the <i>sig_int_id</i> does not match the interface_id of secondary D-port.
<i>Bad Data</i> =	The <i>sig_int_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The interface_id of the D-port equaling signaling D-port. For further information, refer to the <i>Error Address</i> field.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

PRISTAT-C

Software Release: 5E13 and later

1. PRIMARY RATE INTERFACE GROUP STATUS (PRISTAT) AUDIT

The primary rate interface group status (PRISTAT) audit audits the RLPRISTAT relation, introduced by the non-facility associated signaling (NFAS) and D-channel backup (DCBU) features available on the 5ESS® switch. This relation is used by all non operator services position system (OSPS) primary rate interface (PRI).

An NFAS provides a Bellcore compatible PRI defined as $nB + D$, where $n=1-479$. PRI is a collection of B-channels and a signaling D-channel. All B-channels are controlled by that D-channel. Now there are two types of non-OSPS PRI. In the classic definition, n is a maximum of 23 and all B-channels are to be on the same digital signal 1 (DS1) facility known as digital facilities interface (DFI). This is called a custom PRI. In a standard PRI definition, n is a maximum of 479 and the B-channels can spread over multiple facilities with maximum up to 20 facilities within a single SM. The D-channels must reside on the 24th channel of a DFI and must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

DCBU is a Bellcore compatible PRI with $mB + 2D$, where $m = 1 - 478$, and there are 2 D-channels. One D-channel is designated as "active" and the other D-channel is designated as "standby". Should the "active" D-channel fail due to hardware, layer 2 or layer 3, then the "standby" D-channel should become the "active" D channel. Both D-channels must be on the same SM as the B-channels that it is providing signaling for (that is, all of the B-channels and its signaling D-channels are in the same PRI group).

Both NFAS and DCBU apply only to PRIs that can only interface to CPEs (includes the 5ESS® switch "User side PRIs") which are compatible with Bellcore's TR-1268. The existing $23B + D$ PRIs (also known as custom PRI) will not support NFAS and DCBU.

The PRISTAT audit runs in the OSDS-M environment in the segmented mode.

2. DATA STRUCTURES

For details on the fields contained in the rIPRISTAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIS_DTE

Error Code:	BADIS_DTE
Error Description:	The <i>is_dte</i> field of the rIPRISTAT tuple did not match the same field of the static rIDSLEQUIP tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>is_dte</i> field will be updated with the value found in the same field found in the rIDSLEQUIP tuple.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The key field of the rIDSLEQUIP tuple, <i>dsi</i> , which is also one of the D-ports of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>is_dte</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>is_dte</i> field of rIDSLEQUIP tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	The <i>t321_job_id</i> field of the rIPRISTAT tuple has a valid ISPA job_id value indicating that there is a valid ISPA job on the queue ready to be run. But when audit checked this fact through a TM interface, using first the member field of the primary D-port and then the member field of the secondary D-port, it received failure both the times from the TM interface (other than GLSUCCESS).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>t321_job_id</i> field will be updated with the null value, TMNULLJOBID.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>port</i> field of the rIPORTLA tuple of the primary D-port.
<i>Bad Data</i> =	The <i>t321_job_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>port</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 3 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The number of active transfers per PRI group is not correct. The audit just looked into each rPCBLA tuple and counted that number up, recording it in rIPRISTAT.tbct_aud_cnt, and what is there is not equal to what is in rIPRISTAT.num_act_tbct.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>rIPRISTAT.num_act_tbct</i> field will be updated with the value from <i>rIPRISTAT.tbct_aud_cnt</i> .
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>num_act_tbct</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>tbct_aud_cnt</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 4 Error Code: BDPID

Error Code:	BDPID
Error Description:	An rIPRISTAT tuple contains an invalid restart process number (rIPRISTAT.restart_pid).
Possible Error Effect:	Restart procedures may not be applied correctly to this PRI.
Corrective Action Taken:	The audit clears the process field (rIPRISTAT.restart_pid = OSnullpid).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIPRISTAT.restart_pid.procn.
<i>Bad Data</i> =	rIPRISTAT.restart_pid.pcrd.
<i>Good Data</i> =	rIPRISTAT.pcrd.uniq.
<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump</i> =	The rIPRISTAT tuple.

Table 5 Error Code: CADNPORT

Error Code:	CADNPORT

Error Description:	An rIPRISTAT tuple contains an invalid CADN status for either the primary or secondary D-channel port (rIPRISTAT.dchstat). When the error address is 1 or 2, it is checking the primary port 3 or 4, the secondary port. When the error address is 1 or 3, the primary or secondary port should be OOS CADN but is not when it is 2 or 4, the primary or secondary port should not be CADN.
Possible Error Effect:	CADN may be reported to the customer when it's not really there, or a port may be reported as in-service that is really CADN.
Corrective Action Taken:	The audit sets the CADN and status attributes appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1-4.
<i>Bad Data =</i>	The CADN bit in rIPRISTAT.dchstat.
<i>Good Data =</i>	When error address is 1 or 3, the status bit in rIPRISTAT.dchstat. When error address is 2, the list from the primary rIPORTLA tuple. When error address is 4, the list from the secondary rIPORTLA tuple.
<i>Logical Index =</i>	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1 =</i>	The rIPRISTAT tuple.
<i>Dump 2 =</i>	When error address is 1 or 2, the rIPORTLA tuple of the primary D-channel port. When error address is 3 or 4, the rIPORTLA tuple of the secondary D-channel port.

Table 6 Error Code: CALLFLAG_BAD

Error Code:	CALLFLAG_BAD
Error Description:	Audit found the error condition where the signaling D-port field, <i>sig_port</i> , is null, neither of the D-ports, primary or secondary, are owned by any process and yet the <i>block_call</i> bit is not set.
Possible Error Effect:	Attempt to start a PRI call may be made on this PRI group.
Corrective Action Taken:	The audit will set the <i>block_call</i> bit to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>block_call</i> field of the rIPRISTAT tuple.
<i>Good Data =</i>	The member field of the <i>sig_port</i> field of rIPRISTAT tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 7 Error Code: CDPRIM

Error Code:	CDPRIM
Error Description:	Audit found an error condition where the <i>dprimresp</i> field of the rIPRISTAT tuple is set to DBYES, the primary D-port is not null and it is not owned. The primary D-port must be owned, that is the <i>state</i> field of the rIPORTLA tuple of the primary D-port must not be DBPSIDLE, when the <i>dprimresp</i> field is set.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprimresp</i> field will be updated to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The <i>member</i> field of the primary D-port of rIPRISTAT tuple.
<i>Bad Data =</i>	The <i>dprimresp</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>state</i> field of the rIPORTLA tuple of the primary D-port.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1 =</i>	The rIPRISTAT tuple being audited.
<i>Dump 2 =</i>	The rIPORTLA tuple of the primary D-port.

Table 8 Error Code: CDSEC

Error Code:	CDSEC
Error Description:	Audit found an error condition where the <i>dsecresp</i> field of the rIPRISTAT tuple is set to DBYES, the secondary D-port is not null and it is not owned. The secondary D-port must be owned, that is the <i>state</i> field of the rIPORTLA tuple of the secondary D-port must not be DBPSIDLE, when the <i>dsecresp</i> field is set.
Possible Error Effect:	Varies according to conditions.

Corrective Action Taken:	The <i>dsecresp</i> field will be updated to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>member</i> field of the secondary D-port of rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>dsecresp</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>state</i> field of the rIPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the secondary D-port.

Table 9 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	There are two field in the rIPRISTAT tuple that represent process IDs (PIDs), namely, <i>send_pid</i> and <i>hdlb_pid</i> . During the auditing of these fields, the audit could not verify these process IDs, that is, the audit could not find these processes. For further information, refer to the <i>Good Data</i> field for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>process_id</i> field in error will be updated with the null value, OSnullpid. As to which field is nulled, refer to the <i>good data</i> field below.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>procno</i> field of the process ID field to be in error.
<i>Bad Data</i> =	The <i>pcrid</i> field of the process ID field to be in error.
<i>Good Data</i> =	This field can have two different values and it has following meaning: 1 = The audit found error with the <i>send_pid</i> field of the rIPRISTAT tuple. 2 = The audit found error with the <i>hdlb_pid</i> field of the rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 10 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rIPRISTAT dynamic tuple without a corresponding static tuple where the rIPRISTAT does not have the static rIPRIGRP tuple with rIPRISTAT.prigrp = rIPRIGRP.prigrp.
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	The rIPRISTAT tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPRISTAT tuple found to be extra.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 11 Error Code: INV

Error Code:	INV
Error Description:	One of the following fields in the rIPRISTAT tuple does not match the same field in the static rIPRIGRP tuple. • dprim_port.member • dprim_port.module • dsec_port.member

	- dsec_port.module - pri_type - std_nsf_ie - bchan_serv
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The fields in the rIPRISTAT tuple are updated with the values in the static rIPRIGRP tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0 (N/A).
<i>Bad Data =</i>	The <i>pri_type</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>pri_type</i> field of rIPRIGRP tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1=</i>	The rIPRISTAT tuple being audited.
<i>Dump 2=</i>	The rIPRIGRP tuple.

Table 12 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The <i>dprim_int_id</i> field of the rIPRISTAT tuple did not match the <i>interf_id</i> field of the static rIDCHNPRI tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dprim_int_id</i> field will be updated with the value found in the <i>interf_id</i> field of the rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>dprim_int_id</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>interf_id</i> field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 13 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The <i>dsec_int_id</i> field of the rIPRISTAT tuple did not match the <i>interf_id</i> field of the static rIDCHNPRI tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>dsec_int_id</i> field will be updated with the value found in the <i>interf_id</i> field of the rIDCHNPRI tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.
<i>Bad Data =</i>	The <i>dsec_int_id</i> field of rIPRISTAT tuple.
<i>Good Data =</i>	The <i>interf_id</i> field of the rIDCHNPRI tuple.
<i>Logical Index =</i>	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 14 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	The <i>dsec_port</i> field of the rIPRISTAT tuple has been validated from the static rIPRIGRP tuple. And yet audit could not find a valid rIPORTLA tuple for this D-port.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and return. This will cause the PORTLA audit to run as a related audit to PRISTAT audit and should correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPRISTAT tuple being audited.

<i>Bad Data</i> =	The <i>port</i> field of the rIORTLA tuple that audit tried to access.
<i>Good Data</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple that was used as a key to access the rIORTLA tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIORTLA tuple that was unsuccessfully accessed.

Table 15 Error Code: MKBUSY

Error Code:	MKBUSY
Error Description:	Audit found an error condition where the <i>block_call</i> bit was set but the D-port that equaled the signaling D-port was not owned by a process. Audit will report this error in either of the two cases where the <i>sig_port</i> is equal to the primary or the secondary D-port. For further information, refer to the <i>Good Data</i> field.
Possible Error Effect:	The call originations on this PRI group will be stopped till the <i>block_call</i> bit is cleared.
Corrective Action Taken:	Since audit does not know whether the <i>block_call</i> bit was set correctly or not, it will report this error, null out the member field of the <i>sig_port</i> , set the <i>block_call</i> bit to DBYES and will start recovery of both the D-ports. Recovery code will make correct decisions.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>block_call</i> bit of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>state</i> field of rIORTLA tuple of the D-port that equaled the signaling D-port, either primary or secondary. Refer to the <i>Good Data</i> field to determine which D-port.
<i>Good Data</i> =	This field will have one of the two values, 1 or 2 and has following meaning: 1 = The <i>sig_port</i> field was equal to the primary D-port and was not owned by any process. 2 = The <i>sig_port</i> field was equal to the secondary D-port and was not owned by any process.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIORTLA tuple of the D-port, either primary or secondary. Refer to <i>Good Data</i> for further information.

Table 16 Error Code: MKNUL

Error Code:	MKNUL
Error Description:	Audit found an error condition where both the D-ports of the rIPRISTAT tuple were null, that is TMNULL_PORT, and either the signaling D-port, <i>sig_port</i> , was not null or the <i>block_call</i> bit was not set.
Possible Error Effect:	Calls may originate on this PRI group without any D-port in service.
Corrective Action Taken:	The audit will report the error, null out the <i>sig_port</i> field, set the <i>block_call</i> bit to DBYES and return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>block_call</i> field of the rIPRISTAT tuple.
<i>Bad Data</i> =	The member field of the <i>sig_port</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 17 Error Code: MODE

Error Code:	MODE
Error Description:	An rIPRISTAT tuple contains an invalid mode (either AUTO or MANUAL) for either the primary or secondary D-channel port in rIPRISTAT.dchstat. When the error address is 1 or 2, it is checking the primary port 3 or 4, the secondary

	port. When the error address is 1 or 3, the primary or secondary port should be MANUAL but is instead AUTO when it is 2 or 4, the primary or secondary port should be AUTO.
Possible Error Effect:	The mode of this port on the PRI group will be reported incorrectly.
Corrective Action Taken:	The audit sets the mode appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1-4.
<i>Bad Data =</i>	The mode from rIPRISTAT.dchstat.
<i>Good Data =</i>	The mode from the rIORTLA.
<i>Logical Index =</i>	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1 =</i>	The rIPRISTAT tuple being audited.
<i>Dump 2 =</i>	When error address is 1 or 2, the rIORTLA tuple of the primary D-channel port. When error address is 3 or 4, the rIORTLA tuple of the secondary D-channel port.

Table 18 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	An rIPRISTAT tuple is reporting either the primary or secondary D-channel port as not provisioned when it is. When the error address is 1, it is checking the primary port 2, the secondary port.
Possible Error Effect:	The provisioning of this port on the PRI group will be reported incorrectly to the customer.
Corrective Action Taken:	The audit sets the provisioning appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1-2.
<i>Bad Data =</i>	The provisioning from rIPRISTAT.dchstat.
<i>Good Data =</i>	When error address is 1, the primary port (rIPRISTAT.dprim_port) when error address is 2, the secondary port (rIPRISTAT.dsec_port).
<i>Logical Index =</i>	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump =</i>	The rIPRISTAT tuple being audited.

Table 19 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	This error represents various error conditions related to the signaling port, <i>sig_port</i> , field of the rIPRISTAT tuple. To determine what error condition, caused this error, refer to the <i>Good Data</i> field and compare it with the explanation given in this field for each value.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The member field of the <i>sig_port</i> field will be set to null value, the <i>block_call</i> bit will be set to DBYES, and both the D-ports will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field will have one of the two values, and it has following meaning: - GLFAIL or -1 means there was only one D-port available to this PRI group or in other words, this is an NFAS case. - If it is not GLFAIL(-1), then it is the <i>list</i> field of the primary D-port's rIORTLA tuple.
<i>Bad Data =</i>	This field will have one of the two values, and it has following meaning: - If the address field above has a value of GLFAIL(-1), then this is <i>list</i> of the rIORTLA tuple of the only D-port that this rIPRISTAT tuple has. - If the address field above does not have a value of GLFAIL(-1), then this is the <i>list</i> field of the rIORTLA tuple of the secondary D-port.

<i>Good Data</i> =	This field will has four different values, and it has following meaning: 1 = The audit found an error where the <i>sig_port</i> field has a null value (TMNULL_PORT), neither primary nor the secondary D-port is owned by a process and either primary or secondary D-port is on the in service list(the <i>list</i> field of the rIORTLA is TMLTIS). 2 = The audit found an error where the <i>sig_port</i> field is equal to the primary D-port but the primary D-port is not on the in service list, that is the <i>list</i> field of the rIORTLA of the primary D-port in not TMLTIS. 3 = The audit found an error where the <i>sig_port</i> field is equal to the secondary D-port but the secondary D-port is not on the in service list, that is the <i>list</i> field of the rIORTLA of the secondary D-port in not TMLTIS. 4 = The audit found an error where the <i>sig_port</i> field has non null value and yet it is neither equal to the primary nor the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	If the address field has a value of GLFAIL(-1), then this is the dump of the rIORTLA tuple of the only D-channel this rIPRISTAT has, either primary or secondary. Else, this is the dump of the rIORTLA tuple of the primary D-port.
<i>Dump 3</i> =	This dump will be there only in the DCBU case, that is the address field does not have a GLFAIL(-1) value. It is the dump of the rIORTLA tuple of the secondary D-port.

Table 20 Error Code: NOPORTLA

Error Code:	NOPORTLA
Error Description:	The <i>dprim_port</i> field of the rIPRISTAT tuple has been validated form the static rIPRIGRP tuple. And yet audit could not find a valid rIORTLA tuple for this D-port.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will report the error and return. This will cause the PORTLA audit to run as a related audit to PRISTAT audit and should correct the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The <i>port</i> field of the rIORTLA tuple that audit tried to access.
<i>Good Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple that was used as a key to access the rIORTLA tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIORTLA tuple that was unsuccessfully accessed.

Table 21 Error Code: NOT_ZERO

Error Code:	NOT_ZERO
Error Description:	An rIPRISTAT tuple is reporting either the primary or secondary D-channel port as provisioned when it is not. When the error address is 1, it is checking the primary port 2, the secondary port.
Possible Error Effect:	The provisioning of this port on the PRI group will be reported incorrectly to the customer.
Corrective Action Taken:	The audit sets the provisioning appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1-2.
<i>Bad Data</i> =	The provisioning from rIPRISTAT.dchstat.
<i>Good Data</i> =	When error address is 1, the primary port (rIPRISTAT.dprim_port) when error address is 2, the secondary port (rIPRISTAT.dsec_port).

<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 22 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The <i>t321_job_id</i> field of the rIPRISTAT tuple has a valid ISPA job_id and it has been verified. Meaning the t321_job has been queued. This job is queued when a t321 timer expires. And yet the <i>t321_idx</i> field is set meaning a t321 timer is running.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will clear the t321 timer and set the <i>t321_idx</i> field to a null value, TMNULLIDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>list</i> field of the rPORTLA tuple of the primary D-port.
<i>Bad Data</i> =	The <i>t321_idx</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The <i>list</i> field of the rPORTLA tuple of the secondary D-port.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 23 Error Code: ODDLOSTUP

Error Code:	ODDLOSTUP
Error Description:	Using the <i>dsec_port</i> of the rIPRISTAT tuple being audited as the key, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Look at the Manual Action Required field below for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Look at the dump of the rIPRISTAT tuple from this error code. Find out the value of <i>dsec_port</i> field and try to insert an rIDCHNPRI tuple to static ODD. As for the other fields, consult other areas including terminal maintenance and triggers.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The member field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dsec_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 24 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	There are six different fields in the rIPRISTAT tuple that represents an ISPA job_id when an ISPA job is queued to be scheduled. These fields are, <i>t321_job_id</i> , <i>mfe1_job_id</i> , <i>mfe2_job_id</i> , <i>mfr1_job_id</i> , <i>mfr2_job_id</i> , and <i>bchan_job_id</i> . When audit tried to audit one of these fields, the job_id field was found to have neither a valid value within allowable range which is 0 through less than TMNULLJOBID nor it is null (TMNULLJOBID). As to which field is in error, refer to the address field below for an explanation.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The job_id field that was in error will be reset to a null value, TMNULLJOBID.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field can have one of the six different values and it has following meanings: 1 = The <i>t321_job_id</i> field was in error. 2 = The <i>mfe1_job_id</i> field was in error. 3 = The <i>mfe2_job_id</i> field was in error.

	4 = The <i>mfr1_job_id</i> field was in error.
	5 = The <i>mfr2_job_id</i> field was in error.
	6 = The <i>bchan_job_id</i> field was in error.
<i>Bad Data</i> =	The <i>job_id</i> field of the of rIPRISTAT tuple tuple that was in error Refer to the address field value and its explanation as to which field was in error.
<i>Good Data</i> =	The null value of the ISPA <i>job_id</i> , TMNULLJOBID.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 25 Error Code: PORT

Error Code:	PORT
Error Description:	Using the non-null D-port field of the rIPRISTAT tuple as the key, either the <i>dprim_port</i> or <i>dsec_port</i> , audit tried to read the static rIDSLEQUIP tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Look at the Manual Action Required field below for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Look at the dump of the rIPRISTAT tuple from this error code. Find out the value of member field of <i>dprim_port</i> field. If this field was null, find out the member field of the <i>dsec_port</i> field. Using the member field as a key, try to insert an rIDSLEQUIP tuple to static ODD. As for the other fields, consult other areas including triggers.
Dump Description:	
<i>Error Address</i> =	The D-port used to access the rIDSLEQUIP tuple.
<i>Bad Data</i> =	The <i>is_dte</i> field of the rIPRISTAT tuple.
<i>Good Data</i> =	The <i>dsl</i> field(key) of the rIDSLEQUIP tuple. This should be same as the value in <i>Error Address</i> in normal cases.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 26 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve an rIGPFTIMER from the key stored in one of the rIPRISTAT tuple's timer key fields.
Possible Error Effect:	Varies according to conditions including which timer field was being accessed.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data</i> =	The rIGPFTIMER key returned from the read attempt.
<i>Good Data</i> =	The rIGPFTIMER key from the timer key field.
<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1</i> =	The rIPRISTAT tuple.
<i>Dump 2</i> =	The rIGPFTIMER tuple buffer resulting form the read attempt.

Table 27 Error Code: RELQKEY

Error Code:	RELQKEY
Error Description:	This error refers to an error related to one of the five fields of the rIPRISTAT tuple, namely, <i>mfe1_job_id</i> , <i>mfe2_job_id</i> , <i>mfr1_job_id</i> , <i>mfr2_job_id</i> and <i>bchan_job_id</i> . All these field represent an ISPA job that has been queued to be processed. When audit finds these fields having a valid <i>job_id</i> , it verifies that the job is indeed queued through a TM interface. Audit reports this error when it receives a return value of anything other than GLSUCCESS. As to which field was in error, refer to the good data field below for an explanation.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The job_id field that was found in error will be updated with the null value, TMNULLJOBID. Refer to good data field for the job_id field found in error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The return value from the TM interface.
<i>Bad Data =</i>	The job_id field of rIPRISTAT tuple that was in error. Refer to the good data field below to determine which field is this.
<i>Good Data =</i>	This field represents five different values of the job_ids and it has following meanings: 1 = The mfe1_job_id field was in error. 2 = The mfe2_job_id field was in error. 3 = The mfr1_job_id field was in error. 4 = The mfr2_job_id field was in error. 5 = The bchan_job_id field was in error.
<i>Logical Index =</i>	This field has five different values depending upon the value found in the good data field above and it has following meanings: • If the value in the good data field is 1, then this field has a value of the member field of the dprim_port field of rIPRISTAT tuple. • If the value in the good data field is 2, then this field has a value of the member field of the dsec_port field of rIPRISTAT tuple. • If the value in the good data field is 3, then this field has a value of the member field of the dprim_port field of rIPRISTAT tuple. • If the value in the good data field is 4, then this field has a value of the member field of the dsec_port field of rIPRISTAT tuple. • If the value in the good data field is 5, then this field has a value of the member field of the dprim_port field of rIPRISTAT tuple if it is not null else it is the member field of the dsec_port field.
<i>Dump 1 =</i>	The rIPRISTAT tuple being audited.
<i>Dump 2 =</i>	The rPORTLA tuple of the primary D-port if it is not null else it is the rPORTLA tuple of the secondary D-port.

Table 28 Error Code: RL_MISMATCH

Error Code:	RL_MISMATCH
Error Description:	The module field of the sig_port field of the rIPRISTAT tuple did not match the same field of either the primary D-port or the secondary D-port. As to which D-port it should have matched, look at the address field below.
Possible Error Effect:	The PRI calls may fail.
Corrective Action Taken:	The module field of the sig_port field will be updated with the module field of either of the D-port as applicable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field will have one of the two values, 1 or 2 and has following meaning: 1 = The member field of the sig_port field was equal to the member field of the primary D-port and the module field of sig_port does not match the

	module field of the primary D-port.
	2 = The member field of the <i>sig_port</i> field was equal to the member field of the secondary D-port and the module field of <i>sig_port</i> does not match the module field of the secondary D-port.
<i>Bad Data</i> =	The module field of the <i>sig_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the D-port equaling signaling D-port. Refer to the <i>Error Address</i> field.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 29 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	Using the <i>dprim_port</i> of the rIPRISTAT tuple being audited as the key, audit tried to read the static rIDCHNPRI tuple and failed. Audit must find this static ODD tuple and will continue to take this error until it finds it. Look at the Manual Action Required field below for more information.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Report the error and return.
Manual Action Required:	Look at the dump of the rIPRISTAT tuple from this error code. Find out the value of <i>dprim_port</i> field and try to insert an rIDCHNPRI tuple to static ODD. As for the other fields, consult other areas including terminal maintenance and triggers.
Dump Description:	
<i>Error Address</i> =	The address of the rIPRISTAT tuple being audited.
<i>Bad Data</i> =	The member field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The module field of the <i>dprim_port</i> field of rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

Table 30 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	An rIPRISTAT tuple contains a bad status - either in-service (IS) or out-of-service (OOS) for either the primary or secondary D-channel port in rIPRISTAT.dchstat. When the error address is 1 or 2, it is checking the primary port 3 or 4, the secondary port. When the error address is 1 or 3, the primary or secondary port should be OOS but is instead IS when it is 2 or 4, the primary or secondary port should be IS.
Possible Error Effect:	The status of this port on the PRI group will be reported incorrectly to the customers.
Corrective Action Taken:	The audit sets the status appropriately.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1-4.
<i>Bad Data</i> =	The status from rIPRISTAT.dchstat.
<i>Good Data</i> =	The status from the rIPORTLA (rIPORTLA.bsc_state).
<i>Logical Index</i> =	The rIPRISTAT tuple logical key (rIPRISTAT.prigrp).
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	When error address is 1 or 2, the rIPORTLA tuple of the primary D-channel port. When error address is 3 or 4, the rIPORTLA tuple of the secondary D-channel port.

Table 31 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The <i>t321_idx</i> field of the rIPRISTAT tuple indicates whether a T321 timer is running or not and one of the D-channel is in IS_WAIT state. The <i>tservice_idx</i> field also indicates whether a T321 timer is running or not, but the D-channel is in IS state. Both timers are applicable only in the D-channel Back up case and should have a null value in all other cases. But audit found either one of the

	timer to have non-null value when this PRI did not have a D-channel back up. As to which timer has non-null value, look at the address field below.
Possible Error Effect:	The D-channels may get re-initialized.
Corrective Action Taken:	The <i>t321_idx</i> field or <i>tservice_idx</i> field will be reset to its null value, TMTNULLIDX, depending upon which ever was non-null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field will have one of the following values: 1 = The <i>t321_idx</i> field has a non-null value. 2 = The <i>tservice_idx</i> field has a non-null value.
<i>Bad Data</i> =	The <i>t321_idx</i> or <i>tservice_idx</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The null value of the timer index, TMTNULLIDX.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 32 Error Code: TIMERLST

Error Code:	TIMERLST
Error Description:	The <i>t321_idx</i> field of the rIPRISTAT tuple indicates whether a T321 timer is running or not and is applicable only in the D-channel back up case. When this field is set, the timer is running and one of the two D-ports must be in on the WAIT list. When audit made list check, it did not find any one of the two D-ports on the TMLTWAIT list when the <i>t321_idx</i> field was set.
Possible Error Effect:	The D-channels may get re-initialized.
Corrective Action Taken:	The T321 timer will be stopped and the <i>t321_idx</i> field will be reset to its null value, TMTNULLIDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The list field of the rIPORTLA tuple of the primary D-channel of the rIPRISTAT tuple.
<i>Bad Data</i> =	The <i>t321_idx</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The list field of the rIPORTLA tuple of the secondary D-channel of the rIPRISTAT tuple.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump 1</i> =	The rIPRISTAT tuple being audited.
<i>Dump 2</i> =	The rIPORTLA tuple of the primary D-port.
<i>Dump 3</i> =	The rIPORTLA tuple of the secondary D-port.

Table 33 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The <i>sig_int_id</i> field of the rIPRISTAT tuple did not match either the <i>interface_id</i> (DFI) of primary D-port or <i>interface_id</i> of secondary D-port. For further information, refer to the <i>Error Address</i> field.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The <i>sig_int_id</i> field will be updated with the <i>interface_id</i> of either of the D-port as applicable.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	This field will have one of the following values: 1 = The <i>sig_port</i> field was equal to the primary D-port and the <i>sig_int_id</i> does not match the <i>interface_id</i> of primary D-port. 2 = The <i>sig_port</i> field was equal to the secondary D-port and the <i>sig_int_id</i> does not match the <i>interface_id</i> of secondary D-port.

<i>Bad Data</i> =	The <i>sig_int_id</i> field of rIPRISTAT tuple.
<i>Good Data</i> =	The <i>interface_id</i> of the D-port equaling signaling D-port. For further information, refer to the <i>Error Address</i> field.
<i>Logical Index</i> =	Key of the rIPRISTAT tuple being audited, <i>prigrp</i> field.
<i>Dump</i> =	The rIPRISTAT tuple being audited.

PROAD

Software Release: 5E10 and later

1. PROCESS ADMINISTRATION (PROAD) AUDIT

The process administration (PROAD) audit verifies the integrity of the operating system process administration procedures (process fork, creation, and termination) and terminates incompletely created or terminated processes. This situation is determined by the length of time spent in any of these transient states. It also terminates processes that have been suspended from executing for over 30 minutes.

If the PROAD audit is run in correcting mode, it will complete the termination of any incompletely terminated or created processes. Otherwise, it will only list information about such processes that it found.

This audit is initiated routinely by the system integrity monitor (SIM) process and should be run in the correcting mode. This is recommended since problems with process creations or terminations can tie up critical system resources. This audit should rarely report an error. If the audit begins to report errors regularly, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

1.1 Requesting the PROAD Audit

To request the PROAD audit, use the following input message.

```
AUD:PROAD=1[:a|:b|:a,b];
```

where:

a = Indicates whether audit should correct errors. Valid values are:

CORR = Audit should correct errors (default).

NCG = Audit should not correct errors.

b = Indicates what type of report or message to generate. Valid values are:

DETL = Generates a raw data report for each error found and an audit completion message (default). A limited number of raw data reports are generated.

SUM = Generates only an audit completion message when the audit terminates. Raw data reports are not generated for any errors that are found.

To interpret system responses to the AUD:PROAD input message, refer to the APP:AUD Appendix in the Appendixes section of the Input Messages manual (235-600-700).

1.2 Interpreting AUD PROAD Output Messages and Error Reports

The AUD:PROAD messages report the termination status of the process administration audit and/or any errors found while executing the audit. For detailed explanations of the PROAD output messages, refer to the AUD:PROAD output message in the Output Messages manual (235-600-750).

Table 1 lists the PROAD audit error messages and their associated categories.

Error categories include the following.

- Category 1 = Errors that are associated with a breakdown in process termination. This includes processes that were suspended as a precursor to termination. Recovery action when maximum is exceeded -- abort audit.
- Category 2 = Errors that are associated with a breakdown in process creation or fork. Recovery action when maximum is exceeded -- abort audit.

Table 1 PROAD AUDIT ERRORS

Category	Error Number	Meaning
1	100	DCTE use count was decremented (scheduler).
1	110	Term DCT was called (scheduler).
1	120	GRASP/DART was informed (scheduler).
1	130	Entry was unlinked from the dispatch chain (scheduler).
1	140	The address translation buffers (ATBs) were flushed (scheduler).
1	150	Acknowledgment message was allocated and copied (scheduler).
1	160	DCTE was unlinked from the ready/not ready list (scheduler).
1	170	Incarnation count was bumped scheduler).
1	180	Message sent to the capability manager (scheduler).
1	190	Message sent to the process manager (scheduler).
1	200	Capabilities were removed (CMGR).
1	210	Supervisor segments were removed (CMGR).
1	220	Kernel process segments were removed (CMGR).
1	300	Core dump started (PMGR).
1	310	Message sent to the capability manager (PMGR).
2	500	kpstart-pcreate kernel process (PMGR).
2	510	Kernel process DCTE linked to proper chain (PMGR).
2	520	Initialization event sent to kernel process (PMGR).
2	600	pstart-pcreate supervisor process (PMGR) has begun.
2	610	Scheduler awaken (SP) (PMGR).
2	620	pstart completed (SP) (PMGR).
2	630	Execute started (SP) (PMGR).
2	700	Fork started-pfork1 (LIBC).
2	710	dupcaps message sent to FMGR (LIBC).
2	720	pfork2 started (LIBC).
2	730	Sending wakeup event to forked process (LIBC).
1	800	Process suspended.

In summary:

- Errors 100-399 reflect a breakdown in process termination.
 - Errors 100-199 being states in the scheduler.
 - Errors 200-299 being states in the capability manager.
 - Errors 300-399 being states in the process manager.
- Errors 500-699 reflect a breakdown in process creation.
 - Errors 500-599 being states in kernel process creation.
 - Errors 600-699 being states in supervisor process creation.
- Errors 700-799 reflect a breakdown in process fork.
- Error 800 indicates that the process has been suspended from executing for over 30 minutes.

1.3 Responding to AUD:PROAD Error Reports

The recovery action for both categories of errors (process creation errors and process termination errors) is the same. When the error threshold is reached within the error time interval (threshold and interval specified in the auderr RC/V form), the following will occur. The audit will be run in demand mode by SIM, then if the threshold is reached again, the audit will be inhibited by SIM.

The recommended actions to take when many errors are being reported by the PROAD audit, are to save the output and call the technical assistance organization (refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual). By saving the output, you can tell which process and the stage of creation/termination when the errors started to be reported. As long as the audit is running in the correcting mode, then the system resources being used by the incompletely terminated or created process are being returned to the system and no system degradation should occur.

If the PROAD audit was run in the noncorrecting mode, run it again in the correcting mode. If the error persists, refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.

PRTLST

Software Release: 5E10 and later

1. PORT LIST HEAD CELL (PRTLST) AUDIT

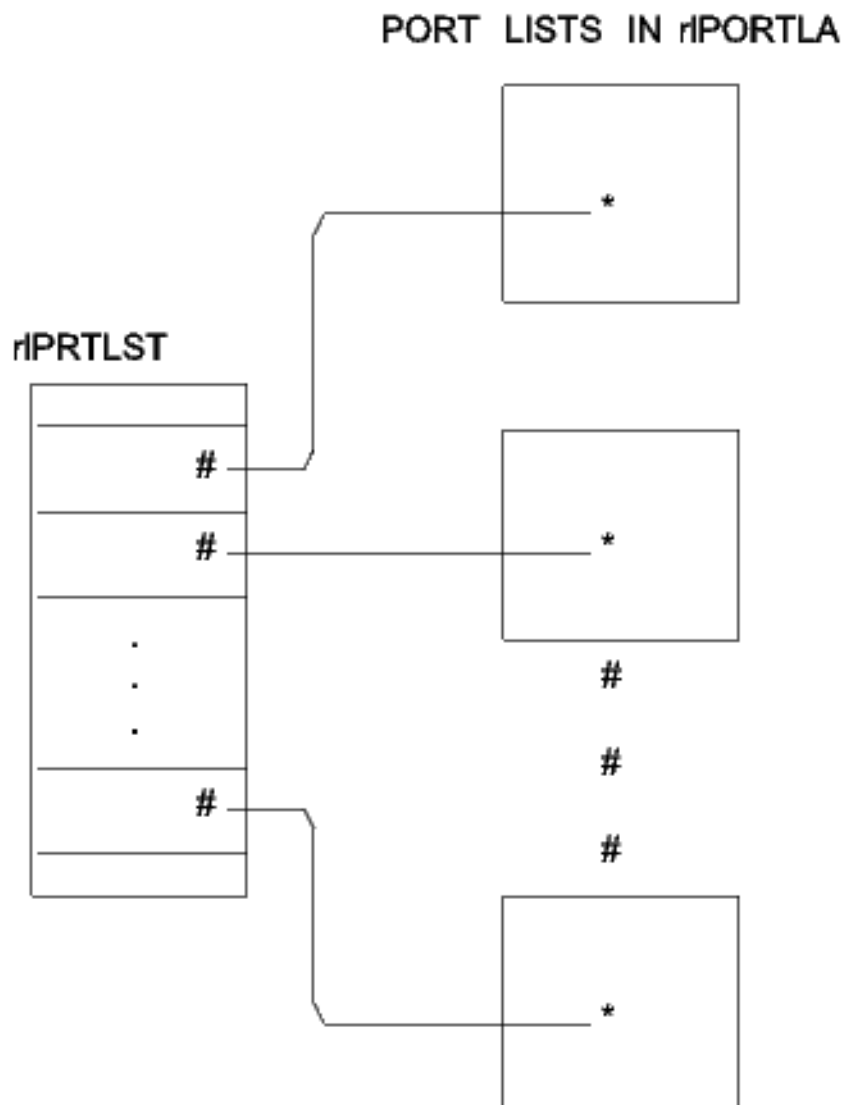
The terminal maintenance subsystem must know the status of ports in order to properly maintain resources. To do this, special tuples called head cells are used. A head cell indicates the first and last port on a port list. The port list relation (RLPRTLST) consists of port list head cells. Each head cell owns a list of port linkage areas. The elements of the list are port linkage areas tuples (rIPORTLA). Each list indicates a particular port maintenance status such as in service, out of service, to be idled, and so forth.

The port list head cells audit (PRTLST) verifies the linkage between the port list head cells and the first and last ports on the corresponding port lists. The head tuple in RLPRTLST links to the first port on the list, which links to other ports with the same status. The last port on the list links to the tail tuple in RLPRTLST and completes the chain. If this chain is broken, terminal maintenance would not be able to determine the status of the ports.

This audit is part of the routing and terminal allocation audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structures checked by the PRTLST audit are the port list relation (RLPRTLST) and the port linkage area relations (RLPORTLA). The RLPRTLST relation has owner linkages with RLPORTLA. Figure 1 shows the linkages established between the two relations.



SM

CP

————— * = Owner-Member Linkage

Figure 1 (PRTLST-1) Port List Head Cell Linkages

3. AUDIT CHECKS

To verify the linkages in the port list relation (RLPRTLST), the audit calls the linkage driver (DRLINK) once for each audit segment. The DRLINK driver verifies that each head cell in RLPRTLST has a valid owner-member linkage to an rPORTLA tuple.

Since the port list head cells contain only linkage, no semantic checks are performed.

4. ERROR HANDLING

All errors result in output messages. If one of the head cells contains erroneous data, it will be zeroed and the list reestablished when the port linkage area audit (PORTLA) runs.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

PSHC-A

Software Release: 5E10 - 5E11

1. PACKET SWITCH HEAD CELL (PSHC) AUDIT

The PSHC audit is part of the Switch Maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes. It audits different headcell structures of the package switching data.

2. DATA STRUCTURES

The PSHC audit uses the packet switching headcell array (PSpshc) and the following arrays:

- LAPB logical link control block (PSrbllcbhc).
- LAPD logical link control block (PSrlcbhc).
- Access link control block (PSralcbhc).
- Internal frame descriptor (PSrfdhd).
- Logical channel control block (PSrlccbhc[0]).
- Logical channel control block billing queue (PSrlccbhc[1]).
- The level 2 timeout head cell (PStimeout).
- Access link data block (PSraldbhc).

3. AUDIT CHECKS

The audit makes linkage checks for the headcell structures of the PS data. It checks the forward (or first) and backward (or last) fields of the headcell, which form linkages.

4. ERROR HANDLING

The audit has different functions to reconstruct different free lists. If an error is found in the headcell structures. These functions are also called by other PM audits.

The audit produces the following error code(s).

Table 1 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	An error was found in the owner field of either the first or last field of the LLCB linked list headcell.
Possible Error Effect:	Creates an error dump.
Corrective Action Taken:	This returns AUFALSE which reinitializes the free linked list owned by the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	"owner" value of the first element of the headcell.
<i>Bad Data =</i>	"owner" value of the last element of the headcell.
<i>Good Data =</i>	Valid "owner" value expected (timing list, timeout list, LAPB free linked list, LAPD free linked list).
<i>Logical Index =</i>	AUcount - indicates:

	4 = LAPD free linked list headcell.
	5 = LAPB free linked list headcell.
<i>Dump =</i>	LLCB free linked headcell (PSRHEADCELL type).

Table 2 Error Code: HEAD

Error Code:	HEAD
Error Description:	An error was found in the headcell of the specified data structure.
Possible Error Effect:	Creates an error dump.
Corrective Action Taken:	This returns AUFALSE which reinitializes the free linked list owned by the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	AUcount - the count indicating which headcell: 1 = IFD free linked list headcell. 2 = LLCB timeout linked list headcell. 3 = LAPB LLCB free linked list headcell. 4 = LAPD LLCB free linked list headcell. 5 = ALCB free linked list headcell. 6 = LCCB free linked list headcell. 7 = LCCB billing linked list headcell. 8 = ALDB free linked list headcell.
<i>Good Data =</i>	Linked list null value.
<i>Logical Index =</i>	Linked list owner.
<i>Dump =</i>	AULNK type as defined in au/AUps/AUextern.h (all size unsigned short): • headcell's first element, • headcell's last element, • current element being audited, • the "back" struct.'s owner field, • the "back" struct.'s next field, • the "back" elements back field, • the current element's owner, • the current element on this linked list, • the current element's back field, • the next element's owner field, • the "next" element on this linked list,

	- the next element's back field, - headcell null value, - data element null value. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and so is filled with 0xaa.

Table 3 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The specified headcell's first or last elements are out of range.
Possible Error Effect:	Prints an error report and continues.
Corrective Action Taken:	AUFALSE is returned and the headcell is nulled. The linked list owned by the headcell is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting range for the first/last headcell element.
<i>Bad Data</i> =	Ending range for the first/last headcell element.
<i>Good Data</i> =	Null value of the first and last.
<i>Logical Index</i> =	AUcount - the count indicating which headcell: 1 = IFD free linked list headcell. 2 = LLCB timeout linked list headcell. 3 = LAPB LLCB free linked list headcell. 4 = LAPD LLCB free linked list headcell. 5 = ALCB free linked list headcell. 6 = LCCB free linked list headcell. 7 = LCCB billing linked list headcell index. 8 = ALDB free linked list headcell.
<i>Dump</i> =	Head cell dump or PSRHEADCELL type.

Table 4 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	An error was found in the LLCB timing or timeout headcell.
Possible Error Effect:	Creates an error dump.
Corrective Action Taken:	This returns AUFALSE which reinitializes the linked list owned by the headcell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	AUcount - the count indicating which headcell: 4 = LAPD linked list headcell. 5 = LAPB linked list headcell.
<i>Good Data</i> =	Linked list null value.
<i>Logical Index</i> =	Linked list owner.
<i>Dump</i> =	AULNK type as defined in au/AUps/AUextern.h (all size unsigned short):

	• headcell's first element, • headcell's last element, • current element being audited, • the "back" struct.'s owner field, • the "back" struct.'s next field, • the "back" elements back field, • the current element's owner, • the current element on this linked list, • the current element's back field, • the next element's owner field, • the "next" element on this linked list, • the next element's back field, • headcell null value, • data element null value. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and so is filled with 0xaa.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUrunning, Audit ID that was running at the time of the interrupt.
<i>Dump</i> =	None.

PSHC-B

Software Release: 5E12 and later

1. PACKET SWITCH HEAD CELL (PSHC) AUDIT

The PSHC audit is part of the switch maintenance audits subgroup. It is part of the integrated services digital network packet (ISDNPKT) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes. It audits different head cell structures of the package switching data.

2. DATA STRUCTURES

The PSHC audit uses the packet switching head cell array (PSpshc) and the following arrays:

- Access link control block (PSralcbbc).
- Logical channel control block (PSrlccbbc[0]).
- Logical channel control block billing queue (PSrlccbbc[1]).
- Access link data block (PSrldbhc).

3. AUDIT CHECKS

The audit makes linkage checks for the head cell structures of the PS data. It checks the forward (or first) and backward (or last) fields of the head cell, which form linkages.

4. ERROR HANDLING

The audit has different functions to reconstruct different free lists. If an error is found in the head cell structures. These functions are also called by other PM audits.

The audit produces the following error code(s).

Table 1 Error Code: HEAD

Error Code:	HEAD
Error Description:	An error was found in the head cell of the specified data structure.
Possible Error Effect:	Creates an error dump.
Corrective Action Taken:	This returns AUFALSE which reinitializes the free linked list owned by the head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	AUcount - the count indicating which head cell. Valid values are: 5 = ALCB free linked list head cell. 6 = LCCB free linked list head cell. 7 = LCCB billing linked list head cell. 8 = ALDB free linked list head cell.
<i>Good Data =</i>	Linked list null value.
<i>Logical Index =</i>	Linked list owner.
<i>Dump =</i>	AULNK type as defined in au/AUps/AUextern.h (all size unsigned short):

	• head cell's first element, • head cell's last element, • current element being audited, • the "back" struct.'s owner field, • the "back" struct.'s next field, • the "back" elements back field, • the current element's owner, • the current element on this linked list, • the current element's back field, • the "next" element's owner field, • the "next" element on this linked list, • the "next" element's back field, • head cell null value, • data element null value. NOTE: 0xaa is the default value and indicates either the field is not used or the values are out of range and the data cannot be read and so is filled with 0xaa.

Table 2 Error Code: HD_CELL

Error Code:	HD_CELL
Error Description:	The specified head cell's first or last elements are out of range.
Possible Error Effect:	Prints an error report and continues.
Corrective Action Taken:	AUFALSE is returned and the head cell is nulled. The linked list owned by the head cell is rebuilt.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Starting range for the first/last head cell element.
<i>Bad Data</i> =	Ending range for the first/last head cell element.
<i>Good Data</i> =	Null value of the first and last.
<i>Logical Index</i> =	AUcount - the count indicating which head cell. Valid values are: 5 = ALCB free linked list head cell. 6 = LCCB free linked list head cell. 7 = LCCB billing linked list head cell index. 8 = ALDB free linked list head cell.
<i>Dump</i> =	Head cell dump or PSRHEADCELL type.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up.
Possible Error Effect:	Single process purges the audit process.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	AUrunning, audit ID that was running at the time of the interrupt.
<i>Dump =</i>	None.

PSPIDBTS-A

Software Release: 5E10 - 5E11

1. PACKET SWITCHING UNIT PERIPHERAL INTERFACE DATA BUS TIME SLOT MAP (PSPIDBTS) AUDIT

The data structure audited by PSPIDBTS, PCpspidbts[], is a summary of the busy/idle status for the packet switching unit (PSU) and peripheral interface data bus (PIDB) time slots. Each shelf in the PSU contains one PIDB. The shelf number is the index into PCpspidbts[]. Each PCpspidbts[] entry is 32 bits long, one for each time slot on the given PIDB. Idle time slots are marked with '0' and busy with '1'.

2. DATA STRUCTURES

PCpspidbts[].

RLPSIUPIDB.

RLPTSB.

3. AUDIT CHECKS

RLPTSB is a dynamic relation which contains one tuple for each PSU PIDB time slot. The busy/idle status kept in PCpspidbts[] is a summary of the rIPTSB data. Thus the PSPIDBTS audit loops through all PCpspidbts[] entries, comparing each bit with the busy/idle status derived from the corresponding rIPTSB tuple. For PSU shelves which are not equipped, the PSPIDBTS audit locates all time slots marked unavailable for use (that is, busy).

4. ERROR HANDLING

Each bit in every PCpspidbts[] entry is recovered to match the status of the master data in RLPTSB.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADPTSB

Error Code:	BADPTSB
Error Description:	The rIPTSB tuple for the given PSU PIDB time slot is linked to an invalid rICBCOM tuple or the rIPTSB tuple is missing.
Possible Error Effect:	A channel is being tied up needlessly (i.e., lost resource).
Corrective Action Taken:	The bit in PCpspidbts for the given PSU shelf and PIDB time slot is marked idle (1). The PTSB audit will put the rIPTSB tuple on the correct list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rIPTSB tuple.
<i>Bad Data =</i>	Bad rICBCOM owner logical key.
<i>Good Data =</i>	rICBCOM shelf idle list logical key.
<i>Logical Index =</i>	rIPTSB key.
<i>Dump =</i>	rIPTSB tuple.

Table 2 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	The given PSU shelf is not equipped but the corresponding PCpspidbts entry is marked idle for at least one time slot.
Possible Error Effect:	PC might attempt to nail up a packet switching port using a non-existent PIDB time slot.

Corrective Action Taken:	The PCpspidbts entry for the given PSU shelf is marked busy (0) for all PIDB time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCpspidbts entry.
<i>Bad Data</i> =	Bad PCpspidbts entry.
<i>Good Data</i> =	Recovered value of PCpspidbts.
<i>Logical Index</i> =	Circuit name for the PSU shelf.
<i>Dump</i> =	NA.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIPTSB tuple for the given PSU PIDB time slot is on either the rCCBCOM busy list or OOS list but the corresponding PCpspidbts time slot bit is marked idle.
Possible Error Effect:	PC might attempt to nail up a packet switching channel using the given PIDB time slot which is already in use or OOS.
Corrective Action Taken:	The bit in PCpspidbts for the given PSU shelf and PIDB time slot is marked busy (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCpspidbts entry.
<i>Bad Data</i> =	Bad PCpspidbts entry.
<i>Good Data</i> =	Recovered value of PCpspidbts.
<i>Logical Index</i> =	rIPTSB key.
<i>Dump</i> =	rIPTSB tuple.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIPTSB tuple for the given PSU PIDB time slot is on the rCCBCOM shelf idle list but the corresponding PCpspidbts time slot bit is marked busy.
Possible Error Effect:	A PIDB time slot is being tied up needlessly (i.e., lost resource).
Corrective Action Taken:	The bit in PCpspidbts for the given PSU shelf and PIDB time slot is marked idle (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCpspidbts entry.
<i>Bad Data</i> =	Bad PCpspidbts entry.
<i>Good Data</i> =	Recovered value of PCpspidbts.
<i>Logical Index</i> =	rIPTSB key.
<i>Dump</i> =	rIPTSB tuple.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	The audit won't clean up until the next time it runs.
Corrective Action Taken:	The audit is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

PSPIDBTS-B

Software Release: 5E12 and later

1. PACKET SWITCHING UNIT PERIPHERAL INTERFACE DATA BUS TIME SLOT MAP (PSPIDBTS) AUDIT

The data structure audited by PSPIDBTS, PCpspidbts[], is a summary of the busy/idle status for the packet switching unit (PSU) and peripheral interface data bus (PIDB) time slots. Each shelf in the PSU contains one PIDB. The shelf number is the index into PCpspidbts[]. Each PCpspidbts[] entry is 32 bits long, one for each time slot on the given PIDB. Idle time slots are marked with '0' and busy with '1'.

2. DATA STRUCTURES

PCpspidbts[].

rIPSIUPIDB.

rIPTSB.

3. AUDIT CHECKS

RLPTSB is a dynamic relation which contains one tuple for each PSU PIDB time slot. The busy/idle status kept in PCpspidbts[] is a summary of the rIPTSB data. Thus the PSPIDBTS audit loops through all PCpspidbts[] entries, comparing each bit with the busy/idle status derived from the corresponding rIPTSB tuple. For PSU shelves which are not equipped, the PSPIDBTS audit locates all time slots marked unavailable for use (that is, busy).

4. ERROR HANDLING

Each bit in every PCpspidbts[] entry is recovered to match the status of the master data in RLPTSB.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADPTSB

Error Code:	BADPTSB
Error Description:	The rIPTSB tuple for the given PSU PIDB time slot is linked to an invalid rICBCOM tuple or the rIPTSB tuple is missing.
Possible Error Effect:	A channel is being tied up needlessly (that is, lost resource).
Corrective Action Taken:	The bit in PCpspidbts for the given PSU shelf and PIDB time slot is marked idle (1). The PTSB audit will put the rIPTSB tuple on the correct list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rIPTSB tuple.
<i>Bad Data =</i>	Bad rICBCOM owner logical key.
<i>Good Data =</i>	rICBCOM shelf idle list logical key.
<i>Logical Index =</i>	rIPTSB key.
<i>Dump =</i>	rIPTSB tuple.

Table 2 Error Code: NO_PIDB

Error Code:	NO_PIDB
Error Description:	The given PSU shelf is not equipped but the corresponding PCpspidbts entry is marked idle for at least one time slot.
Possible Error Effect:	PC might attempt to nail up a packet switching port using a non-existent PIDB time slot.

Corrective Action Taken:	The PCpspidbts entry for the given PSU shelf is marked busy (0) for all PIDB time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCpspidbts entry.
<i>Bad Data</i> =	Bad PCpspidbts entry.
<i>Good Data</i> =	Recovered value of PCpspidbts.
<i>Logical Index</i> =	Circuit name for the PSU shelf.
<i>Dump</i> =	NA.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rIPTSB tuple for the given PSU PIDB time slot is on either the rCCBCOM busy list or OOS list but the corresponding PCpspidbts time slot bit is marked idle.
Possible Error Effect:	PC might attempt to nail up a packet switching channel using the given PIDB time slot which is already in use or OOS.
Corrective Action Taken:	The bit in PCpspidbts for the given PSU shelf and PIDB time slot is marked busy (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCpspidbts entry.
<i>Bad Data</i> =	Bad PCpspidbts entry.
<i>Good Data</i> =	Recovered value of PCpspidbts.
<i>Logical Index</i> =	rIPTSB key.
<i>Dump</i> =	rIPTSB tuple.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rIPTSB tuple for the given PSU PIDB time slot is on the rCCBCOM shelf idle list but the corresponding PCpspidbts time slot bit is marked busy.
Possible Error Effect:	A PIDB time slot is being tied up needlessly (that is, lost resource).
Corrective Action Taken:	The bit in PCpspidbts for the given PSU shelf and PIDB time slot is marked idle (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of bad PCpspidbts entry.
<i>Bad Data</i> =	Bad PCpspidbts entry.
<i>Good Data</i> =	Recovered value of PCpspidbts.
<i>Logical Index</i> =	rIPTSB key.
<i>Dump</i> =	rIPTSB tuple.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	The audit won't clean up until the next time it runs.
Corrective Action Taken:	The audit is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

PSTSCKT

Software Release: 5E10 and later

1. PACKET SWITCHING UNIT TIME SLOT INTERCHANGE UNIT TIME SLOT CIRCUIT MAP (PSTSCKT) AUDIT

The data structure audited by PSTSCKT, PCpstsckt[][], is a summary of the busy/idle status for the packet switching unit (PSU) and time slot interchange unit (TSIU) time slots. Each data interchange (DI) in the TSIU contains up to four PSU time slot sets (PTSS) allocated for PSU to TSIU nail-up time slots. The DI and PTSS are the indices into PCpstsckt[][]. Each PCpstsckt[][] entry is 32 bits long, one for each time slot of the given PTSS. Idle time slots are marked with '0' and busy with '1'.

2. DATA STRUCTURES

PCpstsckt[][].

rICKTDATA.

rICDBCOM.

3. AUDIT CHECKS

RLCDBCOM is a dynamic relation which contains one tuple for each TSIU PSU time slot. The busy/idle status, kept in PCpstsckt[][], is a summary of the rICDBCOM data. Thus the PSTSCKT audit loops through all PCpstsckt[][] entries, comparing each bit with the busy/idle status derived from the corresponding rICDBCOM tuple. For time slots of PTSS which are not equipped, the PSTSCKT audit locates the time slot marked unavailable for use (that is, busy).

4. ERROR HANDLING

Each bit in every PCpstsckt[][] entry is recovered to match the status of the master data in RLCDBCOM.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The rICDBCOM tuple for the given PSU time slot circuit is linked to an invalid rICBCOM tuple or the rICDBCOM tuple is missing.
Possible Error Effect:	A channel is being tied up needlessly (that is, lost resource).
Corrective Action Taken:	The bit in PCpstsckt for the given DI, time slot set, and PIDB time slot is marked idle (1). The CDBCOM audit will put the rICDBCOM tuple on the correct list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rICDBCOM tuple.
<i>Bad Data =</i>	Bad rICBCOM owner logical key.
<i>Good Data =</i>	rICBCOM idle list logical key.
<i>Logical Index =</i>	PSU time slot circuit name.
<i>Dump =</i>	rICDBCOM tuple.

Table 2 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rICDBCOM tuple for the given PSU time slot circuit is on the rICBCOM busy list but the corresponding PCpstsckt time slot bit is marked idle.

Possible Error Effect:	PC might attempt to nail up a packet switching channel using the given time slot circuit which is already in use.
Corrective Action Taken:	The bit in PCpstsckt for the given DI, time slot set, and PIDB time slot is marked busy (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad PCpstsckt entry.
<i>Bad Data =</i>	Bad PCpstsckt entry.
<i>Good Data =</i>	Recovered value of PCpstsckt.
<i>Logical Index =</i>	PSU time slot circuit name.
<i>Dump =</i>	rICDBCOM tuple.

Table 3 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rICDBCOM tuple for the given PSU time slot circuit is on the rICDBCOM idle list but the corresponding PCpstsckt time slot bit is marked busy.
Possible Error Effect:	A channel is being tied up needlessly (that is, lost resource).
Corrective Action Taken:	The bit in PCpstsckt for the given DI, time slot set, and PIDB time slot is marked idle (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad PCpstsckt entry.
<i>Bad Data =</i>	Bad PCpstsckt entry.
<i>Good Data =</i>	Recovered value of PCpstsckt.
<i>Logical Index =</i>	PSU time slot circuit name.
<i>Dump =</i>	rICDBCOM tuple.

Table 4 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	RLCKTDATA is not equipped for a given PSU time slot circuit but the corresponding PCpstsckt time slot bit is marked idle.
Possible Error Effect:	PC might attempt to nail up a packet switching port using a non-existent TSIU time slot circuit channel.
Corrective Action Taken:	The bit in PCpstsckt for the given DI, time slot set, and PIDB time slot is marked busy (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad PCpstsckt entry.
<i>Bad Data =</i>	Bad PCpstsckt entry.
<i>Good Data =</i>	Recovered value of PCpstsckt.
<i>Logical Index =</i>	PSU time slot circuit name.
<i>Dump =</i>	NA.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	The audit won't clean up until the next time it runs.
Corrective Action Taken:	The audit is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

PSX25LN

Software Release: 5E10 and later

1. PACKET SWITCHING X25 LINE (PSX25LN) AUDIT

The PSX25LN audit performs a static-to-dynamic existence check against the dynamic integrated services digital network (ISDN) maintenance data block (ISMDB) relation. This audit is part of the ISDNBASE package and resides on the operating system for distributed switching-switching module (OSDS-SM). The PSX25LN audit is running at the segmented mode as well as the initialization mode during selective initialization.

2. DATA STRUCTURES

The RLPSX25LN is a static relation which defines level 3 data for a digital subscriber line (DSL) D/B channel.

3. AUDIT CHECKS

For every PSX25LN tuple, there should exist a corresponding ISMDB tuple. The ISMDB ID used for all applications that support ISDN protocols. Any missing ISMDB tuple will be inserted.

4. ERROR HANDLING

Any missing ISMDB tuple is inserted and initialized.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	For each tuple in PSX25LN, there should exist a corresponding ISMDB tuple.
Possible Error Effect:	No protocol monitor available on this channel.
Corrective Action Taken:	The tuple is initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the missing ISMDB.
<i>Bad Data</i> =	Key of missing ISMDB.
<i>Good Data</i> =	Expected ISMDB key.
<i>Logical Index</i> =	ISMDB key.
<i>Dump</i> =	Dump of the rIPSX25LN tuple.

PTDB

Software Release: 5E10 and later

1. PERIPHERAL TASK DATA BLOCK (PTDB) AUDIT

The peripheral task data block audit audits the peripheral control and timing (PCT) link based peripheral task data block structure used to send tasks from the switching module (SM) to communicate with different logical units. Initially, there is only one logical unit to which this task will be sent and it is called digital networking unit-synchronous optical network (SONET) (DNU-S).

The DNU-S is a new peripheral for the 5ESSTM switch that provides a cost of goods (COGS) reduction and SONET capabilities. These tasks are passed from SM to the second message handler to the common control (CC) of DNU-S.

The PTDB audit is based in the SMP processor and runs in segmented mode.

2. DATA STRUCTURES

The PTDB audit audits the PCptdbtbl[][] and PCptdbusage[] arrays. The PCptdbtbl[][] array is indexed by logical unit number and TDB number and contains the owning channel control block common area (CCBCOM) data structure along with the quarantine status. The PCptdbusage[] array consists of task allocation map per unit information. It is indexed by logical unit number and will contain the busy/idle status of the TDBs.

3. AUDIT CHECKS

The audit checks these arrays and prints audit errors if needed as follows:

BUSYIDLE	= Check for idle usage bit when PTDB is idle.
CCBCOM_LINK	= Point to point back check between CCB foreground region and PCptdbtbl[][].
IDLE_BUSY	= Check that the usage bit is busy.
NOCCB	= Check for successful retrieval of rCCBCOM tuple.
QUAR	= Check the quarantine status.
QUAR_INV	= Invalid value of quarantine field.
STUCK	= Check if the PTDB is stuck.
ZAP	= Check for system interrupt handling.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The PTDB under audit is idle as indicated by the PCINVALCCB value of the ccbnum field of the PCptdbtbl[AUunit][AUpctdb] array entry. But the

	corresponding usage bit in the PCptdbusage[AUnit] entry is not. AUnit represents the current logical unit number (such as DNU-S) and AUpctdb is the PTDB number.
Possible Error Effect:	Lost resource as this PTDB may not be used.
Corrective Action Taken:	The usage bit for this PTDB will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ccbnum field of PCptdbtbl[AUnit][AUpctdb] entry.
<i>Bad Data =</i>	The value of the bit corresponding to this PTDB in PCptdbusage[AUnit].
<i>Good Data =</i>	1, the correct value of the usage bit.
<i>Logical Key =</i>	AUnit in the upper two bytes and AUpctdb in the lower two bytes.
<i>Dump =</i>	The PCptdbtbl[AUnit][AUpctdb] array entry.

Table 2 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	Either the CCB foreground region does not point back to the current PTDB under audit represented by the PCptdbtbl[AUnit][AUpctdb] array entry or the rCCBCOM tuple is on the idle list. AUnit represents the current logical unit number (such as DNU-S) and AUpctdb is the PTDB number.
Possible Error Effect:	Lost resource as this PTDB may not be used.
Corrective Action Taken:	The PTDB is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The quar field of PCptdbtbl[AUnit][AUpctdb] entry.
<i>Bad Data =</i>	The ccbnum field of PCptdbtbl[AUnit][AUpctdb] entry.
<i>Good Data =</i>	The ccbnum field of rCCBCOM.
<i>Logical Key =</i>	AUnit in the upper two bytes and AUpctdb in the lower two bytes.
<i>Dump 1 =</i>	The rCCBCOM tuple associated with the PTDB.
<i>Dump 2 =</i>	The CCB foreground region, pcCCBFG, associated with this PTDB.

Table 3 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The PTDB under audit is in use as indicated by the valid value of the ccbnum field of the PCptdbtbl[AUnit][AUpctdb] array entry. But the corresponding usage bit in the PCptdbusage[AUnit] entry is not marked busy. AUnit represents the current logical unit number (such as DNU-S) and AUpctdb is the PTDB number.
Possible Error Effect:	This PTDB may be thought to be not in use when it is.
Corrective Action Taken:	The PTDB is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ccbnum field of PCptdbtbl[AUnit][AUpctdb] entry.
<i>Bad Data =</i>	The value of the bit corresponding to this PTDB in PCptdbusage[AUnit].
<i>Good Data =</i>	NA.
<i>Logical Key =</i>	AUnit in the upper two bytes and AUpctdb in the lower two bytes.
<i>Dump =</i>	The PCptdbtbl[AUnit][AUpctdb] array entry.

Table 4 Error Code: NOCCB

Error Code:	NOCCB
Error Description:	The rCCBCOM tuple just obtained is not the correct rCCBCOM tuple as the ccbnum fields of rCCBCOM and of the PCptdbtbl[AUnit][AUpctdb] array entry do not match. AUnit represents the current logical unit number (such as DNU-S) and AUpctdb is the PTDB number.
Possible Error Effect:	The task represented by this PTDB may not be performed.
Corrective Action Taken:	The PTDB is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The quar field of PCptdbtbl[AUnit][AUpctdb] entry.
<i>Bad Data =</i>	The ccbnum field of rCCBCOM.
<i>Good Data =</i>	The ccbnum field of PCptdbtbl[AUnit][AUpctdb] entry.
<i>Logical Key =</i>	AUnit in the upper two bytes and AUpctdb in the lower two bytes.
<i>Dump =</i>	None.

Table 5 Error Code: QUAR

Error Code:	QUAR
Error Description:	The PTDB is quarantined as indicated by the quar field in the task ownership table, PCptdbtbl[AUunit][AUptdb] array entry. AUunit represents the current logical unit number (such as DNU-S) and AUptdb is the PTDB number.
Possible Error Effect:	The task represented by this PTDB will not be performed.
Corrective Action Taken:	The PTDB is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ccbnum field of PCptdbtbl[AUunit][AUptdb] entry.
<i>Bad Data =</i>	The quar field of PCptdbtbl[AUunit][AUptdb] entry.
<i>Good Data =</i>	The good value of the quar field, AUNOTSET.
<i>Logical Key =</i>	AUunit in the upper two bytes and AUptdb in the lower two bytes.
<i>Dump =</i>	None.

Table 6 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The PTDB's quar field in the task ownership table entry, PCptdbtbl[AUunit][AUptdb], has an invalid value. AUunit represents the current logical unit number (such as DNU-S) and AUptdb is the PTDB number.
Possible Error Effect:	The task represented by this PTDB will not be performed.
Corrective Action Taken:	The PTDB is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ccbnum field of PCptdbtbl[AUunit][AUptdb] entry.
<i>Bad Data =</i>	The quar field of PCptdbtbl[AUunit][AUptdb] entry.
<i>Good Data =</i>	The good value of the quar field, AUNOTSET.
<i>Logical Key =</i>	AUunit in the upper two bytes and AUptdb in the lower two bytes.
<i>Dump =</i>	None.

Table 7 Error Code: STUCK

Error Code:	STUCK
Error Description:	The PTDB is being transient timed as indicated by the quar field in the task ownership table, PCptdbtbl[AUunit][AUptdb] array entry. But it is in this state longer than 5 minutes for an unacknowledged task and with a valid rCCBCOM tuple. AUunit represents the current logical unit number (such as DNU-S) and AUptdb is the PTDB number.
Possible Error Effect:	The task represented by this PTDB will not be performed.
Corrective Action Taken:	The PTDB is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ccbnum field of PCptdbtbl[AUunit][AUptdb] entry.
<i>Bad Data =</i>	The quar field of PCptdbtbl[AUunit][AUptdb] entry.
<i>Good Data =</i>	NA.
<i>Logical Key =</i>	AUunit in the upper two bytes and AUptdb in the lower two bytes.
<i>Dump =</i>	The task information structure for the CCB foreground region, jblSTASK, associated with this PTDB.

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audits will take this error as the result of a write protection violation, out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Audits may have caused the interrupt so action must be taken not to roll in the interrupts.
Corrective Action Taken:	PCptdbtbl[][] array may have caused audit to access out of range data so delete this entry and idle the usage bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the entry in PCptdbtbl[][] array with the AUunit and AUptdb as the indices.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Key</i> =	AUunit in the upper two bytes and AUpctdb in the lower two bytes.
<i>Dump</i> =	None.

PTQUE

Software Release: 5E10 and later

1. PICKUP QUEUE AUDIT (PTQUE) AUDIT

An interprocessor message is sent to the administrative module (AM) when a line having the pickup terminating feature enters the ringing state. The interprocessor message requests the placement of the line on the associated call pickup group's queue. A member tuple is removed from the idle list, initialized, and linked to the line's associated queue. When another line assigned to the pickup group having the pickup originating feature dials the pickup code (for example, *7), software in the AM removes the first member of the queue, which represents the longest ringing call, from the pickup group specified in the messages from the switching module (SM). The longest ringing call is then connected to the line performing the pickup. The pickup queue audit (PTQUE) ensures the consistency of the data structures involved in this mechanism.

The PTQUE audit is part of the routing and terminal allocation subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

This audit checks the owner-member linkage between the pickup owner relation (RLPUOWNER), a dynamic relation in the AM, and its member tuples in the pickup queue relation (RLPTQUE), a dynamic relation in the AM. It also checks that the number of consecutive non-vacant rIPTQUE tuples is at least that of the value returned by routing and terminal allocation (RTA) subsystem primitive (RTnoptque). Figure 1 depicts these data structures.

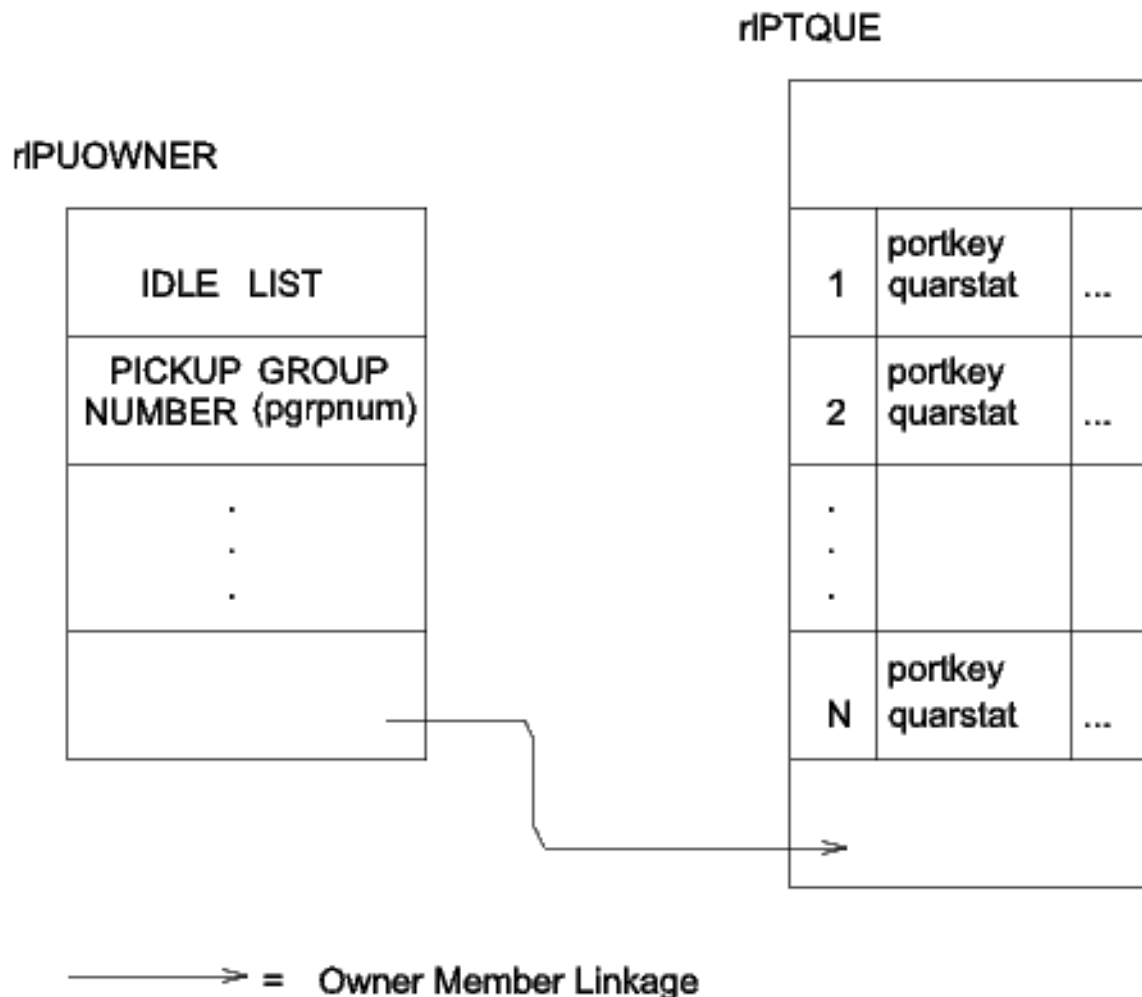


Figure 1 (PTQUE-1) Data Structures of the PTQUE Audit

3. AUDIT CHECKS

The PTQUE audit performs existence checks on tuples 1 through N in RLPTQUE, where N is the value returned by RTnoptque.

It corresponds to the number of members in RLPTQUE. For each non-vacant tuple, the audit performs syntactic checks by calling the AUtuple function. Neither the linkage driver nor the static driver can be used because rIPTQUE uses double indexing access and requires sequential keys.

The audit steps through rIPTQUE tuples using two data base functions. Since the number of non-vacant rIPTQUE tuples is office-dependent, the RTnoptque function is needed to calculate it. The rIPTQUE tuples must be consecutive starting at member 1. The number calculated by RTnoptque is two times the number of pickup groups.

4. ERROR HANDLING

If the audit finds vacant tuples in RLPTQUE (gaps in the 1 through N sequence), it inserts them into the idle list in tuple 1 of RLPUOWNER and schedules the pickup group assignment audit (PUGPAST).

If a syntactic check fails, the audit initializes the tuple, links it to the idle list, and schedules the PUGPAST audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rIPTQUE tuple was detected.
Possible Error Effect:	When recent change attempts to create a tuple, it will already exist.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key value.
<i>Dump</i> =	rIPTQUE tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing from rIPTQUE.
Possible Error Effect:	Call pickup queue entries may be denied.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address where tuple should have been.
<i>Bad Data</i> =	Valid values are: 1 = Indicates tuple was created. 0 = Indicates unable to create tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of lost tuple.
<i>Dump</i> =	NA.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets and/or asserts.
Corrective Action Taken:	Zero the current tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rIPTQUE tuple.

PTRDB

Software Release: 5E10 and later

1. POINTER DATA BLOCK (PTRDB) AUDIT

The pointer data block (PTRDB) audit verifies the physical linkages to the QRYDB data block and the TAPOSDB or ACCSDB data block.

In addition, the audit verifies that idle tuples are linked to a idle list head cell tuple in the RLDB_HEADC relation, and that for busy rIPTRDB tuple, there is no rIDB_HEADC owner tuple linked, and that there is a rIPCBLA tuple linked. In addition, for each busy rIPTRDB tuple, the audit verifies that there is a live process. The audit will verify the linkages to the rIQRYDB, and rITAPOSDB or rIACCSDB. This audit runs in the routine and elevated segmented modes.

2. DATA STRUCTURES

The rIPTRDB tuple and related data structures are shown in Figure 1 .

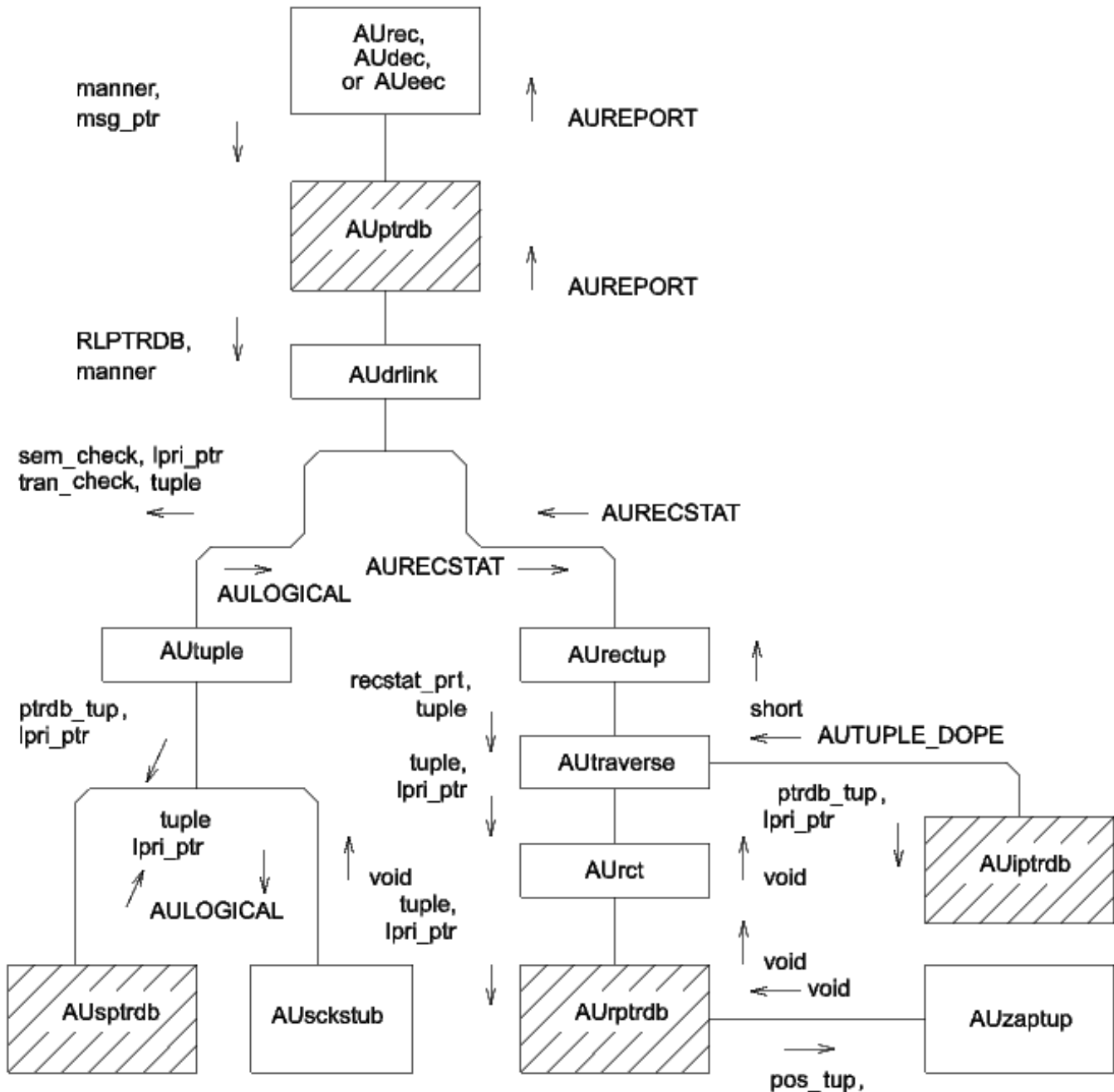


Figure 1 (PTRDB-1) PTRDB Audit Structure Chart

3. AUDIT CHECKS

The PTRDB audit performs the following:

- (1) Each initialized address field points to the proper tuple of the corresponding relation.
- (2) If a tuple goes through recovery the corresponding QRYDB is also recovered.
- (3) The pointer to the ACCSDB and the pointer to the TAPOSDB are never initialized at the same time.

- (4) The PTRDB tuple and its associated QRYDB tuple are checked to see that they share the same process number.
- (5) If there is a pseudo ACCS process that has been around more than five minutes, purges it.

4. ERROR HANDLING

When an error is detected in the PTRDB tuple, the audit will recover the bad tuple. The TAPOSDB, the ACCSDB and the QRYDB audits will also be scheduled to run in the elevated mode.

NOTE: If the RLPTTRDB points to a position and the TAPOSDB audit finds an error. The TAPOSDB audit will send a message to the ACD/AM to notify it that the position has been taken out of service.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	An idle rIPTRDB tuple was found linked to the RLDB_HEADC idle list head cell, but the other linkages in the tuple were not all zero. All idle rIPTRDB tuples (rIPTRDB.ptr_hdc.st_owrk - DBSFDB_HEA == DBPTRIDLE) must have all other linkages marked to zero, including the accsptr , tapptr and qryprr fields.
Possible Error Effect:	Possible exhaustion of OSPS operator position resources.
Corrective Action Taken:	The rIPTRDB tuple is recovered to an idle condition (all linkages are marked to zero, and the tuple is recovered to the RLDB_HEADC idle list head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTRDB tuple.
<i>Bad Data =</i>	For non-semantic linkage errors, the contents of the particular linkage that was found to be non-zero is dumped. Otherwise, zero.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTRDB tuple (rIPTRDB.procnum).
<i>Dump =</i>	The contents of the rIPTRDB tuple.

Table 2 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	The process number in the PCBLA (procnum) does not match the one in the PTRDB.
Possible Error Effect:	Possible asserts because data has become corrupted.
Corrective Action Taken:	The PCBLA tuple is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPTRDB.
<i>Bad Data =</i>	Address of rIPCBLA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPTRDB number.
<i>Dump =</i>	rIPTRDB.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A rIPTRDB tuple was found with a corrupted linkage to the RLDB_HEADC idle list head cell relation, or, the tuple was in a busy state but there was no associated rIPCBLA tuple linked. All rIPTRDB tuples should either be linked to the idle list head cell, or the rIPTRDB.ptr_hdc.st_owrk member linkage should be zero, and a rIPCBLA tuple should be linked using rIPTRDB.ptqrl.lnk.

Possible Error Effect:	Possible exhaustion of OSPS operator position resources.
Corrective Action Taken:	The rIPTRDB tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTRDB tuple.
<i>Bad Data =</i>	The contents of the linkage to the RLDB_HEADC head cell (rIPTRDB.ptr hdc.st ownk - DBSFDB HEA).
<i>Good Data =</i>	The linkage value the head cell linkage should be set to (DALNKNUL).
<i>Logical Index =</i>	The key of the rIPTRDB tuple.
<i>Dump =</i>	The contents of the rIPTRDB tuple.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	Either the first or third pointer in the PTRDB must be idle. This was not the case.
Possible Error Effect:	Someone didn't clean up or data corruption has occurred.
Corrective Action Taken:	The tuple is zeroed. Operator logged off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPTRDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPTRDB number.
<i>Dump =</i>	rIPTRDB.

Table 5 Error Code: PTR1

Error Code:	PTR1
Error Description:	The first pointer in the PTRDB relation is bad.
Possible Error Effect:	The ACCSDB relation that this relation links to may be inaccessible directly.
Corrective Action Taken:	The tuple is zeroed out and the operator is logged off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPTRDB.
<i>Bad Data =</i>	ptr1.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPTRDB number.
<i>Dump =</i>	rIPTRDB.

Table 6 Error Code: PTR2

Error Code:	PTR2
Error Description:	The second pointer in the PTRDB relation is bad.
Possible Error Effect:	The QRYDB this relation points to may be inaccessible directly.
Corrective Action Taken:	The tuple is zeroed out and the operator is logged off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPTRDB.
<i>Bad Data =</i>	ptr2.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPTRDB number.
<i>Dump =</i>	rIPTRDB.

Table 7 Error Code: PTR3

Error Code:	PTR3
Error Description:	The third pointer in the RLPTDRDB relation is bad.
Possible Error Effect:	The TAPOSDB this relation points to may be inaccessible directly.
Corrective Action Taken:	The tuple is zeroed out and the operator is logged off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPTRDB.
<i>Bad Data =</i>	ptr3.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPTRDB number.
<i>Dump =</i>	rIPTRDB.

Table 8 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	A pseudo ACCS process has been found in a stuck condition. The process should not be around for more than five minutes.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The PTRDB process number specified by ptrdb is zeroed and the process is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIPTRDB.
<i>Bad Data =</i>	Address of semi global data area.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rIPTRDB number.
<i>Dump =</i>	rIPTRDB.

Table 9 Error Code: TPID

Error Code:	TPID
Error Description:	A busy rIPTRDB tuple is not associated with a live terminal process. All busy rIPTRDB tuples (rIPTRDB.ptr_hdc.st_ownk == DALNKNUL) must be linked to an associated rIPCBLA tuple (rIPTRDB.ptqrl.lnk - DBSFPCBLA == rIPCBLA.procnum) which is associated with a live terminal process.
Possible Error Effect:	Possible exhaustion of OSPS operator position resources.
Corrective Action Taken:	The rIPTRDB tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTRDB tuple.
<i>Bad Data =</i>	The process number that was marked in the rIPTRDB tuple (rIPTRDB.ptqrl.st_ownk - DBSFPCBLA).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTRDB tuple (rIPTRDB.procnum).
<i>Dump =</i>	The contents of the rIPTRDB tuple.

PTSB-A**Software Release:** 5E10 only**1. PERIPHERAL INTERFACE DATA BUS TIME SLOT BLOCK (PTSB) AUDIT**

The integrated services line unit (ISLU) is connected to either the packet switch unit (PSU) or the time slot interchange unit (TSIU) by peripheral interface data buses (PIDBs). Similarly, the PSU is connected to either the ISLU or TSIU by PIDBs. The remote integrated services line unit (RISLU) is connected to the TSIU by PIDBs. Each PIDB contains 32 time slots. Each of these time slots is represented in software by the PIDB time slot data block relation (RLPTSB). The PTSB audit ensures the integrity of the RLPTSB relation.

The PTSB audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and non-segmented modes.

2. DATA STRUCTURES

The audit depends on the following relations: the dynamic PIDB time slot data block relation (RLPTSB), the static circuit data relation (RLCKTDATA), the static digital subscriber line (DSL) equipage relation (RLDSLEQUIP), and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these relations.

Other data structures linked to rIPTSB tuple include the channel common block common area relation (RLCCBCOM), the channel data block relation (RLCHDB), and the D-channel port block relation (RLDPB). Lastly, a one-to-one linkage exists between monitored and cut-through rIPTSB tuples for remote digital test access (RDTA) path connections.

3. AUDIT CHECKS

This audit verifies the time slot use against the PTSB CD use and the CD configuration. The audit invokes the static driver (DRSTATIC) to step through the tuples of RLPTSB relation. It performs the following checks.

- (a) The audit performs a dynamic-to-static existence check between rIPTSB and rLCKTDATA. If rLCKTDATA does not have a corresponding tuple or if the equipment status is unequipped, the audit prints an error message and deletes the rIPTSB tuple.
- (b) The audit performs point-to/point-back checks on the linkages to rLCHDB, rLCCBCOM and rLCQB.
- (c) The audit invokes the semantic check function if it detects no linkage errors. The audit determines how the rIPTSB tuple is used.
- (d) The audit checks the in-service/out-of-service (IS/OOS) status of the associated PIDB in rLSMEST.
 - (1) If rLSMEST says the PIDB is OOS, the rIPTSB tuple should be on the OOS list.
 - (2) If rLSMEST says the PIDB is IS, the rIPTSB tuple should be on the idle list or busy.
- (e) If the rIPTSB tuple is associated with a directly connected PIDB (PCDCON) or an ISLU cut-through PICB (PCPISCUT), the audit performs the following checks.
 - (1) If all D-channel state (dstate) fields are idle, the tuple should be linked onto the idle list of the Channel Control Block Common Area Relation (rLCCBCOM). All other linkages should be null.
 - (2) If a D-channel is busy, the audit checks each of the four quarter time slots. The D-channel (dchan) field of idle D-channels must be 0xff. If the dstate field is busy, point-to point-back checks are made to rLDPB. It performs a range check on the dchan field. The audit accesses rLDSLEQUIP and compares the dsl_group and member fields to the corresponding fields in rIPTSB. If they do not match, the audit

sets the dchan and dstate fields to idle.

- (f) If the rIPTSB tuple is associated with a simplex B-channel (PCPSIMPB) or a duplex B-channel (PCPDUPB), the audit makes these checks.
 - (1) If the rIPTSB tuple is linked to the idle list, its other linkages should be null.
 - (2) If rIPTSB tuple is not linked onto the idle list head cell, the audit checks the ranges of its linkages to rICHDB and rCCBCOM.
- (g) For on demand B-channel packet services (ODB), the audit calculates the logical PH and channel number from the rIPTSB port name and the parity in use in the associated D-channel rIDPB tuple in order to verify that there is a corresponding rIODPSOP static tuple. Then the PH and channel number are verified with static data (rIODPSOP).
- (h) The audit verifies the CD use in rIPTSB based on the CD configuration.

For outgoing remote digital test access (PCORDTA), the audit verifies that a one-to-one linkage exists between monitored and cut-through rIPTSB tuples.

The current and new error detection and recovery schemes performed by AUdflnkchk() of the PTSB audit are:

Table 1 Current and New Error Detection and Recovery Schemes

rIPTSB to PCdfnlink PTPB ^a Check	rIPTSB to PCdfnblink PTPB ^a Check	Error Found ^b	Recovery Action
success	success	no	
success	failure	yes	rIPTSB to idle list
failure	failure	yes	rIPTSB to idle list
failure	not done	yes	rIPTSB to idle list
Notes: a. point-to-point-back b. Audit reports an error and schedules DFMAP and DFNIBBLE audits when an error is found.			

A PTPB check will be put into this function for rIPTSB and PCdfnblink[][]. The shelf, pdb_ts, and pdb_nb will be abstracted from rIPTSB and used to index into the PCdfnblink[][] array. This will provide a rIPTSB physical key which can be compared with the key found in the rIPTSB that is passed as a parameter to the AUdflnkchk() function. If the PTPB check fails, the AUdflnkchk() function will return AUFALSE to the calling function AUDchchk() which directs the audit to recover this rIPTSB tuple.

4. ERROR HANDLING

Errors detected with a D-channels are confined to that D-channel. Generally, erroneous rIPTSB tuples are linked to the idle list. The process stored in the process control block linkage area relation (RLPCBLA) may be purged for B-channels. The RLPTSB relation is recovered and linked to the idle list. For ordered B-channel calls, if there is no related rIODPSOP tuple or PH and channel numbers, the call is invalid. The PTSB tuple is idled, and all related call data is recovered.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 2 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	Point-to point-back checks between the rIPTSB tuple (pfdb.bptsb.bchan) associated with a B-channel circuit switch call and the rCCBCOM tuple (port) failed.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The port field (port) from the rCCBCOM tuple.
<i>Good Data =</i>	The B-channel port field (pfdb.bptsb.bchan) from the rIPTSB tuple.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump 1 =</i>	The rCCBCOM tuple.
<i>Dump 2 =</i>	The rIPTSB tuple found in error.

Table 3 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	The linkage field (ptccb.st_ownk) from the rIPTSB tuple under audit to the rCCBCOM tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 4 Error Code: CDCONFIG

Error Code:	CDCONFIG
Error Description:	The CD use field in the PTSB was found to be incorrect based on the actual CD configuration for this ODB call.
Possible Error Effect:	Call processing cannot determine which CD is being used, which could lead to escalating asserts.
Corrective Action Taken:	Valid values are: Active/Standby Configuration = The rIPTSB cduse field will be recovered. Load Shared Configuration = The rIPTSB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rIPTSB tuple.
<i>Bad Data =</i>	The CD stored in the PTSB.
<i>Good Data =</i>	The CD configuration.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The pcPATHCONFIG global array entry for this line unit.

Table 5 Error Code: CHAN_ZERO

Error Code:	CHAN_ZERO
Error Description:	One of the quarter time slots associated with this rIPTSB tuple is in an idle state but the member field is not in an idle state (0xFF).
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the member field from the rIPTSB tuple found in error.
<i>Good Data =</i>	0xFF.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 6 Error Code: CHDB_LINK

Error Code:	CHDB_LINK
Error Description:	The linkage field (pt_ch.link) from the rIPTSB tuple under audit to the rCHDB tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 7 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	Could not read the rCKTDATA tuple for the PIDB circuit associated with this rIPTSB tuple during recovery of the rIPTSB tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 8 Error Code: CQB_LINK

Error Code:	CQB_LINK
Error Description:	The linkage field (ptcqo.st_headk or ptcqo.st_tailk) from the rIPTSB tuple under

	audit to the rICQB tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 9 Error Code: DATA_CON_ERR

Error Code:	DATA_CON_ERR
Error Description:	This error occurs for one of two possible cases: (1) For a directly connected PIDB timeslot, the rLUNIT_CKT static tuple exists for the given rIPTSB dynamic tuple being audited. However a corresponding rPSIUPIDB static tuple could not be accessed (where rLUNIT_CKT.cktname = rPSIUPIDB.psiu_shelf). (2) For a "cut-thru" PIDB time slot, the rPIDB static tuple exists for the given rIPTSB dynamic tuple being audited. However, a corresponding rPICB static tuple could not be accessed (where rPICB.picb = rPIDB.(dsgcpicb & \ 1)). Both of these cases constitute a static data inconsistency which will cause this error to fire until the data is corrected by office personnel.
Possible Error Effect:	Possible asserts, call failures, and lost RLPTSB resources.
Corrective Action Taken:	None.
Manual Action Required:	There are two possible actions: (1) For the case where a rPSIUPIDB tuple is missing, insert the missing tuple into the database by setting the tuple key (rPSIUPIDB.psiu_shelf) to the cktname field in RLUNIT_CKT. For this error case, this value will be dumped as the logical key. (2) For the case where the rPICB tuple is missing, insert the missing tuple into the database by setting the tuple key (rPICB.picb) to the expected key value calculated by the audit. For this error case, this value will be dumped as the logical key.
Dump Description:	
<i>Error Address =</i>	The relation ID of the missing tuple (RLPSIUPIDB, or RLPICB).
<i>Bad Data =</i>	For a missing RLPICB tuple, the expected key value calculated by the audit for rPICB.picb. Otherwise, NA.
<i>Good Data =</i>	The logical key of the rIPTSB tuple under audit.
<i>Logical Index =</i>	For a missing RLPICB tuple, the expected key value calculated by the audit for rPICB.picb, otherwise, the key of RLUNIT_CKT (rLUNIT_CKT.ckt_name).
<i>Dump =</i>	NA

Table 10 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The rIPTSB tuple and the data fanout link (PCdfllink[]) array or the data fanout

	nibble link (PCdfnblink[]) array failed point-to point-back checks.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The logical key field (ptsb) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The protocol handler shelf << 24 The relative protocol handler << 16 The protocol handler data bus time slot << 4 The protocol handler data bus nibble.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The data fanout link array (PCdfnlink[]) entry.
<i>Dump 3 =</i>	The data fanout nibble link array (PCdfnblink[]) entry.

Table 11 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	One of the quarter time slots associated with this rIPTSB tuple is in a busy state but the point-to point-back check between the rIPTSB tuple and the rIDPB tuple failed.
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The rIDPB tuple.

Table 12 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The logical PH and channel number stored in the PTSB could not be successfully verified with the PH and channel number stored in static data (rIODPSOP) for this ODB call. There are 3 possible reasons for this error: (1) There was either an incorrect party bit set, or no party bit set at all, in the D-channel DPB, which corresponds to this ODB call. (2) An rIODPSOP static tuple could not be found for the D-channel port (key of the D-channel DPB) and party bit set in the party flags register which were obtained from the D-channel DPB tuple. (3) The logical PH and channel number stored in static data (rIODPSOP) did not match the ones found in the rIPTSB tuple.

Possible Error Effect:	A user might be using the ODB feature without being a subscriber to this feature.
Corrective Action Taken:	The PTSB will be recovered to an idle state, and the DPB audit will be scheduled as a related audit to clean up the B-channel information in the B-channel rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The party found the D-channel DPB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The B-channel DPB with key corresponding to the PTSB portname.
<i>Dump 3 =</i>	The D-channel DPB with key calculated from the B-port.
NOTE: If the read of rIODPSOP was successful,	the following data will also be dumped:
<i>Dump 4 =</i>	The rIODPSOP static tuple obtained from the D-channel port and party.

Table 13 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	An attempt was made to read the rEQUIPDSL tuple for either a quarter or non-ODB full time slot based on the logical PH stored in the rIPTSB, but the rEQUIPDSL tuple was found not to exist.
Possible Error Effect:	The data in the rIPTSB tuple is either corrupted or invalid, and it cannot be determined what equipage is associated with the tuple.
Corrective Action Taken:	Valid values are: Full Time Slots = The tuple and recovered to an idle state. Non Full Time Slots = The quarter time slot information will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The logical PH number found in the rIPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump =</i>	The rIPTSB tuple.

Table 14 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The dynamic rIPTSB tuple is in the database but the corresponding static rCKTDATA tuple does not exist in the database.
Possible Error Effect:	Call processing may erroneously try to use the rIPTSB tuple.
Corrective Action Taken:	The extra rIPTSB tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 15 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	All quarter time slots are in an idle state but the rIPTSB tuple is not linked onto the out-of-service or idle in-service list.
Possible Error Effect:	The rIPTSB tuple is lost to the system causing ISDN call failures.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 16 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	The rIPTSB is in an idle state but it is not linked to either the idle in-service list or the out-of-service list. The list that the rIPTSB tuple should be linked to is dependent on the state of the PIDB circuit. The state of the PIDB circuit is determined by information in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service (bas_stat = SMOOS) state or in the growth state (qual_1 = SMGR) then the rIPTSB tuple should be linked to the out-of-service list. Otherwise, the rIPTSB tuple should be linked to the idle in-service list.
Possible Error Effect:	If the rSMEST tuple indicates that the rIPTSB tuple is to be linked to the idle in-service list and the rIPTSB tuple is not, the rIPTSB tuple is lost to the system. This will cause ISDN customers to be without service. If the rSMEST tuple indicates that the rIPTSB tuple is to be linked to the out-of-service list and the rIPTSB tuple is not, could cause the system to take peripheral interrupts.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data</i> =	The correct list the rIPTSB tuple should be linked on.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump 1</i> =	The rIPTSB tuple found in error.
<i>Dump 2</i> =	The rSMEST tuple for the associated PIDB circuit.
<i>Dump 3</i> =	The rSMEST tuple for the associated PIDB circuit on the other service group (optional).

Table 17 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	One of the quarter time slots associated with this rIPTSB tuple has an invalid D-channel state (dstate).
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	If the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used as a full time slot the rIPTSB tuple will be reinitialized and linked to the idle in-service list. Otherwise, if the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used on a quarter time slot basis, the quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The state field (use) from the rIPTSB tuple found in error.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 18 Error Code: MISSING

Error Code:	MISSING
Error Description:	The audit found a missing tuple in a static relation which is being used to provide existence checks on the rIPTSB tuple under audit. Specific information on which relation's tuple could not be accessed is provided in the <i>Dump Description</i> .
Possible Error Effect:	Possible asserts, call failures, and lost RLPTSB resources.
Corrective Action Taken:	The rIPTSB tuple under audit will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The relation ID of the static relation that the audit tried to access.
<i>Bad Data</i> =	Varies according to the relation being accessed. Table 19 provides the relationship between the relation which the audit tried to access, and the <i>Bad Data</i> that is dumped.
<i>Good Data</i> =	The key of the rIPTSB tuple.
<i>Logical Index</i> =	Varies according to the relation being accessed. Table 20 provides the relationship between the relation which the audit tried to access, and the <i>Logical Key</i> that is dumped.
<i>Dump</i> =	For the case where a rCKTDATA tuple could not be accessed, the associated rPSIUPIDB static tuple is dumped (rPSIUPIDB.dpidb[PIDB number]). For all other cases, no data is dumped.

Table 19 Bad Data Dumped

Relation ID	Bad Data Dumped
RLCKTDDATA	PSU Connect Number
RLPIDB	NA
RLUNIT_CKT	Unit Type (SMPSIU)
RLUNITDATA	rUNITDATA.utype and rUNITDATA.unit

Table 20 Logical Key Dumped

Relation ID	Logical Key Dumped
RLCKTDDATA	rCKTDATA.name
RLPIDB	The PIDB number
RLUNIT_CKT	rUNIT_CKT.ckt_idx
RLUNITDATA	rUNITDATA.ckt_idx

Table 21 Error Code: KEY

Error Code:	KEY
Error Description:	The key field (ptsb) of the rIPTSB tuple has been corrupted.
Possible Error Effect:	This rIPTSB tuple will be unusable by the switch and may cause asserts from peripheral control. If this time slot is associated with a D-channel or packet B-channel these channels will remain out-of-service.
Corrective Action Taken:	The key field (ptsb) of the rIPTSB tuple will be corrected to the appropriate value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The key field (ptsb) from the rIPTSB tuple.
<i>Good Data</i> =	The correct value of the key field (ptsb)
<i>Logical Index</i> =	The correct value of the key field (ptsb).
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 22 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One of the time slots associated with this rIPTSB tuple is in a busy state but the

	point-to point-back check between the rIPTSB tuple and the rIDPB tuple failed.
Possible Error Effect:	The rIPTSB tuple is lost to the system causing ISDN or Packet Pipe call failures.
Corrective Action Taken:	The PTSB will be recovered to an idle state, and the DPB audit will be scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 23 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static rIDSLEQUIP tuple associated with the rIPTSB tuple under audit was found to be missing in the database. This could be due to bad data in the rIPTSB tuple or ODD inconsistencies between rIDSLEQUIP and rIEQUIPDSL.
Possible Error Effect:	If this error is due to ODD inconsistencies between rIDSLEQUIP and rIEQUIPDSL the port will continue to be setup and torn down over and over by peripheral control and audits.
Corrective Action Taken:	If the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used as a full time slot the rIPTSB tuple will be reinitialized and linked to the idle in-service list. Otherwise, if the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used on a quarter time slot basis, the quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	If this error rolls out over and over on the same rIPTSB tuple then there are ODD inconsistencies between the RIDSLEQUIP and RLEQUIPDSL relations. The port, DSL group, member, and time slot information can be obtained from the dump of the rIPTSB tuple. The port information can be used to access the rIDSLEQUIP tuple and the DSL group and member can be used to access the rIEQUIPDSL tuple. These tuples should then point to each other. If not these tuples should be corrected using ODBE.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 24 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Found a linkage error when trying to traverse from the rIPTSB tuple to the owning rICBCOM tuple.
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The return code from the DA primitive that was trying to traverse the linkage

	from the rIPTSB tuple to the rCCBCOM tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 25 Error Code: PIDB

Error Code:	PIDB
Error Description:	Point-to point-back checks failed between the dynamic rIPTSB tuple and the static rIDSLEQUIP tuple. The directly connected peripheral interface data bus field (dpidb), the time slot field (timeslot), or the quarter time slot field (qts) failed the point-to point-back check.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not affect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple under audit.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump 1</i> =	The rIPTSB tuple found in error.
<i>Dump 2</i> =	The rIDSLEQUIP tuple.

Table 26 Error Code: PORT

Error Code:	PORT
Error Description:	The B-channel port stored in the PTSB did not match the port obtained from rEQUIPDSL for a non-ODB full time slot case.
Possible Error Effect:	The inconsistency may cause call processing applications to fire asserts.
Corrective Action Taken:	The rIPTSB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The rIPTSB tuple.
<i>Bad Data</i> =	The B-channel port found in the PTSB.
<i>Good Data</i> =	The port found in rEQUIPDSL.
<i>Logical Index</i> =	The key of the rIPTSB tuple.
<i>Dump 1</i> =	The rIPTSB tuple.
<i>Dump 2</i> =	The rEQUIPDSL tuple.

Table 27 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The one-to-one linkage between monitored and cut-through rIPTSB tuples (rIPTSB.pt_ch.lnk) does not exist for a remote protocol access PTSB use of outgoing remote digital test access (PCORDTA). A relationship exists such that if the monitored rIPTSB.use field is marked PCORDTA (outgoing remote digital test access), the tuple's one-to-one linkage field rIPTSB.pt_ch.lnk should be non-zero, and should contain the key of the corresponding cut-through rIPTSB tuple to which it should be linked.
Possible Error Effect:	Remote protocol test access data will be in an inconsistent state, and as a result, test actions and operations may be impaired or fail.
Corrective Action Taken:	The monitored rIPTSB tuple will be recovered to an idle state, and the corresponding linkage in the cut-through rIPTSB tuple will be idled out if one exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the monitored rIPTSB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the monitored rIPTSB tuple.
<i>Dump</i> =	The monitored rIPTSB tuple.

Table 28 Error Code: READ

Error Code:	READ
Error Description:	The static rPICB tuple associated with the rIPTSB tuple under audit was not found in the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None, the code will assume that the rIPTSB tuple is associated with a ISLU instead of a RISLU.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the buffer containing the contents of the rPICB tuple.
<i>Bad Data =</i>	The key field (picb) of the rPICB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (picb) of the rPICB tuple.
<i>Dump =</i>	NA.

Table 29 Error Code: RELPIDB

Error Code:	RELPIDB
Error Description:	The rIPTSB tuple and the integrated services line unit local interface data bus time slot array (PCislidbts[]) failed point-to-point-back checks.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The relative PIDB field (relpidbts) from the PCislidbts[] array.
<i>Good Data =</i>	The relative PIDB field (rel_pidb) from the rIPTSB tuple.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 30 Error Code: RELPTR

Error Code:	RELPTR
Error Description:	For an ODB call, an attempt was made to obtain a corresponding D-channel DPB from the key of the B-channel DPB stored on the PTSB, but either the PTSB portname has become corrupted or the D-channel DPB does not exist.
Possible Error Effect:	The on demand packet service order profile data (logical PH and channel number) cannot be accessed, and there is no D-channel DPB available.
Corrective Action Taken:	The PTSB tuple will be recovered to an idle state, and the DPB audit will be scheduled as a related audit in case the D-channel DPB is missing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The D-channel port (key) of the DPB which was obtained.
<i>Good Data =</i>	The D-channel port calculated from the B-channel port found in the rIPTSB tuple.
<i>Logical Index =</i>	The key of the PTSB.
<i>Dump =</i>	The rIPTSB tuple.

Table 31 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple associated with the rIPTSB tuple under audit is not in the database.
Possible Error Effect:	Call processing and maintenance activities on the time slot associated with this rIPTSB tuple may cause asserts.
Corrective Action Taken:	The CKTDATA audit will be scheduled as a related audit to insert the rSMEST

	tuple back into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 32 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rIPTSB tuple state field (use) indicates that this tuple is being used as a full time slot, however, the D-channel state (dstate) is in an invalid state.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The D-channel state field (dstate) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 33 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The state field (use) of the rIPTSB tuple is in an invalid state.
Possible Error Effect:	This rIPTSB tuple may be lost to the system. It can also cause asserts by the application program because of the invalid state.
Corrective Action Taken:	The rIPTSB tuple reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The state field (use) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 34 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rIPTSB tuple was found stuck in a transient state for longer than 5 minutes when trying to setup or tear down one of the quarter time slots.
Possible Error Effect:	The quarter time slot that was stuck in a setup or tear down state is lost to the system. If this time slot was a directly connected time slot the D-channel or packet B-channel associated with this quarter time slot would not be able to be established.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The transient bit flag (autime) from the rIPTSB tuple.
<i>Good Data</i> =	DBNO.
<i>Logical Index</i> =	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump</i> =	The rIPTSB tuple under audit.

Table 35 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	The time slot usage field (ts_use) in the rIPTSB tuple does not agree with the way the time slot is being used as specified by the state field (use).
Possible Error Effect:	Maintenance activities on this time slot can result in asserts and single process purges.
Corrective Action Taken:	The time slot use field (tsuse) in the rIPTSB tuple will be updated to be consistent with the use of the rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	Actual value of the tsuse field.
<i>Good Data</i> =	Valid value for tsuse field.
<i>Logical Index</i> =	Logical key of the rIPTSB tuple.
<i>Dump</i> =	Dump of the rIPTSB tuple.

Table 36 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PTSB audit does not recognize the peripheral interface data bus (PIDB) type specified by the PIDB type (pidbtype) field in the associated rICKTDATA tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PTSB audit must be modified to include the new PIDB type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple under audit.
<i>Bad Data</i> =	PIDB type field (pidbtype) from the rICKTDATA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) from the rIPTSB tuple under audit.
<i>Dump 1</i> =	The rIPTSB tuple found in error.
<i>Dump 2</i> =	The associated rICKTDATA tuple.

Table 37 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The rIPTSB tuple state field (use) is marked as unavailable but the rIPTSB tuple is not linked onto the unavailable list (DBPTSBNA).
Possible Error Effect:	This rIPTSB tuple may be erroneously allocated to a call.
Corrective Action Taken:	The rIPTSB tuple will be reinitialized and linked to the unavailable list (DBPTSBNA).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple in error.
<i>Bad Data</i> =	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data</i> =	The value of the unavailable list (DBPTSBNA).
<i>Logical Index</i> =	The logical key field (ptsb) from the rIPTSB tuple under audit.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 38 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rSMEST tuple associated with this rIPTSB tuple is in an out-of-service state, however, the rIPTSB tuple is not linked onto the out-of-service list.
Possible Error Effect:	If the rIPTSB tuple remains on an in-service list when the PIDB circuit is out-of-service may cause peripheral interrupts.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked onto the out-of-service list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data =</i>	The out-of-service list value (DBPTSBOOS).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 39 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIPTSB tuple is in an idle state (use = PCPNUL) and is linked to the idle in-service list or the out-of-service list, however, it is still linked to other resources. These resources may be either a rICHDB or rICQB tuples. The rIPTSB tuple was not properly reinitialized before it was placed on the idle in-service list or out-of-service list.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the linkage to be non-null.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

PTSB-B

Software Release: 5E11 - 5E13

1. PERIPHERAL INTERFACE DATA BUS TIME SLOT BLOCK (PTSB) AUDIT

The integrated services line unit (ISLU) is connected to either the packet switch unit (PSU) or the time slot interchange unit (TSIU) by peripheral interface data buses (PIDBs). Similarly, the PSU is connected to either the ISLU or TSIU by PIDBs. The remote integrated services line unit (RISLU) is connected to the TSIU by PIDBs. Each PIDB contains 32 time slots. Each of these time slots is represented in software by the PIDB time slot data block relation (RLPTSB). The PTSB audit ensures the integrity of the RLPTSB relation.

The PTSB audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-M) environment in the segmented and non-segmented modes.

2. DATA STRUCTURES

The audit depends on the following relations: the dynamic PIDB time slot data block relation (RLPTSB), the static circuit data relation (RLCKTDATA), the static digital subscriber line (DSL) equipage relation (RLDSLEQUIP), and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these relations. Other data structures linked to rIPTSB tuple include the channel common block common area relation (RLCCBCOM), the channel data block relation (RLCHDB), and the D-channel port block relation (RLDPB). Lastly, a one-to-one linkage exists between monitored and cut-through rIPTSB tuples for remote digital test access (RDTA) path connections.

3. AUDIT CHECKS

This audit verifies the time slot use against the PTSB CD use and the CD configuration. The audit invokes the static driver (DRSTATIC) to step through the tuples of RLPTSB relation. It performs the following checks.

- (a) The audit performs a dynamic-to-static existence check between rIPTSB and rLCKTDATA. If rLCKTDATA does not have a corresponding tuple or if the equipment status is unequipped, the audit prints an error message and deletes the rIPTSB tuple.
- (b) The audit performs point-to/point-back checks on the linkages to rLCHDB, rLCCBCOM and rLCQB.
- (c) The audit invokes the semantic check function if it detects no linkage errors. The audit determines how the rIPTSB tuple is used.
- (d) The audit checks the in-service/out-of-service (IS/OOS) status of the associated PIDB in rLSMEST.
 - (1) If rLSMEST says the PIDB is OOS, the rIPTSB tuple should be on the OOS list.
 - (2) If rLSMEST says the PIDB is IS, the rIPTSB tuple should be on the idle list or busy.
- (e) If the rIPTSB tuple is associated with a directly connected PIDB (PCDCON) or an ISLU cut-through PICB (PCPISCUT), the audit performs the following checks.
 - (1) If all D-channel state (dstate) fields are idle, the tuple should be linked onto the idle list of the channel control block common area relation (rLCCBCOM). All other linkages should be null.
 - (2) If a D-channel is busy, the audit checks each of the four quarter time slots. The D-channel (dchan) field of idle D-channels must be 0xff. If the dstate field is busy, point-to point-back checks are made to rLDPB. It performs a range check on the dchan field. The audit accesses rLDSLEQUIP and compares the dsl_group and member fields to the corresponding fields in rIPTSB. If they do not match, the audit sets the dchan and dstate fields to idle.

- (f) If the rIPTSB tuple is associated with a simplex B-channel (PCPSIMPB) or a duplex B-channel (PCPDUPB), the audit makes these checks.
- (1) If the rIPTSB tuple is linked to the idle list, its other linkages should be null.
 - (2) If rIPTSB tuple is not linked onto the idle list head cell, the audit checks the ranges of its linkages to rICHDB and rCCBCOM.
- (g) For on demand B-channel (ODB) packet services, the audit calculates the logical PH and channel number from the rIPTSB port name and the parity in use in the associated D-channel rIDPB tuple in order to verify that there is a corresponding rIODPSOP static tuple. Then the PH and channel number are verified with static data (rIODPSOP).
- (h) The audit verifies the CD use in rIPTSB based on the CD configuration.

For outgoing remote digital test access (PCORDTA), the audit verifies that a one-to-one linkage exists between monitored and cut-through rIPTSB tuples.

The current and new error detection and recovery schemes performed by AUdflnkchk() of the PTSB audit are:

Table 1 Current and New Error Detection and Recovery Schemes

rIPTSB to PCdfnlink PTPB ^a Check	rIPTSB to PCdfnblink PTPB ^a Check	Error Found ^b	Recovery Action
success	success	no	
success	failure	yes	rIPTSB to idle list
failure	failure	yes	rIPTSB to idle list
failure	not done	yes	rIPTSB to idle list
Notes: a. point-to-point-back b. Audit reports an error and schedules DFMAP and DFNIBBLE audits when an error is found.			

A PTPB check will be put into this function for rIPTSB and PCdfnblink[][]. The shelf, pdb_ts, and pdb_nb will be abstracted from rIPTSB and used to index into the PCdfnblink[][] array. This will provide a rIPTSB physical key which can be compared with the key found in the rIPTSB that is passed as a parameter to the AUdflnkchk() function. If the PTPB check fails, the AUdflnkchk() function will return AUFALSE to the calling function AUdchchk() which directs the audit to recover this rIPTSB tuple.

4. ERROR HANDLING

Errors detected with a D-channels are confined to that D-channel. Generally, erroneous rIPTSB tuples are linked to the idle list. The process stored in the process control block linkage area relation (RLPCBLA) may be purged for B-channels. The RLPTSB relation is recovered and linked to the idle list. For ordered B-channel calls, if there is no related rIODPSOP tuple or PH and channel numbers, the call is invalid. The PTSB tuple is idled, and all related call data is recovered.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 2 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT

Error Description:	Point-to point-back checks between the rIPTSB tuple (pfdb.bptsb.bchan) associated with a B-channel circuit switch call and the rCCBCOM tuple (port) failed.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The port field (port) from the rCCBCOM tuple.
<i>Good Data =</i>	The B-channel port field (pfdb.bptsb.bchan) from the rIPTSB tuple.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump 1 =</i>	The rCCBCOM tuple.
<i>Dump 2 =</i>	The rIPTSB tuple found in error.

Table 3 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	The linkage field (ptccb.st_ownk) from the rIPTSB tuple under audit to the rCCBCOM tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 4 Error Code: CDCONFIG

Error Code:	CDCONFIG
Error Description:	The CD use field in the PTSB was found to be incorrect based on the actual CD configuration for this ODB call.
Possible Error Effect:	Call processing cannot determine which CD is being used, which could lead to escalating asserts.
Corrective Action Taken:	One of the following: Active/standby configuration: The rIPTSB cduse field will be recovered. Load shared configuration: The rIPTSB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rIPTSB tuple.
<i>Bad Data =</i>	The CD stored in the PTSB.
<i>Good Data =</i>	The CD configuration.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The pcPATHCONFIG global array entry for this line unit.

Table 5 Error Code: CHAN_ZERO

Error Code:	CHAN_ZERO
Error Description:	One of the quarter time slots associated with this rIPTSB tuple is in an idle state but the member field is not in an idle state (0xff).
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the member field from the rIPTSB tuple found in error.
<i>Good Data =</i>	0xFF.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 6 Error Code: CHDB_LINK

Error Code:	CHDB_LINK
Error Description:	The linkage field (pt_ch.link) from the rIPTSB tuple under audit to the rCHDB tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 7 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	Could not read the rCKTDATA tuple for the PIDB circuit associated with this rIPTSB tuple during recovery of the rIPTSB tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 8 Error Code: CQB_LINK

Error Code:	CQB_LINK
Error Description:	The linkage field (ptcq.st_headk or ptcq.st_tailk) from the rIPTSB tuple under audit to the rCQB tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.

Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 9 Error Code: DATA_CON_ERR

Error Code:	DATA_CON_ERR
Error Description:	This error occurs for one of two possible cases: (1) For a directly connected PIDB timeslot, the rUNIT_CKT static tuple exists for the given rIPTSB dynamic tuple being audited. However a corresponding rPSIUPIDB static tuple could not be accessed (where rUNIT_CKT.cktname = rPSIUPIDB.psiu_shelf) or, (2) For a "cut-thru" PIDB time slot, the rPIDB static tuple exists for the given rIPTSB dynamic tuple being audited. However, a corresponding rPICB static tuple could not be accessed (where rPICB.picb = rPIDB.(dsgcpicb & \ 1)). Both of these cases constitute a static data inconsistency which will cause this error to fire until the data is corrected manually.
Possible Error Effect:	Possible asserts, call failures, and lost RLPTSB resources.
Corrective Action Taken:	None.
Manual Action Required:	Corresponds to Error Description as follows: (1) For the case where a rPSIUPIDB tuple is missing, insert the missing tuple into the database by setting the tuple key (rPSIUPIDB.psiu_shelf) to the <i>cktname</i> field in RLUNIT_CKT. For this error case, this value will be dumped as the logical key. (2) For the case where the rPICB tuple is missing, insert the missing tuple into the data base by setting the tuple key (rPICB.picb) to the expected key value calculated by the audit. For this error case, this value will be dumped as the logical key.
Dump Description:	
<i>Error Address =</i>	The relation ID of the missing tuple (RLPSIUPIDB, or RLPICB).
<i>Bad Data =</i>	For a missing RLPICB tuple, the expected key value calculated by the audit for rPICB.picb. Otherwise, NA.
<i>Good Data =</i>	The logical key of the rIPTSB tuple under audit.
<i>Logical Index =</i>	For a missing RLPICB tuple, the expected key value calculated by the audit for rPICB.picb, otherwise, the key of RLUNIT_CKT (rUNIT_CKT.ckt_name).
<i>Dump =</i>	NA.

Table 10 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The rIPTSB tuple and the data fanout link (PCdflink[]) array or the data fanout nibble link (PCdfnblink[]) array failed point-to-point-back checks.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The logical key field (ptsb) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The protocol handler shelf << 24 The relative protocol handler << 16 The protocol handler data bus time slot << 4 The protocol handler data bus nibble.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The data fanout link array (PCdflink[]) entry.
<i>Dump 3 =</i>	The data fanout nibble link array (PCdfnblink[]) entry.

Table 11 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	One of the quarter time slots associated with this rIPTSB tuple is in a busy state but the point-to point-back check between the rIPTSB tuple and the rIDPB tuple failed.
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The rIDPB tuple.

Table 12 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The logical PH and channel number stored in the PTSB could not be successfully verified with the PH and channel number stored in static data (rIODPSOP) for this ODB call. There are 3 possible reasons for this error: (1) There was either an incorrect party bit set, or no party bit set at all, in the D-channel DPB, which corresponds to this ODB call. (2) An rIODPSOP static tuple could not be found for the D-channel port (key of the D-channel DPB) and party bit set in the party flags register which were obtained from the D-channel DPB tuple. (3) The logical PH and channel number stored in static data (rIODPSOP) did not match the ones found in the rIPTSB tuple.
Possible Error Effect:	A user might be using the ODB feature without being a subscriber to this

	feature.
Corrective Action Taken:	The PTSB will be recovered to an idle state, and the DPB audit will be scheduled as a related audit to clean up the B-channel information in the B-channel rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The party found the D-channel DPB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The B-channel DPB with key corresponding to the PTSB portname.
<i>Dump 3 =</i>	The D-channel DPB with key calculated from the B-port.
NOTE: If the read of rIODPSOP was successful,	the following data will also be dumped:
<i>Dump 4 =</i>	The rIODPSOP static tuple obtained from the D-channel port and party.

Table 13 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	An attempt was made to read the rEQUIPDSL tuple for either a quarter or non-ODB full time slot based on the logical PH stored in the rIPTSB, but the rEQUIPDSL tuple was found not to exist.
Possible Error Effect:	The data in the rIPTSB tuple is either corrupted or invalid, and it cannot be determined what equipage is associated with the tuple.
Corrective Action Taken:	Corrective action taken as follows: Full Time Slots: The tuple and recovered to an idle state. Non-Full Time Slots: The quarter time slot information will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The logical PH number found in the rIPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump =</i>	The rIPTSB tuple.

Table 14 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The dynamic rIPTSB tuple is in the data base but the corresponding static rCKTDATA tuple does not exist in the data base.
Possible Error Effect:	Call processing may erroneously try to use the rIPTSB tuple.
Corrective Action Taken:	The extra rIPTSB tuple will be deleted from the data base.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 15 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	All quarter time slots are in an idle state but the rIPTSB tuple is not linked onto the out-of-service or idle in-service list.
Possible Error Effect:	The rIPTSB tuple is lost to the system causing ISDN call failures.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 16 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	The rIPTSB is in an idle state but it is not linked to either the idle in-service list or the out-of-service list. The list that the rIPTSB tuple should be linked to is dependent on the state of the PIDB circuit. The state of the PIDB circuit is determined by information in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service (bas_stat = SMOOS) state or in the growth state (qual_1 = SMGR) then the rIPTSB tuple should be linked to the out-of-service list. Otherwise, the rIPTSB tuple should be linked to the idle in-service list.
Possible Error Effect:	If the rSMEST tuple indicates that the rIPTSB tuple is to be linked to the idle in-service list and the rIPTSB tuple is not, the rIPTSB tuple is lost to the system. This will cause ISDN customers to be without service. If the rSMEST tuple indicates that the rIPTSB tuple is to be linked to the out-of-service list and the rIPTSB tuple is not, could cause the system to take peripheral interrupts.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data</i> =	The correct list the rIPTSB tuple should be linked on.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump 1</i> =	The rIPTSB tuple found in error.
<i>Dump 2</i> =	The rSMEST tuple for the associated PIDB circuit.
<i>Dump 3</i> =	The rSMEST tuple for the associated PIDB circuit on the other service group (optional).

Table 17 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	One of the quarter time slots associated with this rIPTSB tuple has an invalid D-channel state (dstate).
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	If the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used as a full time slot the rIPTSB tuple will be reinitialized and linked to the idle in-service list. Otherwise, if the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used on a quarter time slot basis, the quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The state field (use) from the rIPTSB tuple found in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.

<i>Dump =</i>	The rIPTSB tuple found in error.

Table 18 Error Code: MISSING

Error Code:	MISSING
Error Description:	The audit found a missing tuple in a static relation which is being used to provide existence checks on the rIPTSB tuple under audit. Specific information on which relation's tuple could not be accessed is provided in the <i>Dump Description</i> below.
Possible Error Effect:	Possible asserts, call failures, and lost RLPTSB resources.
Corrective Action Taken:	The rIPTSB tuple under audit will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The relation ID of the static relation that the audit tried to access.
<i>Bad Data =</i>	Varies according to the relation being accessed. The following list provides the relationship between the relation which the audit tried to access, and the <i>Bad Data</i> that is dumped: Relation ID = Bad data dumped. RLCKTTDATA = PSU connect number. RLPIDB = NA. RLUNIT_CKT = Unit type (SMPSIU). RLUNITDATA = rUNITDATA.utype and rUNITDATA.unit.
<i>Good Data =</i>	The key of the rIPTSB tuple.
<i>Logical Index =</i>	Varies according to the relation being accessed. The following list provides the relationship between the relation which the audit tried to access, and the <i>Logical Key</i> that is dumped: Relation ID = Logical key dumped. RLCKTDATA = rICKTDATA.name. RLPIDB = The PIDB number. RLUNIT_CKT = rUNIT_CKT.ckt_idx. RLUNITDATA = rUNITDATA.ckt_idx.
<i>Dump =</i>	For the case where a rICKTDATA tuple could not be accessed, the associated rIPSIUPIDB static tuple is dumped (rIPSIUPIDB.dpidb[PIDB number]). For all other cases, no data is dumped.

Table 19 Error Code: KEY

Error Code:	KEY
Error Description:	The key field (ptsb) of the rIPTSB tuple has been corrupted.
Possible Error Effect:	This rIPTSB tuple will be unusable by the switch and may cause asserts from peripheral control. If this time slot is associated with a D-channel or packet B-channel these channels will remain out-of-service.
Corrective Action Taken:	The key field (ptsb) of the rIPTSB tuple will be corrected to the appropriate value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The key field (ptsb) from the rIPTSB tuple.
<i>Good Data =</i>	The correct value of the key field (ptsb).

<i>Logical Index</i> =	The correct value of the key field (ptsb).
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 20 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One of the time slots associated with this rIPTSB tuple is in a busy state but the point-to point-back check between the rIPTSB tuple and the rIDPB tuple failed.
Possible Error Effect:	The rIPTSB tuple is lost to the system causing ISDN or packet pipe call failures.
Corrective Action Taken:	The PTSB will be recovered to an idle state, and the DPB audit will be scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 21 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static rIDSLEQUIP tuple associated with the rIPTSB tuple under audit was found to be missing in the database. This could be due to bad data in the rIPTSB tuple or ODD inconsistencies between rIDSLEQUIP and rIEQUIPDSL.
Possible Error Effect:	If this error is due to ODD inconsistencies between rIDSLEQUIP and rIEQUIPDSL the port will continue to be setup and torn down over and over by peripheral control and audits.
Corrective Action Taken:	If the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used as a full time slot the rIPTSB tuple will be reinitialized and linked to the idle in-service list. Otherwise, if the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used on a quarter time slot basis, the quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	If this error rolls out over and over on the same rIPTSB tuple then there are ODD inconsistencies between the RLDSLEQUIP and RLEQUIPDSL relations. The port, DSL group, member, and time slot information can be obtained from the dump of the rIPTSB tuple. The port information can be used to access the rIDSLEQUIP tuple and the DSL group and member can be used to access the rIEQUIPDSL tuple. These tuples should then point to each other. If not these tuples should be corrected using ODBE.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 22 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Found a linkage error when trying to traverse from the rIPTSB tuple to the owning rICCB COM tuple.
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat =

	SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The return code from the DA primitive that was trying to traverse the linkage from the rIPTSB tuple to the rCCBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 23 Error Code: PIDB

Error Code:	PIDB
Error Description:	Point-to point-back checks failed between the dynamic rIPTSB tuple and the static rIDSLEQUIP tuple. The directly connected peripheral interface data bus field (dpidb), the time slot field (timeslot), or the quarter time slot field (qts) failed the point-to point-back check.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not affect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The rIDSLEQUIP tuple.

Table 24 Error Code: PORT

Error Code:	PORT
Error Description:	The B-channel port stored in the PTSB did not match the port obtained from rEQUIPDSL for a non-ODB full time slot case.
Possible Error Effect:	The inconsistency may cause call processing applications to fire asserts.
Corrective Action Taken:	The rIPTSB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rIPTSB tuple.
<i>Bad Data =</i>	The B-channel port found in the PTSB.
<i>Good Data =</i>	The port found in rEQUIPDSL.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The rEQUIPDSL tuple.

Table 25 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The one-to-one linkage between monitored and cut-through rIPTSB tuples (rIPTSB.pt_ch.lnk) does not exist for a remote protocol access PTSB use of outgoing remote digital test access (PCORDTA). A relationship exists such that if the monitored rIPTSB.use field is marked PCORDTA (outgoing remote digital test access), the tuple's one-to-one linkage field rIPTSB.pt_ch.lnk should be non-zero, and should contain the key of the corresponding cut-through rIPTSB tuple to which it should be linked.
Possible Error Effect:	Remote protocol test access data will be in an inconsistent state, and as a result, test actions and operations may be impaired or fail.
Corrective Action Taken:	The monitored rIPTSB tuple will be recovered to an idle state, and the corresponding linkage in the cut-through rIPTSB tuple will be idled out if one exists.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the monitored rIPTSB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the monitored rIPTSB tuple.
<i>Dump =</i>	The monitored rIPTSB tuple.

Table 26 Error Code: READ

Error Code:	READ
Error Description:	The static rPICB tuple associated with the rIPTSB tuple under audit was not found in the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None, the code will assume that the rIPTSB tuple is associated with a ISLU instead of a RISLU.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the buffer containing the contents of the rPICB tuple.
<i>Bad Data =</i>	The key field (picb) of the rPICB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (picb) of the rPICB tuple.
<i>Dump =</i>	NA.

Table 27 Error Code: RELPIDB

Error Code:	RELPIDB
Error Description:	The rIPTSB tuple and the integrated services line unit local interface data bus time slot array (PCislidbts[]) failed point-to point-back checks.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The relative PIDB field (relpidbts) from the PCislidbts[] array.
<i>Good Data =</i>	The relative PIDB field (rel_pidb) from the rIPTSB tuple.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 28 Error Code: RELPTR

Error Code:	RELPTR
Error Description:	For an ODB call, an attempt was made to obtain a corresponding D-channel DPB from the key of the B-channel DPB stored on the PTSB, but either the PTSB portname has become corrupted or the D-channel DPB does not exist.
Possible Error Effect:	The on demand packet service order profile data (logical PH and channel number) cannot be accessed, and there is no D-channel DPB available.
Corrective Action Taken:	The PTSB tuple will be recovered to an idle state, and the DPB audit will be scheduled as a related audit in case the D-channel DPB is missing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The D-channel port (key) of the DPB which was obtained.
<i>Good Data =</i>	The D-channel port calculated from the B-channel port found in the rIPTSB tuple.
<i>Logical Index =</i>	The key of the PTSB.
<i>Dump =</i>	The rIPTSB tuple.

Table 29 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rISMEST tuple associated with the rIPTSB tuple under audit is not in the data base.
Possible Error Effect:	Call processing and maintenance activities on the time slot associated with this rIPTSB tuple may cause asserts.
Corrective Action Taken:	The CKTDATA audit will be scheduled as a related audit to insert the rISMEST tuple back into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 30 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rIPTSB tuple state field (use) indicates that this tuple is being used as a full time slot, however, the D-channel state (dstate) is in an invalid state.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The D-channel state field (dstate) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 31 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The state field (use) of the rIPTSB tuple is in an invalid state.
Possible Error Effect:	This rIPTSB tuple may be lost to the system. It can also cause asserts by the application program because of the invalid state.
Corrective Action Taken:	The rIPTSB tuple reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The state field (use) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 32 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rIPTSB tuple was found stuck in a transient state for longer than 5 minutes when trying to setup or tear down one of the quarter time slots.
Possible Error Effect:	The quarter time slot that was stuck in a setup or tear down state is lost to the system. If this time slot was a directly connected time slot the D-channel or

	packet B-channel associated with this quarter time slot would not be able to be established.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The transient bit flag (autime) from the rIPTSB tuple.
<i>Good Data =</i>	DBNO.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 33 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	The time slot usage field (ts_use) in the rIPTSB tuple does not agree with the way the time slot is being used as specified by the state field (use).
Possible Error Effect:	Maintenance activities on this time slot can result in asserts and single process purges.
Corrective Action Taken:	The time slot use field (tsuse) in the rIPTSB tuple will be updated to be consistent with the use of the rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	Actual value of the tsuse field.
<i>Good Data =</i>	Valid value for tsuse field.
<i>Logical Index =</i>	Logical key of the rIPTSB tuple.
<i>Dump =</i>	Dump of the rIPTSB tuple.

Table 34 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PTSB audit does not recognize the peripheral interface data bus (PIDB) type specified by the PIDB type (pidbtype) field in the associated rICKTDATA tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PTSB audit must be modified to include the new PIDB type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	PIDB type field (pidbtype) from the rICKTDATA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple under audit.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The associated rICKTDATA tuple.

Table 35 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The rIPTSB tuple state field (use) is marked as unavailable but the rIPTSB tuple is not linked onto the unavailable list (DBPTSBNA).
Possible Error Effect:	This rIPTSB tuple may be erroneously allocated to a call.
Corrective Action Taken:	The rIPTSB tuple will be reinitialized and linked to the unavailable list (DBPTSBNA).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple in error.
<i>Bad Data =</i>	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data =</i>	The value of the unavailable list (DBPTSBNA).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 36 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rSMEST tuple associated with this rIPTSB tuple is in an out-of-service state, however, the rIPTSB tuple is not linked onto the out-of-service list.
Possible Error Effect:	If the rIPTSB tuple remains on an in-service list when the PIDB circuit is out-of-service may cause peripheral interrupts.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked onto the out-of-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data =</i>	The out-of-service list value (DBPTSBOOS).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 37 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIPTSB tuple is in an idle state (use = PCPNUL) and is linked to the idle in-service list or the out-of-service list, however, it is still linked to other resources. These resources may be either a rICHDB or rICQB tuples. The rIPTSB tuple was not properly reinitialized before it was placed on the idle in-service list or out-of-service list.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the linkage to be non-null.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

PTSB-C

Software Release: 5E14 and later

1. PERIPHERAL INTERFACE DATA BUS TIME SLOT BLOCK (PTSB) AUDIT

The integrated services line unit (ISLU) is connected to either the packet switch unit (PSU) or the time slot interchange unit (TSIU) by peripheral interface data buses (PIDBs). Similarly, the PSU is connected to either the ISLU or TSIU by PIDBs. The remote integrated services line unit (RISLU) is connected to the TSIU by PIDBs. Each PIDB contains 32 time slots. Each of these time slots is represented in software by the PIDB time slot data block relation (RLPTSB). The PTSB audit ensures the integrity of the RLPTSB relation.

The PTSB audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-M) environment in the segmented and non-segmented modes.

2. DATA STRUCTURES

The audit depends on the following relations: the dynamic PIDB time slot data block relation (RLPTSB), the static circuit data relation (RLCKTDATA), the static digital subscriber line (DSL) equipage relation (RLDSLEQUIP), and the switch maintenance equipment status table relation (RLSMEST). Figure 1 shows these relations. Other data structures linked to rIPTSB tuple include the channel common block common area relation (RLCCBCOM), the channel data block relation (RLCHDB), and the D-channel port block relation (RLDPB). Lastly, a one-to-one linkage exists between monitored and cut-through rIPTSB tuples for remote digital test access (RDTA) path connections.

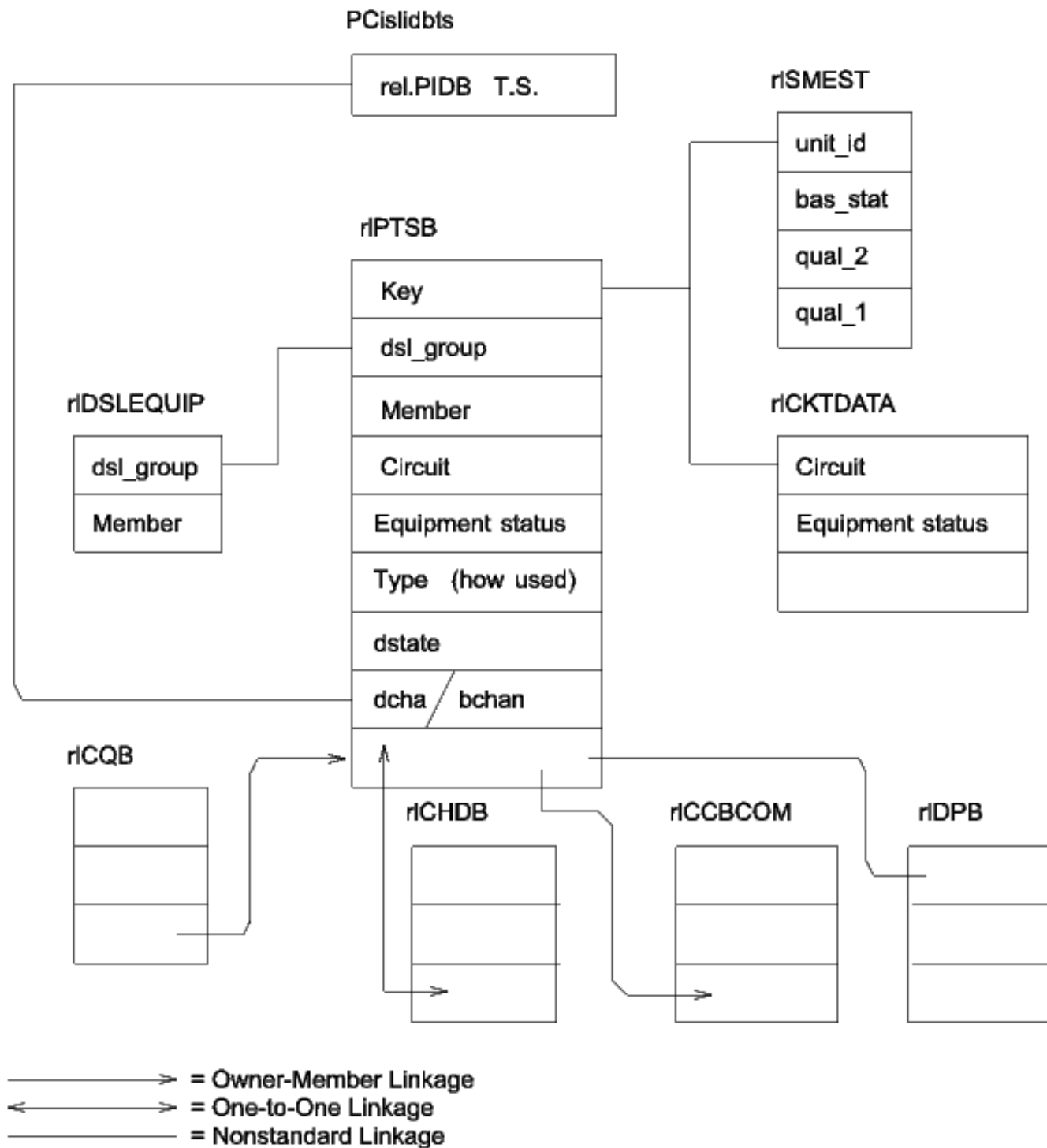


Figure 1 (PTSB-1) PTSB Data Structures

3. AUDIT CHECKS

This audit verifies the time slot use against the PTSB CD use and the CD configuration. The audit invokes the static driver (DRSTATIC) to step through the tuples of RLPTS relation. It performs the following checks.

- (a) The audit performs a dynamic-to-static existence check between rIPTSB and rICKTDATA. If rICKTDATA does not have a corresponding tuple or if the equipment status is unequipped, the audit prints an error message and deletes the rIPTSB tuple.
- (b) The audit performs point-to/point-back checks on the linkages to rICHDB, rCCBCOM and rCQB.
- (c) The audit invokes the semantic check function if it detects no linkage errors. The audit determines how the rIPTSB tuple is used.
- (d) The audit checks the in-service/out-of-service (IS/OOS) status of the associated PIDB in rSMEST.
 - (1) If rSMEST says the PIDB is OOS, the rIPTSB tuple should be on the OOS list.
 - (2) If rSMEST says the PIDB is IS, the rIPTSB tuple should be on the idle list or busy.
- (e) If the rIPTSB tuple is associated with a directly connected PIDB (PCDCON) or an ISLU cut-through PICB (PCPISCUT), the audit performs the following checks.
 - (1) If all D-channel state (dstate) fields are idle, the tuple should be linked onto the idle list of the channel control block common area relation (rCCBCOM). All other linkages should be null.
 - (2) If a D-channel is busy, the audit checks each of the four quarter time slots. The D-channel (dchan) field of idle D-channels must be 0xff. If the dstate field is busy, point-to point-back checks are made to rIDPB. It performs a range check on the dchan field. The audit accesses rIDSLEQUIP and compares the dsl_group and member fields to the corresponding fields in rIPTSB. If they do not match, the audit sets the dchan and dstate fields to idle.
- (f) If the rIPTSB tuple is associated with a simplex B-channel (PCPSIMPB) or a duplex B-channel (PCPDUPB), the audit makes these checks.
 - (1) If the rIPTSB tuple is linked to the idle list, its other linkages should be null.
 - (2) If rIPTSB tuple is not linked onto the idle list head cell, the audit checks the ranges of its linkages to rICHDB and rCCBCOM.
- (g) For on demand B-channel (ODB) packet services, the audit calculates the logical PH and channel number from the rIPTSB port name and the parity in use in the associated D-channel rIDPB tuple in order to verify that there is a corresponding rIODPSOP static tuple. Then the PH and channel number are verified with static data (rIODPSOP).
- (h) The audit verifies the CD use in rIPTSB based on the CD configuration.

For outgoing remote digital test access (PCORDTA), the audit verifies that a one-to-one linkage exists between monitored and cut-through rIPTSB tuples.

The current and new error detection and recovery schemes performed by AUdflnkchk() of the PTSB audit are:

Table 1 Current and New Error Detection and Recovery Schemes

rIPTSB to PCdfnlink PTPB ^a Check	rIPTSB to PCdfnblink PTPB ^a Check	Error Found ^b	Recovery Action
success	success	no	
success	failure	yes	rIPTSB to idle list
failure	failure	yes	rIPTSB to idle list

failure	not done	yes	rIPTSB to idle list
Notes: a. point-to-point-back b. Audit reports an error and schedules DFMAP and DFNIBBLE audits when an error is found.			

A PTPB check will be put into this function for rIPTSB and PCdfnblink[][]. The shelf, pdb_ts, and pdb_nb will be abstracted from rIPTSB and used to index into the PCdfnblink[][] array. This will provide a rIPTSB physical key which can be compared with the key found in the rIPTSB that is passed as a parameter to the AUdflnkchk() function. If the PTPB check fails, the AUdflnkchk() function will return AUFALSE to the calling function AUdchchk() which directs the audit to recover this rIPTSB tuple.

4. ERROR HANDLING

Errors detected with a D-channels are confined to that D-channel. Generally, erroneous rIPTSB tuples are linked to the idle list. The process stored in the process control block linkage area relation (RLPCBLA) may be purged for B-channels. The RLPTSB relation is recovered and linked to the idle list. For ordered B-channel calls, if there is no related rIODPSOP tuple or PH and channel numbers, the call is invalid. The PTSB tuple is idled, and all related call data is recovered.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 2 Error Code: ASSOC_PORT

Error Code:	ASSOC_PORT
Error Description:	Point-to point-back checks between the rIPTSB tuple (pfdb.bptsb.bchan) associated with a B-channel circuit switch call and the rCCBCOM tuple (port) failed.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The port field (port) from the rCCBCOM tuple.
<i>Good Data =</i>	The B-channel port field (pfdb.bptsb.bchan) from the rIPTSB tuple.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump 1 =</i>	The rCCBCOM tuple.
<i>Dump 2 =</i>	The rIPTSB tuple found in error.

Table 3 Error Code: CCBCOM_LINK

Error Code:	CCBCOM_LINK
Error Description:	The linkage field (ptccb.st_ownk) from the rIPTSB tuple under audit to the rCCBCOM tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 4 Error Code: CDCONFIG

Error Code:	CDCONFIG
Error Description:	The CD use field in the PTSB was found to be incorrect based on the actual CD configuration for this ODB call.
Possible Error Effect:	Call processing cannot determine which CD is being used, which could lead to escalating asserts.
Corrective Action Taken:	One of the following: Active/standby configuration: The rIPTSB cduse field will be recovered. Load shared configuration: The rIPTSB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rIPTSB tuple.
<i>Bad Data =</i>	The CD stored in the PTSB.
<i>Good Data =</i>	The CD configuration.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The pcPATHCONFIG global array entry for this line unit.

Table 5 Error Code: CHAN_ZERO

Error Code:	CHAN_ZERO
Error Description:	One of the quarter time slots associated with this rIPTSB tuple is in an idle state but the member field is not in an idle state (0xff).
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the member field from the rIPTSB tuple found in error.
<i>Good Data =</i>	0xFF.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 6 Error Code: CHDB_LINK

Error Code:	CHDB_LINK
Error Description:	The linkage field (pt_ch.link) from the rIPTSB tuple under audit to the rCHDB tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index</i> =	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 7 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	Could not read the rCKTDATA tuple for the PIDB circuit associated with this rIPTSB tuple during recovery of the rIPTSB tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 8 Error Code: CQB_LINK

Error Code:	CQB_LINK
Error Description:	The linkage field (ptcqb.st_headk or ptcqb.st_tailk) from the rIPTSB tuple under audit to the rCQB tuple has been found to be out-of-range.
Possible Error Effect:	May cause asserts by the application program.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index</i> =	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 9 Error Code: DATA_CON_ERR

Error Code:	DATA_CON_ERR
Error Description:	This error occurs for one of two possible cases: (1) For a directly connected PIDB timeslot, the rUNIT_CKT static tuple exists for the given rIPTSB dynamic tuple being audited. However a corresponding rPSIUPIDB static tuple could not be accessed (where rUNIT_CKT.cktname = rPSIUPIDB.psiu_shelf) or, (2) For a "cut-thru" PIDB time slot, the rPIDB static tuple exists for the given rIPTSB dynamic tuple being audited. However, a corresponding rPICB static tuple could not be accessed (where rPICB.picb = rPIDB.(dsgcpicb & 1)).

	Both of these cases constitute a static data inconsistency which will cause this error to fire until the data is corrected manually.
Possible Error Effect:	Possible asserts, call failures, and lost RLPTSB resources.
Corrective Action Taken:	None.
Manual Action Required:	Corresponds to Error Description as follows: (1) For the case where a rLPSIUPIDB tuple is missing, insert the missing tuple into the database by setting the tuple key (rLPSIUPIDB.psiu_shelf) to the <i>cktname</i> field in RLUNIT_CKT. For this error case, this value will be dumped as the logical key. (2) For the case where the rLPICB tuple is missing, insert the missing tuple into the data base by setting the tuple key (rLPICB.picb) to the expected key value calculated by the audit. For this error case, this value will be dumped as the logical key.
Dump Description:	
<i>Error Address</i> =	The relation ID of the missing tuple (RLPSIUPIDB, or RLPICB).
<i>Bad Data</i> =	For a missing RLPICB tuple, the expected key value calculated by the audit for rLPICB.picb. Otherwise, NA.
<i>Good Data</i> =	The logical key of the rIPTSB tuple under audit.
<i>Logical Index</i> =	For a missing RLPICB tuple, the expected key value calculated by the audit for rLPICB.picb, otherwise, the key of RLUNIT_CKT (rLUNIT_CKT.ckt_name).
<i>Dump</i> =	NA.

Table 10 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The rIPTSB tuple and the data fanout link (PCdfnlink[]) array or the data fanout nibble link (PCdfnblink[]) array failed point-to-point-back checks.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIPTSB tuple.
<i>Bad Data</i> =	The logical key field (ptsb) from the rIPTSB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The protocol handler shelf << 24 or The relative protocol handler << 16 or The protocol handler data bus time slot << 4 or The protocol handler data bus nibble.
<i>Dump 1</i> =	The rIPTSB tuple found in error.
<i>Dump 2</i> =	The data fanout link array (PCdfnlink[]) entry.
<i>Dump 3</i> =	The data fanout nibble link array (PCdfnblink[]) entry.

Table 11 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	One of the quarter time slots associated with this rIPTSB tuple is in a busy state but the point-to-point-back check between the rIPTSB tuple and the rIDPB tuple failed.
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The rIDPB tuple.

Table 12 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The logical PH and channel number stored in the PTSB could not be successfully verified with the PH and channel number stored in static data (rIODPSOP) for this ODB call. There are 3 possible reasons for this error: (1) There was either an incorrect party bit set, or no party bit set at all, in the D-channel DPB, which corresponds to this ODB call. (2) An rIODPSOP static tuple could not be found for the D-channel port (key of the D-channel DPB) and party bit set in the party flags register which were obtained from the D-channel DPB tuple. (3) The logical PH and channel number stored in static data (rIODPSOP) did not match the ones found in the rIPTSB tuple.
Possible Error Effect:	A user might be using the ODB feature without being a subscriber to this feature.
Corrective Action Taken:	The PTSB will be recovered to an idle state, and the DPB audit will be scheduled as a related audit to clean up the B-channel information in the B-channel rIDPB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The party found the D-channel DPB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The B-channel DPB with key corresponding to the PTSB portname.
<i>Dump 3 =</i>	The D-channel DPB with key calculated from the B-port.
NOTE: If the read of rIODPSOP was successful, the following data will also be dumped:	
<i>Dump 4 =</i>	The rIODPSOP static tuple obtained from the D-channel port and party.

Table 13 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	An attempt was made to read the rEQUIPDSL tuple for either a quarter or non-ODB full time slot based on the logical PH stored in the rIPTSB, but the rEQUIPDSL tuple was found not to exist.
Possible Error Effect:	The data in the rIPTSB tuple is either corrupted or invalid, and it cannot be determined what equipage is associated with the tuple.
Corrective Action Taken:	Corrective action taken as follows: Full Time Slots: The tuple and recovered to an idle state. Non-Full Time Slots: The quarter time slot information will be idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The logical PH number found in the rIPTSB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIPTSB tuple.

<i>Dump</i> =	The rIPTSB tuple.

Table 14 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The dynamic rIPTSB tuple is in the data base but the corresponding static rCKTDATA tuple does not exist in the database.
Possible Error Effect:	Call processing may erroneously try to use the rIPTSB tuple.
Corrective Action Taken:	The extra rIPTSB tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 15 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	All quarter time slots are in an idle state but the rIPTSB tuple is not linked onto the out-of-service or idle in-service list.
Possible Error Effect:	The rIPTSB tuple is lost to the system causing ISDN call failures.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 16 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	The rIPTSB is in an idle state but it is not linked to either the idle in-service list or the out-of-service list. The list that the rIPTSB tuple should be linked to is dependent on the state of the PIDB circuit. The state of the PIDB circuit is determined by information in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service (bas_stat = SMOOS) state or in the growth state (qual_1 = SMGR) then the rIPTSB tuple should be linked to the out-of-service list. Otherwise, the rIPTSB tuple should be linked to the idle in-service list.
Possible Error Effect:	If the rSMEST tuple indicates that the rIPTSB tuple is to be linked to the idle in-service list and the rIPTSB tuple is not, the rIPTSB tuple is lost to the system. This will cause ISDN customers to be without service. If the rSMEST tuple indicates that the rIPTSB tuple is to be linked to the out-of-service list and the rIPTSB tuple is not, could cause the system to take peripheral interrupts.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data</i> =	The correct list the rIPTSB tuple should be linked on.

<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump 1</i> =	The rIPTSB tuple found in error.
<i>Dump 2</i> =	The rSMEST tuple for the associated PIDB circuit.
<i>Dump 3</i> =	The rSMEST tuple for the associated PIDB circuit on the other service group (optional).

Table 17 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	One of the quarter time slots associated with this rIPTSB tuple has an invalid D-channel state (dstate).
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	If the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used as a full time slot the rIPTSB tuple will be reinitialized and linked to the idle in-service list. Otherwise, if the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used on a quarter time slot basis, the quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIPTSB tuple found in error.
<i>Bad Data</i> =	The state field (use) from the rIPTSB tuple found in error.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump</i> =	The rIPTSB tuple found in error.

Table 18 Error Code: MISSING

Error Code:	MISSING
Error Description:	The audit found a missing tuple in a static relation which is being used to provide existence checks on the rIPTSB tuple under audit. Specific information on which relation's tuple could not be accessed is provided in the <i>Dump Description</i> below.
Possible Error Effect:	Possible asserts, call failures, and lost RLPTSB resources.
Corrective Action Taken:	The rIPTSB tuple under audit will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The relation ID of the static relation that the audit tried to access.
<i>Bad Data</i> =	Varies according to the relation being accessed. The following list provides the relationship between the relation which the audit tried to access, and the <i>Bad Data</i> that is dumped: Relation ID = Bad data dumped. RLCKTTDATA = PSU connect number. RLPIDB = NA. RLUNIT_CKT = Unit type (SMPSIU). RLUNITDATA = rUNITDATA.utype and rUNITDATA.unit.
<i>Good Data</i> =	The key of the rIPTSB tuple.
<i>Logical Index</i> =	Varies according to the relation being accessed. The following list provides the relationship between the relation which the audit tried to access, and the <i>Logical Key</i> that is dumped:

	Relation ID = Logical key dumped. RLCKTDATA = rICKTDATA.name. RLPIDB = The PIDB number. RLUNIT_CKT = rUNIT_CKT.ckt_indx. RLUNITDATA = rUNITDATA.ckt_indx.
<i>Dump =</i>	For the case where a rICKTDATA tuple could not be accessed, the associated rPSIUPIDB static tuple is dumped (rPSIUPIDB.dpidb[PIDB number]). For all other cases, no data is dumped.

Table 19 Error Code: KEY

Error Code:	KEY
Error Description:	The key field (ptsb) of the rIPTSB tuple has been corrupted.
Possible Error Effect:	This rIPTSB tuple will be unusable by the switch and may cause asserts from peripheral control. If this time slot is associated with a D-channel or packet B-channel these channels will remain out-of-service.
Corrective Action Taken:	The key field (ptsb) of the rIPTSB tuple will be corrected to the appropriate value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The key field (ptsb) from the rIPTSB tuple.
<i>Good Data =</i>	The correct value of the key field (ptsb).
<i>Logical Index =</i>	The correct value of the key field (ptsb).
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 20 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One of the time slots associated with this rIPTSB tuple is in a busy state but the point-to-point-back check between the rIPTSB tuple and the rIDPB tuple failed.
Possible Error Effect:	The rIPTSB tuple is lost to the system causing ISDN or packet pipe call failures.
Corrective Action Taken:	The PTSB will be recovered to an idle state, and the DPB audit will be scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	The rIPTSB tuple usage field (rIPTSB.use).
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 21 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static rIDSLEQUIP tuple associated with the rIPTSB tuple under audit was found to be missing in the database. This could be due to bad data in the rIPTSB tuple or ODD inconsistencies between rIDSLEQUIP and rEQUIPDSL.
Possible Error Effect:	If this error is due to ODD inconsistencies between rIDSLEQUIP and rEQUIPDSL the port will continue to be setup and torn down over and over by peripheral control and audits.
Corrective Action Taken:	If the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used as a full time slot the rIPTSB tuple will be reinitialized and linked to the idle in-service list. Otherwise, if the rIPTSB tuple state field (use) indicates that the PIDB time slot is being used on a quarter time slot basis, the quarter time slot

	information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	If this error rolls out over and over on the same rIPTSB tuple then there are ODD inconsistencies between the RLDSLEQUIP and RLEQUIPDSL relations. The port, DSL group, member, and time slot information can be obtained from the dump of the rIPTSB tuple. The port information can be used to access the rIDSLEQUIP tuple and the DSL group and member can be used to access the rEQUIPDSL tuple. These tuples should then point to each other. If not these tuples should be corrected using ODBE.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 22 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	Found a linkage error when trying to traverse from the rIPTSB tuple to the owning rICCB COM tuple.
Possible Error Effect:	May cause the application programs to fire asserts because of this data inconsistency.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rISMEST tuple. If the rISMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The return code from the DA primitive that was trying to traverse the linkage from the rIPTSB tuple to the rICCB COM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) of the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 23 Error Code: PIDB

Error Code:	PIDB
Error Description:	Point-to point-back checks failed between the dynamic rIPTSB tuple and the static rIDSLEQUIP tuple. The directly connected peripheral interface data bus field (dpidb), the time slot field (timeslot), or the quarter time slot field (qts) failed the point-to point-back check.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not affect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The rIDSLEQUIP tuple.

Table 24 Error Code: PORT

Error Code:	PORT

Error Description:	The B-channel port stored in the PTSB did not match the port obtained from rEQUIPDSL for a non-ODB full time slot case.
Possible Error Effect:	The inconsistency may cause call processing applications to fire asserts.
Corrective Action Taken:	The rIPTSB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rIPTSB tuple.
<i>Bad Data =</i>	The B-channel port found in the PTSB.
<i>Good Data =</i>	The port found in rEQUIPDSL.
<i>Logical Index =</i>	The key of the rIPTSB tuple.
<i>Dump 1 =</i>	The rIPTSB tuple.
<i>Dump 2 =</i>	The rEQUIPDSL tuple.

Table 25 Error Code: PTSB_LINK

Error Code:	PTSB_LINK
Error Description:	The one-to-one linkage between monitored and cut-through rIPTSB tuples (rIPTSB.pt_ch.lnk) does not exist for a remote protocol access PTSB use of outgoing remote digital test access (PCORDTA). A relationship exists such that if the monitored rIPTSB.use field is marked PCORDTA (outgoing remote digital test access), the tuple's one-to-one linkage field rIPTSB.pt_ch.lnk should be non-zero, and should contain the key of the corresponding cut-through rIPTSB tuple to which it should be linked.
Possible Error Effect:	Remote protocol test access data will be in an inconsistent state, and as a result, test actions and operations may be impaired or fail.
Corrective Action Taken:	The monitored rIPTSB tuple will be recovered to an idle state, and the corresponding linkage in the cut-through rIPTSB tuple will be idled out if one exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the monitored rIPTSB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the monitored rIPTSB tuple.
<i>Dump =</i>	The monitored rIPTSB tuple.

Table 26 Error Code: READ

Error Code:	READ
Error Description:	The static rPICB tuple associated with the rIPTSB tuple under audit was not found in the database.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None, the code will assume that the rIPTSB tuple is associated with a ISLU instead of a RISLU.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the buffer containing the contents of the rPICB tuple.
<i>Bad Data =</i>	The key field (picb) of the rPICB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (picb) of the rPICB tuple.
<i>Dump =</i>	NA.

Table 27 Error Code: RELPIDB

Error Code:	RELPIDB
Error Description:	The rIPTSB tuple and the integrated services line unit local interface data bus time slot array (PCislidbts[]) failed point-to-point-back checks.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle

	in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The relative PIDB field (relpidbts) from the PCislidbts[] array.
<i>Good Data =</i>	The relative PIDB field (rel_pidb) from the rIPTSB tuple.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 28 Error Code: RELPTR

Error Code:	RELPTR
Error Description:	For an ODB call, an attempt was made to obtain a corresponding D-channel DPB from the key of the B-channel DPB stored on the PTSB, but either the PTSB portname has become corrupted or the D-channel DPB does not exist.
Possible Error Effect:	The on demand packet service order profile data (logical PH and channel number) cannot be accessed, and there is no D-channel DPB available.
Corrective Action Taken:	The PTSB tuple will be recovered to an idle state, and the DPB audit will be scheduled as a related audit in case the D-channel DPB is missing.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple.
<i>Bad Data =</i>	The D-channel port (key) of the DPB which was obtained.
<i>Good Data =</i>	The D-channel port calculated from the B-channel port found in the rIPTSB tuple.
<i>Logical Index =</i>	The key of the PTSB.
<i>Dump =</i>	The rIPTSB tuple.

Table 29 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	A rTSP2APP static tuple exists in the database that contains a application circuit name (rTSP2APP.app_name), but the corresponding static rCKTDATA tuple is missing from the database.
Possible Error Effect:	Asserts or interrupts may occur due to the missing rCKTDATA tuple.
Corrective Action Taken:	None.
Manual Action Required:	A rCKTDATA tuple should be inserted into the database with a key (rCKTDATA.name) or circuit name that matches the value of <i>Logical Key</i> in the Dump Description .
Dump Description:	
<i>Error Address =</i>	Relation identification of RLCKTDATA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Logical key (rIPTSB.psb) of the rIPTSB tuple.
<i>Logical Index =</i>	Application circuit name.
<i>Dump =</i>	rTSP2APP tuple.

Table 30 Error Code: SMEST

Error Code:	SMEST
Error Description:	The rSMEST tuple associated with the rIPTSB tuple under audit is not in the database.
Possible Error Effect:	Call processing and maintenance activities on the time slot associated with this rIPTSB tuple may cause asserts.
Corrective Action Taken:	The CKTDATA audit will be scheduled as a related audit to insert the rSMEST tuple back into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 31 Error Code: STATE_IDL

Error Code:	STATE_IDL
Error Description:	The rIPTSB tuple state field (use) indicates that this tuple is being used as a full time slot, however, the D-channel state (dstate) is in an invalid state.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The D-channel state field (dstate) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 32 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The state field (use) of the rIPTSB tuple is in an invalid state.
Possible Error Effect:	This rIPTSB tuple may be lost to the system. It can also cause asserts by the application program because of the invalid state.
Corrective Action Taken:	The rIPTSB tuple reinitialized and linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The state field (use) from the rIPTSB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 33 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rIPTSB tuple was found stuck in a transient state for longer than 5 minutes when trying to setup or tear down one of the quarter time slots.
Possible Error Effect:	The quarter time slot that was stuck in a setup or tear down state is lost to the system. If this time slot was a directly connected time slot the D-channel or packet B-channel associated with this quarter time slot would not be able to be established.
Corrective Action Taken:	The quarter time slot information in the rIPTSB tuple that was found in error is initialized to an idle state. Error recovery of this quarter time slot should not effect the other three quarter time slots.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The transient bit flag (autime) from the rIPTSB tuple.
<i>Good Data =</i>	DBNO.
<i>Logical Index =</i>	Logical key field (ptsb) of the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple under audit.

Table 34 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	The time slot usage field (ts_use) in the rIPTSB tuple does not agree with the

	way the time slot is being used as specified by the state field (use).
Possible Error Effect:	Maintenance activities on this time slot can result in asserts and single process purges.
Corrective Action Taken:	The time slot use field (tsuse) in the rIPTSB tuple will be updated to be consistent with the use of the rIPTSB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	Actual value of the tsuse field.
<i>Good Data =</i>	Valid value for tsuse field.
<i>Logical Index =</i>	Logical key of the rIPTSB tuple.
<i>Dump =</i>	Dump of the rIPTSB tuple.

Table 35 Error Code: TYPE

Error Code:	TYPE
Error Description:	The PTSB audit does not recognize the peripheral interface data bus (PIDB) type specified by the PIDB type (pidbtype) field in the associated rICKTDATA tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The PTSB audit must be modified to include the new PIDB type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple under audit.
<i>Bad Data =</i>	PIDB type field (pidbtype) from the rICKTDATA tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple under audit.
<i>Dump 1 =</i>	The rIPTSB tuple found in error.
<i>Dump 2 =</i>	The associated rICKTDATA tuple.

Table 36 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The rIPTSB tuple state field (use) is marked as unavailable but the rIPTSB tuple is not linked onto the unavailable list (DBPTSBNA).
Possible Error Effect:	This rIPTSB tuple may be erroneously allocated to a call.
Corrective Action Taken:	The rIPTSB tuple will be reinitialized and linked to the unavailable list (DBPTSBNA).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple in error.
<i>Bad Data =</i>	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data =</i>	The value of the unavailable list (DBPTSBNA).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple under audit.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 37 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	The rSMEST tuple associated with this rIPTSB tuple is in an out-of-service state, however, the rIPTSB tuple is not linked onto the out-of-service list.
Possible Error Effect:	If the rIPTSB tuple remains on an in-service list when the PIDB circuit is out-of-service may cause peripheral interrupts.
Corrective Action Taken:	The rIPTSB tuple is reinitialized and linked onto the out-of-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The list field (ptccm.st_ownk - DBSFCCBCOM) from the rIPTSB tuple found in error.
<i>Good Data =</i>	The out-of-service list value (DBPTSBOOS).
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 38 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIPTSB tuple is in an idle state (use = PCPNUL) and is linked to the idle in-service list or the out-of-service list, however, it is still linked to other resources. These resources may be either a rCHDB or rCQB tuples. The rIPTSB tuple was not properly reinitialized before it was placed on the idle in-service list or out-of-service list.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the linkage to be non-null.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 39 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rIPTSB tuple is in an idle state (use = PCPNUL) and is linked to the idle in-service list or the out-of-service list, however, it is still linked to other resources. These resources may be either a rCHDB or rCQB tuples. The rIPTSB tuple was not properly reinitialized before it was placed on the idle in-service list or out-of-service list.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The rIPTSB tuple is linked to the appropriate list according to the state of the PIDB circuit as specified in the rSMEST tuple. If the rSMEST tuple indicates that the PIDB circuit is in an out-of-service state (bas_stat = SMOOS) or in the growth state (qual_1 = SMGR) the rIPTSB tuple will be linked to the out-of-service list. Otherwise, the rIPTSB tuple will be linked to the idle in-service list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPTSB tuple found in error.
<i>Bad Data =</i>	The contents of the linkage to be non-null.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	The logical key field (ptsb) from the rIPTSB tuple found in error.
<i>Dump =</i>	The rIPTSB tuple found in error.

Table 40 Error Code: ZERO_TUP_BAD

Error Code:	ZERO_TUP_BAD
Error Description:	The dynamic rIPTSB tuple is in the database, but the corresponding static rTSP2APP tuple indicates that the rIPTSB tuple should not exist based on the application circuit name (rTSP2APP.app_name) being GLNULCKT.
Possible Error Effect:	Call processing may erroneously try to use the rIPTSB tuple.
Corrective Action Taken:	The extra rIPTSB tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Facility circuit name (rUNIT_CKT.cktname).
<i>Bad Data =</i>	Application circuit name (rTSP2APP.app_name).
<i>Good Data =</i>	Transport circuit name (rTSP2APP.tsp_name).
<i>Logical Index =</i>	Logical key of the rIPTSB tuple (rIPTSB.ptsb).
<i>Dump 1 =</i>	rIPTSB tuple.
<i>Dump 2 =</i>	rTSP2APP tuple.
<i>Dump 3 =</i>	rUNIT_CKT tuple.
<i>Dump 4 =</i>	rPICB tuple.

<i>Dump 5</i> = rIPIDB tuple.

27. PU - QO

PUENTRY

Software Release: 5E10 and later

1. CALL PICKUP ENTRY (PUENTRY) AUDIT

Call pick-up is a feature of the business and residential custom services (BRCS) package that allows a ringing telephone to be answered from another telephone. It is assigned in the local office and stored in office-dependent data (ODD).

This relation will contain tuples that have information stored about a call pickup queued ringing call. As a telephone rings that has call pickup terminating assigned, a RLPU_ENTRY tuple will be taken off the idle list and linked to the head cell designated by its pickup group (RLPU_QUEUE tuple).

This audit verifies the consistency of the dynamic relation RLPU_ENTRY in all SMs, in routine and elevated segmented modes.

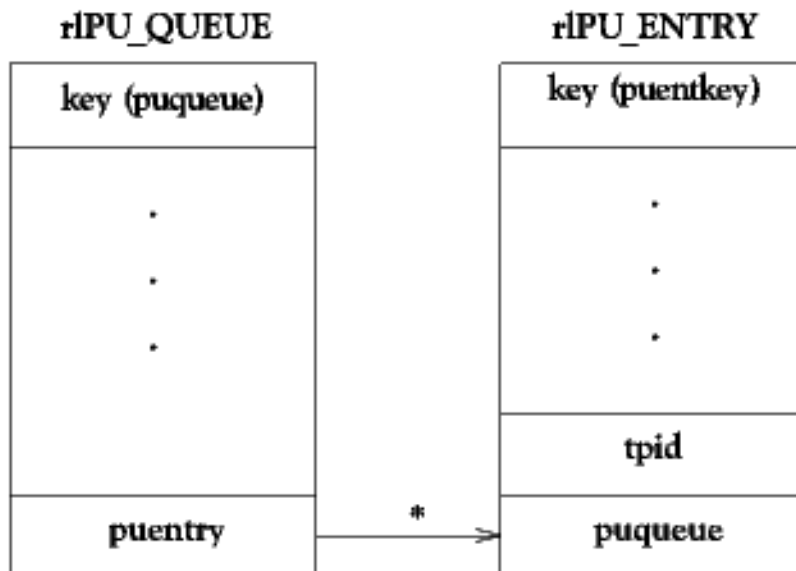
2. DATA STRUCTURES

Each tuple of the RLPU_ENTRY dynamic relation stores information about a call pick-up queued call, and is a compacted indexed relation.

The PUENTRY audit references the following dynamic relations:

RLPU_QUEUE = One tuple of RLPU_QUEUE acts as the idle list headcell for the relation RLPU_ENTRY, linked through *rlpu_queue.puentry*.

Figure 1 shows the relationship of these data structures.



=set member linkage

*The rIPU_QUEUE tuple with the key DBIDLEPUQUE is the idle list headcell for RLPU_ENTRY; otherwise, busy rIPU_ENTRY tuples are linked to the rIPU_QUEUE tuple whose Call Pick-up queue they are on.

Figure 1 (PUENTRY-1) Data Structures of the PUENTRY Audit

3. AUDIT CHECKS

The PUENTRY audit uses the DRINASEK audit driver. For each rIPU_ENTRY tuple found in the relation, the audit makes the following checks:

NONNULL = If the rIPU_ENTRY tuple is idle, the terminating process ID should be NULL.

If the rIPU_ENTRY tuple is busy:

NULLCNTL = The terminating process ID should not be OSnullpid.

PID = When the terminating process ID is on the current SM, the process should be alive.

4. ERROR HANDLING

The audit produces the following error codes and those generated by the DRINASEK driver.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The rIPU_ENTRY tuple is linked to the idle list, but the terminating process ID is not null.
Possible Error Effect:	This tuple is on the idle list but may still be connected to an active call. If the tuple is retrieved for a new call, very strange things may occur.
Corrective Action Taken:	The terminating process ID is nulled out; no recovery is scheduled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPU_ENTRY tuple where the error was found.
<i>Bad Data =</i>	The ID of the terminating process (<i>rIPU_ENTRY.tpid</i>).
<i>Good Data =</i>	The owner of the rIPU_ENTRY tuple (<i>rIPU_ENTRY.st_ownnk - DBSFPU_QUE</i>).
<i>Logical Index =</i>	The key to the rIPU_ENTRY tuple (<i>rIPU_ENTRY.puentkey</i>).
<i>Dump =</i>	The rIPU_ENTRY tuple where the error was found.

Table 2 Error Code: NULLCNTL

Error Code:	NULLCNTL
Error Description:	The rIPU_ENTRY tuple is linked to a busy list, but the terminating process ID is null.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIPU_ENTRY tuples.
Corrective Action Taken:	The rIPU_ENTRY tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPU_ENTRY tuple where the error was found.
<i>Bad Data =</i>	The ID of the terminating process (<i>rIPU_ENTRY.tpid</i>).
<i>Good Data =</i>	The owner of the rIPU_ENTRY tuple (<i>rIPU_ENTRY.st_ownnk - DBSFPU_QUE</i>).
<i>Logical Index =</i>	The key to the rIPU_ENTRY tuple (<i>rIPU_ENTRY.puentkey</i>).
<i>Dump =</i>	The rIPU_ENTRY tuple where the error was found.

Table 3 Error Code: PID

Error Code:	PID
Error Description:	The rIPU_ENTRY tuple is linked to a busy list and the terminating process ID is on the current SM, but the process is no longer alive.
Possible Error Effect:	This resource may not be needed any longer for the call. This may cause possible call failures if the system exhausts the number of available rIPU_ENTRY tuples.
Corrective Action Taken:	The rIPU_ENTRY tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIPU_ENTRY tuple where the error was found.
<i>Bad Data =</i>	The ID of the terminating process (<i>rIPU_ENTRY.tpid</i>).
<i>Good Data =</i>	The owner of the rIPU_ENTRY tuple (<i>rIPU_ENTRY.st_ownnk - DBSFPU_QUE</i>).
<i>Logical Index =</i>	The key to the rIPU_ENTRY tuple (<i>rIPU_ENTRY.puentkey</i>).
<i>Dump =</i>	The rIPU_ENTRY tuple where the error was found.

PUGPAST

Software Release: 5E10 and later

1. PICKUP GROUP ASSIGNMENT (PUGPAST) AUDIT

A call pickup group is a feature of the business and residence custom services-II (BRCS-II) package that allows one phone to answer calls to another phone. Pickup groups are assigned in the local office; they are stored in office-dependent data (ODD). The pickup group assignment audit (PUGPAST) verifies the consistency of the relations involved with the assignment of pickup groups.

This audit is part of the routing and terminal allocation audits subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

This audit checks two fields of the pickup owner relation (RLPUOWNER), a dynamic relation in the AM: the pickup group number and the group size. The call_count field in rIPUOWNER is administered by the RTA subsystem. RTA will quarantine the rIPUOWNER tuple whenever this count exceeds the size of the group field value. The audit verifies these fields with the group number field and size fields in the pickup group assignment relation (RLPUGPAST), a static relation in the AM. The audit also checks the owner-member linkage between RLPUOWNER and the pickup queue relation (RLPTQUE). Figure 1 shows these data structures.

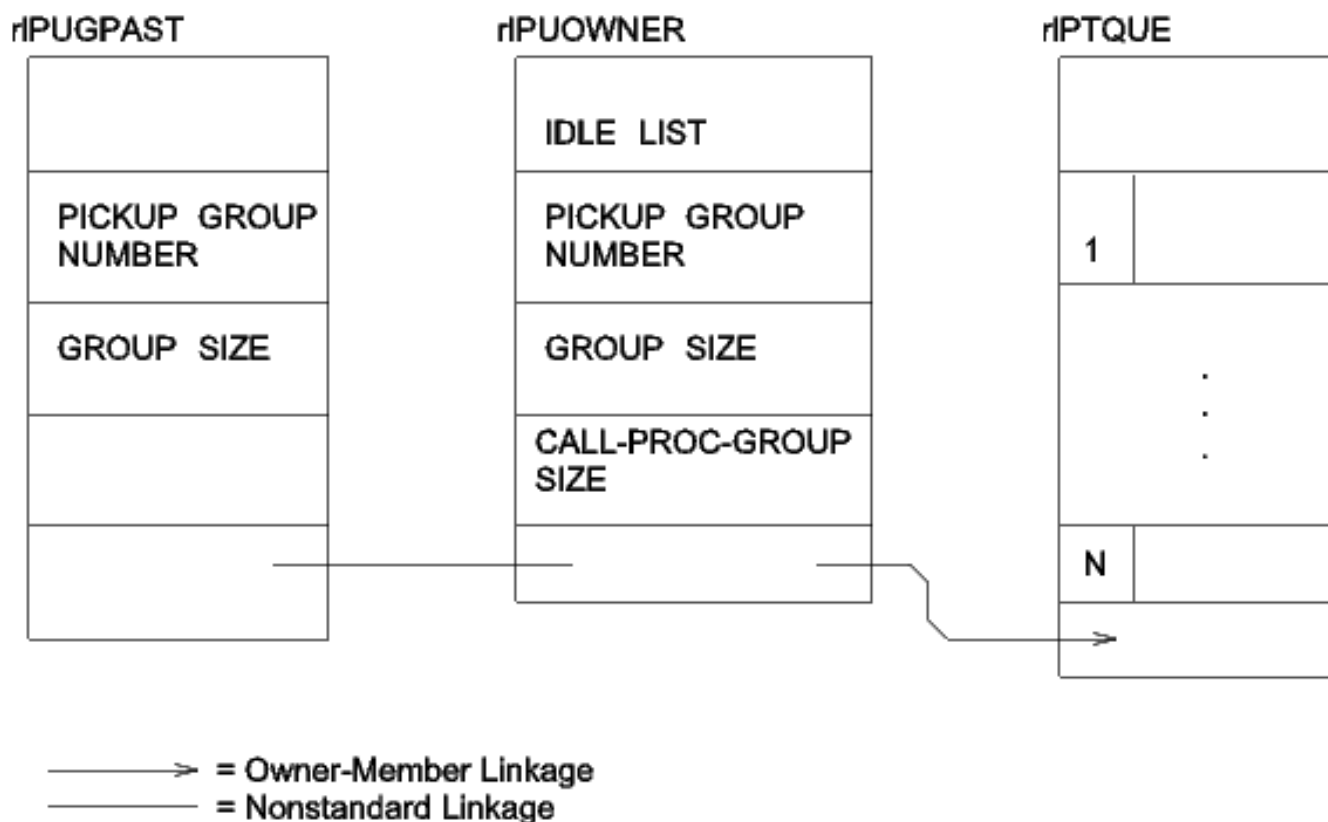


Figure 1 (PUGPAST-1) Data Structures of the PUGPAST Audit

3. AUDIT CHECKS

The PUGPAST audit performs syntactic and semantic checks.

- (a) First the audit performs static-to-dynamic existence checks. It calls the static driver (DRSTATIC) to step through rPUGPAST to obtain the pickup group number and ensure that a corresponding tuple exists in RLPUOWNER. If no tuple exists in RLPUOWNER, one is created.
- (b) The audit performs syntactic checks on each rPUOWNER tuple. The checks include a key check, quarantine check, and linkage checks of the owner-member linkage between rPUOWNER and rPTQUE.
- (c) The audit then performs a semantic check on the group size fields. The RLPUOWNER relation defines the size of the group. The RLPUOWNER relation contains the size if the group field is populated from RLPUOWNER. If there is a discrepancy between these fields, the audit reports an error.

4. ERROR HANDLING

If the audit fails the existence check by not finding in RLPUOWNER a corresponding tuple for one listed in RLPUOWNER, the audit inserts a tuple in RLPUOWNER and schedules the pickup queue audit (PTQUE).

If the linkage check fails, the tuple in RLPUOWNER is initialized and linked to the idle list. The PTQUE audit is scheduled.

If the call processing group size field in rPUOWNER is not the same size as the size defined by rPUGPAST, it is corrected and the PTQUE audit is scheduled.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The call count field in rPUOWNER is nonzero with a zero head key.
Possible Error Effect:	Queues may unnecessarily be flushed.
Corrective Action Taken:	The call count field is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple.
<i>Bad Data</i> =	call_count.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Pickup group of the tuple.
<i>Dump</i> =	rPUOWNER tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Static data assignment indicates the call pickup owner relation rPUOWNER should have a tuple.
Possible Error Effect:	Inability to queue a call pickup member on the indicated pickup group's queue.
Corrective Action Taken:	The tuple is created.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address where tuple should have been.
<i>Bad Data</i> =	Valid values are: 1 = Tuple created. 0 = Tuple was not created.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Pickup group of the tuple.
<i>Dump</i> =	NA.

Table 3 Error Code: SGROUP

Error Code:	SGROUP
Error Description:	Size (of group) field in rIPUOWNER contains incorrect value.
Possible Error Effect:	All the members linked to this queue may be unnecessarily idled.
Corrective Action Taken:	Tuple is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad size of group value.
<i>Good Data</i> =	Good size of group value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rIPUOWNER tuple.

PUGSM

Software Release: 5E10 and later

1. CALL PICKUP GROUP (PUGSM) AUDIT

Call pick-up is a feature of the business and residential custom services (BRCS) package that allows a ringing telephone to be answered from another telephone. It is assigned in the local office and stored in office-dependent data (ODD).

This audit verifies the consistency of the static relation RLPUG_SM.

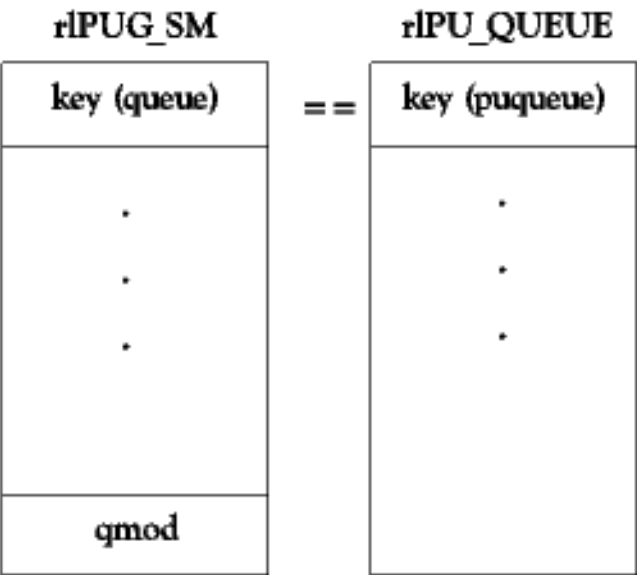
2. DATA STRUCTURES

The RLPUG_SM relation indicates on which SM a given call pick-up group is being queued. RLPUG_SM is a hashed relation.

The PUGSM audit references the following dynamic relations:

RLPU_QUEUE = One tuple of the dynamic relation RLPU_QUEUE must exist for each tuple in relation RLPUG_SM.

Figure 1 shows the relationship of these data structures.



For each rIPUG SM tuple where the qmod is the current SM, there exists one rIPU_QUEUE tuple.

Figure 1 (PUGSM-1) Data Structures of the PUGSM Audit

3. AUDIT CHECKS

The PUQUEUE audit uses the DRSTATIC audit driver, and has no other audit errors to report other than the

standard driver errors.

4. ERROR HANDLING

The audit produces only the error codes generated by the DRSTATIC driver.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

PUOWNER

Software Release: 5E10 and later

1. PICKUP OWNER (PUOWNER) AUDIT

A call pickup group defines the phones that are allowed to answer a call to another phone once the call is in the ringing state. Included in the business and residence custom services-II (BRCS-II) call pickup feature are call pickup, directed call pickup, directed pickup without barge-in, loudspeaker paging, code calling, and trunk answer from any station. Calls to a member of a pickup group are placed on call pickup queues. The pickup queues and the other data structures managing the pickup groups are administered by the routing and terminal allocation (RTA) subsystem.

The PUOWNER audit is part of the routing and terminal allocation audits subgroup. It runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

This audit operates on the pickup owner relation (RLPUOWNER), a dynamic relation in the AM, and the pickup group assignment relation (RLPUGPAST), a static relation in the AM. The RLPUGPAST relation is an office-dependent data (ODD) recent changeable relation that contains lists of pickup groups. The RLPUOWNER relation is populated from RLPUGPAST and should contain a tuple for each pickup group in RLPUGPAST. Figure 1 shows these relations.

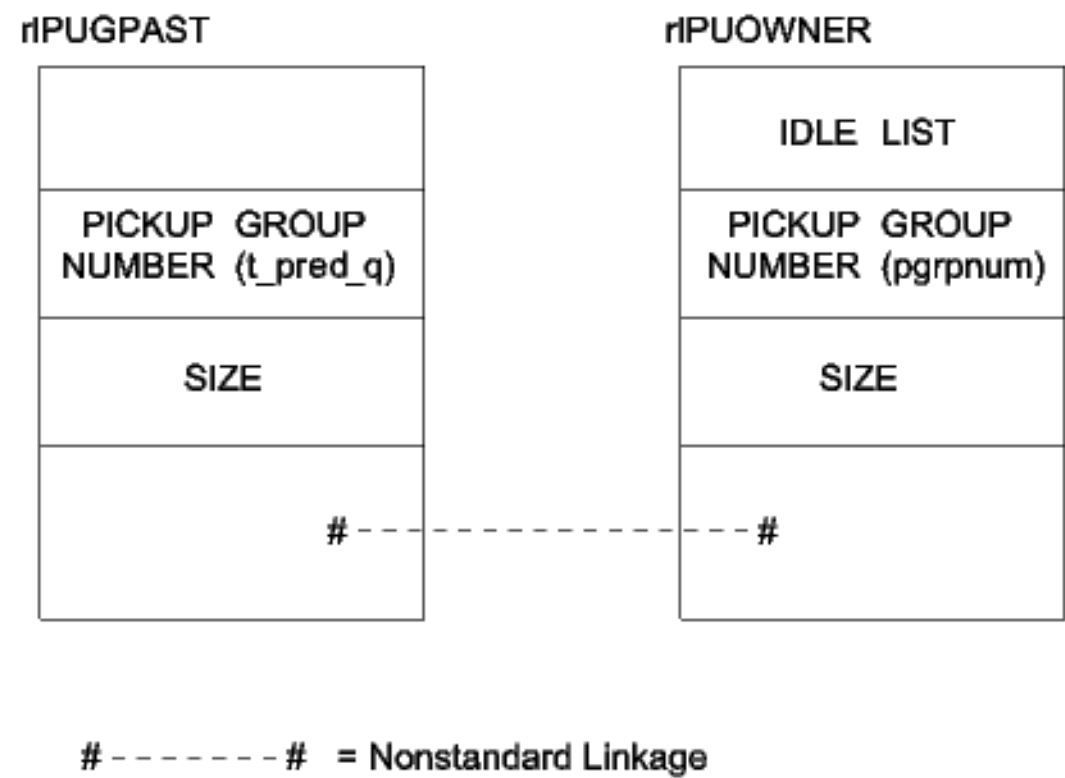


Figure 1 (PUOWNER-1) Data Structures of the PUOWNER Audit

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) to perform dynamic-to-static existence checks between RLPUOWNER and RLPUGPAST. The audit uses the pickup group number (pgrpnum) as the key and ensures that

there is a corresponding tuple (t_pred_q) in RLPUGPAST for each rIPUOWNER tuple.

4. ERROR HANDLING

The audit deletes surplus rIPUOWNER tuples.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	Static data indicates the owner relation RLPUOWNER should not have this tuple.
Possible Error Effect:	When assigning the pickup group, an assert will fire.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

PUQUEUE

Software Release: 5E10 and later

1. CALL PICKUP QUEUE (PUQUEUE) AUDIT

Call pick-up is a feature of the business and residential custom services (BRCS) package that allows a ringing telephone to be answered from another telephone. It is assigned in the local office and stored in office-dependent data (ODD).

This audit verifies the consistency of the dynamic relation RLPU_QUEUE.

2. DATA STRUCTURES

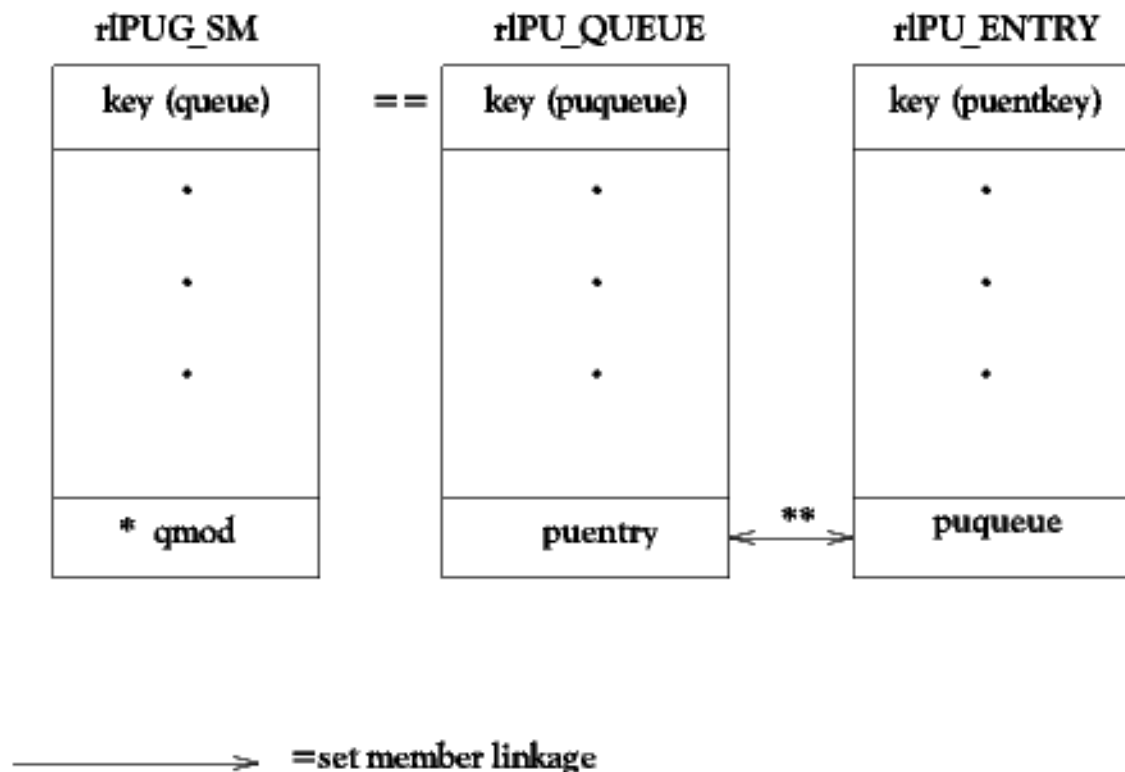
The first tuple of the RLPU_QUEUE dynamic relation is the idle headcell for the RLPU_ENTRY tuples. Each other tuple of the RLPU_QUEUE dynamic relation stores information about a call pick-up group and acts as the owner of the call pick-up queue. RLPU_QUEUE is a compacted indexed relation.

The PUQUEUE audit references the following dynamic relations:

RLPUG_SM = For each tuple in RLPUG_SM, there exists a tuple with the same key in RLPU_QUEUE.
There is no explicit linkage.

RLPU_ENTRY = One tuple of RLPU_QUEUE acts as the idle list headcell for the relation RLPU_ENTRY. The other tuples of RLPU_QUEUE will be linked to appropriate RLPU_ENTRY tuples when calls are made using that call pick-up queue. Tuples are linked to RLPU_ENTRY through *RLPU_QUEUE.puentry*.

Refer to Figure 1 .



***For each rPUG_SM tuple where the qmod is the current SM, there exists one rIPU_QUEUE tuple.**

****The rIPU_QUEUE tuple with the key DBIDLEPUQUE is the idle list headcell for RLPU_ENTRY; otherwise, busy rIPU_ENTRY tuples are linked to the rIPU_QUEUE tuple whose Call Pick-up queue they are on.**

Figure 1 (PUQUEUE-1) Data Structures of the PUQUEUE Audit

3. AUDIT CHECKS

The PUQUEUE audit uses the DRINDEX audit driver, and has no other audit errors to report other than the standard driver errors.

4. ERROR HANDLING

The audit produces only error codes generated by the DRINDEX driver.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

PVC

Software Release: 5E10 and later

1. PERMANENT VIRTUAL CIRCUIT (PVC) AUDIT

The permanent virtual circuit audit is an interprocessor audit that checks data in the protocol handlers (PHs) against data in the switching module processor (SMP).

The PVC audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

The PVC audit uses the packet switching logical channel control block array (PSlccb) in the PH and the permanent virtual circuit information relation (RLPVCINFO) in the SMP.

3. AUDIT CHECKS

The audit checks the permanent virtual circuit data stored in the PSlccb array in the PHs against the data stored in the RLPVCINFO relation in the SMP.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BQSIZE

Error Code:	BQSIZE
Error Description:	The "s_ wsize" field in the PH's PSlccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("r_ wdsize").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSlccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 2 Error Code: CINDX

Error Code:	CINDX
Error Description:	The "aphend" field in the PH's PSlccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("callind").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSlccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 3 Error Code: DDAMABIL

Error Code:	DDAMABIL
Error Description:	The "billfig" field in the PH's PSlccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("billable").
Possible Error Effect:	The PVC could be billed incorrectly.
Corrective Action Taken:	The PSlccb[] is updated to agree with the RLPVCINFO relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSlccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 4 Error Code: EXTQSIZE

Error Code:	EXTQSIZE
Error Description:	The "s_psize" field in the PH's PSlccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation (r_pktsize).
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSlccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 5 Error Code: INTHRUPT

Error Code:	INTHRUPT
Error Description:	The "send" throughput class in the PH's PSlccb[].vc_tpc disagrees with the "r_thruput" field in rIPVCINFO.
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSlccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 6 Error Code: OUTHRUPT

Error Code:	OUTHRUPT
Error Description:	The "receive" throughput class in the PH's PSlccb[].vc_tpc disagrees with the "s_thruput" field in rIPVCINFO.
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSlccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	NA.

Table 7 Error Code: LVLNUM

Error Code:	LVLNUM
Error Description:	The "pvc_id_num" field in the PH's PSlccb[] array does not agree with the "pvc_id_num" field in the SMP ODD RLPVCINFO relation.
Possible Error Effect:	The billing records for that PVC will not be correct.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The LCN number.
<i>Bad Data</i> =	Value of the pvc_id_num field in the PSiccb[] array.
<i>Good Data</i> =	Value of the pvc_id_num field in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 8 Error Code: PKT_TYPE

Error Code:	PKT_TYPE
Error Description:	The "fac_type" field in the PH's PSiccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("fcltype").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSiccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 9 Error Code: PQSIZE

Error Code:	PQSIZE
Error Description:	The "r_psize" field in the PH's PSiccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("s_pktsize").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSiccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 10 Error Code: R_RMSTAT

Error Code:	R_RMSTAT
Error Description:	The "rem_lcn" field in the PH's PSiccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("remote_lcn").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSiccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 11 Error Code: RQSIZE

Error Code:	RQSIZE
Error Description:	The "r_wsize" field in the PH's PSiccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("s_wdsize").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSiccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 12 Error Code: TIDBAD

Error Code:	TIDBAD
Error Description:	The "Inkid_g1p" field in the PH's PSiccb[] array does not agree with the value in the SMP ODD RLPVCINFO relation ("Inkid_g1p").
Possible Error Effect:	The PVC could be out of order.
Corrective Action Taken:	The PVC is torn down and routine port conditioning should restore it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The LCCB index.
<i>Bad Data</i> =	Value found in the PSiccb[] array.
<i>Good Data</i> =	Value found in the RLPVCINFO relation.
<i>Logical Index</i> =	PH channel number.
<i>Dump</i> =	Message with the data which was in the RLPVCINFO relation.

Table 13 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit has been entered by audit control to clean up after a reset.
Possible Error Effect:	Causes resets and or asserts.
Corrective Action Taken:	The audit is restarted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

QGP**Software Release:** 5E10 and later**1. QUAD-LINK GATEWAY PROCESSOR (QGP) AUDIT**

The QGP audit consists of checks done for both the QGP and QGL. This audit is run in the SMKP environment and uses a two pass mechanism for all checks other than the internal PCD state check.

2. DATA STRUCTURES

The QGP audit makes reference to the following data structures:

- SMdpc[] (PCD information related to QGP).
- SMdqml[] (PCD information related to QGL).
- RLSCORE relation (stores ONTC information).
- RLSPC relation (stores PC unit information).

3. AUDIT CHECKS

The following checks are performed to audit the QGPs:

ZAP	= A reset was taken during the audit.
INT_STATE	= Check for invalid QGP PCD state with regards to QGP equipage.
PARENT	= Check for inconsistent QGP state with parent MSCU.
EXTER	= Verify if QGP is OOS-MIP, there is something on SMARS queue for QGP or MSCU.
EXTER2	= Verify that if parent MSCU is ACT, then QGP is not OOS-FE.
IJA	= Verify IJA status for QGP.
DMAM	= Verify if QGP is in DMAM state, then something is on DMAM's queue.
HWSW_IJA	= Verify QGP SW IJA state against HW IJA state.

The following checks are performed to audit the QGLs:

ZAP	= A reset was taken during the audit.
UNEQUIP	= Verify that a QGL is OOS-UNEQ or not.
ODDLOSTTUP	= Verify read of RLSCORE for QGL.
BADDATAR	= Verify <i>qtmslnk</i> and <i>pseudo_lnk</i> values in PCD for QGL.
OUTOFRANGE	= Verify that <i>qtmslnk</i> number is within range for QGL.
INT_STATE	= Check for invalid QGL PCD state with regards to QGP equipage.
PARENT	= Check for inconsistent QGL state with parent QGP and QTMSLNK.

EXTER = Verify that if parent QGP and QTMSLNK are ACT, then QGL is not OOS-FE.

EXTER2 = Verify that a QGL is truly in an OOS-GROW-RMV case.

HWSW_CONFIG = Verify SW QGL PCD state against QGL state in QGP.

4. ERROR HANDLING

Different recovery is taken depending upon the error reported. Refer to the **Corrective Action Taken** section of the individual error.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	The qtmslnk number or the pseudo_lnk number in SMDqgl[][] was found to be incorrect.
Possible Error Effect:	Incorrect link information could be used.
Corrective Action Taken:	The audit updates the qtmslnk and pseudo_lnk to their correct values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID [0x38 for quad gateway link (QGL)].
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	Packet bus media access controller (PBMAC) number.
<i>Logical Index</i> =	Message switch (MSGs) side physical quad gateway processor (QGP) (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 2 Error Code: DMAM

Error Code:	DMAM
Error Description:	The QGP is in an out-of-service (OOS) state where there should be some deferred work being done on it, but deferred maintenance administrator and monitor (DMAM) does not have an entry in its queue for this unit.
Possible Error Effect:	QGP is OOS with no automatic action pending to restore it to service.
Corrective Action Taken:	Audit requests a conditional restore of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x36 for QGP).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 3 Error Code: EXTER

Error Code:	EXTER
Error Description:	The QGP is in an out-of service maintenance in progress (OOS-MIP) state, and there is no entry for it on the switch maintenance analysis and recovery strategies (SMARS) queue. The QGL is in an OOS-FE state, and both parents (QTMSLNK and QGP) are ACT.
Possible Error Effect:	The QGP is OOS-MIP, and there is no maintenance action pending on it. QGL not ACT when it should be.
Corrective Action Taken:	The QGP state is written to OOS AUTO RMV, and a conditional restore is requested on the QGP. The QGL will be conditionally restored.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Unit ID (0x36 for QGP, 0x38 for QGL).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	Packet bus media access controller (PBMAC) number (for QGL unit only).
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 4 Error Code: EXTER2

Error Code:	EXTER2
Error Description:	The QGP has been found to be out-of-service family of equipment (OOS-FE) when the parent message switch control unit (MSCU) is active (ACT). For the QGL, the OOS-GROW state is inconsistent with a check of the number of the operational QGPs.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in switch maintenance kernel process (SMKP). Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The state is written to OOS-AUTO-RMV for the QGP, and a conditional restore of the QGP is requested. If the error is related to the QGL, then the QGL is placed in a valid state consistent with its parents.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x36 for QGP, 0x38 for QGL).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	PBMAC number (for QGL unit only).
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 5 Error Code: HWSW_CONFIG

Error Code:	HWSW_CONFIG
Error Description:	The software (SW) state of the QGL does not agree with the hardware (HW) state of the QGL.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	A conditional restore is requested for the QGL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x38 for QGL).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	PBMAC number.
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 6 Error Code: HWSW_IIA

Error Code:	HWSW_IIA
Error Description:	The QGP software and hardware interrupt inhibit states do not agree.
Possible Error Effect:	Hardware interrupts could be inhibited when the software says they are allowed, or allowed when the software says they are inhibited.
Corrective Action Taken:	The audit requests the hardware interrupt inhibit status to be set to what the software interrupt inhibit status is. If the interrupt was inhibit manual, the audit changes them to inhibit automatic.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x36 for QGP).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 7 Error Code: IIA

Error Code:	IIA

Error Description:	The QGP software interrupt inhibit state is invalid.
Possible Error Effect:	Fault recovery could be asserting in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The interrupt inhibit state is written to a state that is consistent with the state of the unit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID (0x36 for QGP).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	MSGs side physical QGP (32 or 33).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 8 Error Code: INT_STATE

Error Code:	INT_STATE
Error Description:	The QGP or QGL has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID (0x36 for QGP, 0x38 for QGL).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	PBMAC number (for QGL unit only).
<i>Logical Index =</i>	MSGs side physical QGP (32 or 33).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 9 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit was unable to read the rISCORE tuple associated with this QGL.
Possible Error Effect:	Unable to perform checks that require data from RLSCORE.
Corrective Action Taken:	None.
Manual Action Required:	User needs to update the ODD to correct this problem.
Dump Description:	
<i>Error Address =</i>	Unit ID (0x38 for QGL).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	PBMAC number.
<i>Logical Index =</i>	MSGs side physical QGP (32 or 33).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The qtmslnk number retrieved from SMDqgl[][] was found to not be within the range of valid quad linked packet switch time multiplexed switch network control and timing I-link (QTMSLNK) numbers.
Possible Error Effect:	Incorrect information returned from SMddread().
Corrective Action Taken:	Related audits run to clean up this error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID (0x38 for QGL).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	PBMAC number.
<i>Logical Index =</i>	MSGs side physical QGP (32 or 33).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 11 Error Code: PARENT

Error Code:	PARENT
Error Description:	The QGP or QGL has been found to be in an invalid state with relation to its parents. For the QGP, the parent is the MSCU (message switch control unit). For the QGL, the parents are the QGP and the QTMSLNK with an associated

	unit of QLPS.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the unit in a valid state with relation to its parents and associated units (if applicable).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x36 for QGP and 0x38 for QGL).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	PBMAC number (for QGL unit only).
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 12 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	According to the number of operational QGPs, this QGL should be OOS-UNEQ-NULL.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit updates the state of the QGL to OOS-UNEQ-NULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x38 for QGL).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	PBMAC number.
<i>Logical Index</i> =	MSGs side physical QGP (32 or 33).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 13 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit clears SMAuditbit and returns.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUpcdpass.
<i>Bad Data</i> =	SMAuditbit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

QISAT

Software Release: 5E10 and later

1. ISAT QUEUE INFORMATION (QISAT) AUDIT

The QISAT audit verifies that for every rIQ_ISAT tuple which exists there is a corresponding rIRT_MHG tuple. The rIRT_MHG tuple is checked to make sure the hunt type is set to either RTH_ISAT or RTH_LGISAT. The audit also verifies that the module in the rIRT_MHG tuple is set to the current module.

The QISAT audit uses the DRINDEX driver which steps through each non-vacant tuple and calls the existence function to do checks on the tuple. If the tuple being checked is found to be in error, the audit prints an error and the tuple is deleted.

The dynamic relation RLQ_ISAT contains the isat index which corresponds to the hunt group. This relation is keyed by the isat_idx and contains the current size of the entire isat queue, and also the quarantine status. The static RLRT_MHG relation contains the multi-line hunt group (MLHG) information.

2. DATA STRUCTURES

The RLQ_ISAT relation contains the isat index corresponding to the hunt group. The QISAT audit is part of the AURtaque module. QISAT runs in the segmented mode and operates in the OSDS-M environment.

3. AUDIT CHECKS

The objectives of the QISAT audit are to check to see if the hunt type in RLRT_MHG is RTH_ISAT or RTH_LGISAT and that the qmodule is set to the current module. If these conditions are met, an rIQ_ISAT tuple should exist. If these conditions are not met, the tuple is considered as an extra tuple and it is deleted. The DRINDEX driver checks the key and quarantine bit to see if they are correct.

4. ERROR HANDLING

When an error is detected in the existence function, the tuple is deleted. An error message is printed on the ROP. After this, audit control requests that the RTMHG audit be run in the elevated mode.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BQSIZE

Error Code:	BQSIZE
Error Description:	Basic queue size in owner relation for isat subqueues relation RLQ_ISAT contains incorrect value.
Possible Error Effect:	Multiline hunt group queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Basic queue size is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Bad basic queue size value.
<i>Good Data =</i>	Good basic queue size value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQ_ISAT tuple.

Table 2 Error Code: EXTQSIZE

Error Code:	EXTQSIZE

Error Description:	Extended queue size in owner relation for isat subqueues relation RLQ_ISAT contains incorrect value.
Possible Error Effect:	Multiline hunt group queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Extended queue size is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad extended queue size value.
<i>Good Data</i> =	Good extended queue size value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQ_ISAT tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an extra rlQ_ISAT dynamic tuple. A relationship exists such that for every rlQ_ISAT tuple there should be a corresponding rllCX_ICI static tuple where (key) rllCX_ICI.isat.idx = (key) rlQ_ISAT.icx_ici.
Possible Error Effect:	Malfunction of ISAT MLHGs which are hunted and queued may take place.
Corrective Action Taken:	The extra rlQ_ISAT dynamic tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Logical address of the rlQ_ISAT tuple in error.
<i>Bad Data</i> =	The logical key of RLQ_ISAT (isat_idx).
<i>Good Data</i> =	NA
<i>Logical Index</i> =	The logical key of RLQ_ISAT (isat_idx).
<i>Dump</i> =	The contents of the rlQ_ISAT tuple.

Table 4 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	This error code is used for two cases in this audit. One case is the QU_OS read and the ICX_ICI read. If either of these relations return DBNO_MATCH on the read then there is a missing static tuple.
Possible Error Effect:	One will not be able to read the static data associated to the dynamic relation RLQ_ISAT.
Corrective Action Taken:	None.
Manual Action Required:	There is manual action required.
Dump Description:	
<i>Error Address</i> =	Address of the tuple in error.
<i>Bad Data</i> =	Group number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of either the QU_OS or ICX_ICI tuple in error.
<i>Dump</i> =	NA.

QLLCB-A

Software Release: 5E10 - 5E11

1. QLPS LOGICAL LINK CONTROL BLOCK (QLLCB) AUDIT

QLLCBs (QLPS logical link control blocks, elements of `QLlcb[ ]`) are used to represent layer 2 protocol information for a quad linked packet switch (QLPS) network endpoint. QLLCBs also own lists of frames associated at protocol layer 2 with the endpoint they represent.

The QLLCB audit also audits the message transport logical link control block (MTLLCB) in the second message handler (MH) of the digital networking unit-synchronous optical network (DNU-S). The DNU-S is a new peripheral for the 5ESS[®] switch that provides a cost of goods (COGS) reduction and synchronous optical network (SONET) capabilities. The second message handler provides all of the MH resident operational and maintenance software for establishing and maintaining communication paths using the peripheral control and timing (PCT) link between the SMP of SM-2000 and the DNU-S. Link access protocol for D-channel (LAPD) will be used to communicate to the DNU-S using the PCT links over three fixed timeslots as control to each common control (CC) of DNU-S.

The QLLCB audit runs in a segmented mode and is part of the MH0 and second MH base level loop.

2. DATA STRUCTURES

The following data is used for the QLLCB audit:

`QLlcb[ ]` = QLPS logical link control blocks

`QLlmbd[ ]` = QLPS link manager data blocks

`SMqpipestate[ ][ ]` = QPIPE state per network

In the second MH, the QLLCB audit checks the `MTlcb[ ]` array made up of MTLLCB data structure elements. Some of the fields audited out of this data structure are:

chan = Channel number.

link_id = Logical link ID.

sapi = SAPI value.

teib_ix = TEIB index.

tidx = Timer index..

ttype = Timer type.

3. AUDIT CHECKS

The following checks are performed by the QLLCB audit:

BAD_HEAD_TAIL = Check for QLLCB corrupted head cell.

BADIDLE = Check for uninitialized idle QLLCB.

BADIDLE2 = Check ttype and tidx fields.

BADLINK	= Check for bad linkage between QLLCB and owned FD.
BDCOUNT	= Check QLLCB transmit queue count.
BDLINK	= Check linkage to PCCB.
BUSYIDLE	= Check for null MTLLCB fields.
CORRUPT_IDLE	= Check for default values for idle MTLLCB.
IDLE_CODE	= Check for null fields used by MH0 for QLPS.
INV	= Check contents of level 2 profile data.
INV_TIMER	= Check QLLCB (or MTLLCB) in second MH state against existence of timer.
NONNULL	= Check for non-idle QLLCB without an active or expired timer.
NULL3LINK	= Check QLLCB FD ownership fields for out-of-range values.
ONE2ONE	= Check linkages associated with the QLLCB.
OUTOFRANGE	= Range check the fields used by MTLLCB.
OWNLINK	= Verify that non-idle QLLCB is linked to an LMDB.
SAPI_BAD	= Check sapi field.
STATE	= Check for state consistency amongst internal data.
TIMERLNK	= Check linkage to timer element.
ZAP	= A reset was taken during the audit.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	A QLLCB head cell (quad linked packet switched network logical link control block, of type QLHEADC) is corrupted. Good data indicates which list was being checked. Specifically, it is not the case that both QLHEADC.first and QLHEADC.last are null (= QLLIXNULL), and one or more of the following is true: • Either the first or last field is null (= QLLIXNULL), but the other field is not. • One or both of the first and last fields are out-of-range (> QLLNLLCBS). • The back_llcb field of the QLLCB indexed by QLHEADC.first is not null (!= QLLIXNULL).

	The forw_llcb field of the QLLCB indexed by QLHEADC.last is not null ($\neq$ QLLIXNULL).
Possible Error Effect:	Good data indicates whether the list being checked was the QLLCB timing list or QLLCB free list. If the list being checked is the QLLCB timing list, level 2 timers on the quad linked packet switched (QLPS) links may not be processed properly. If the list being checked is the QLLCB free list, level 2 linkages may not be set up because of a resource failure. In either case, the QLPS network may be degraded.
Corrective Action Taken:	The audit sets QLHEADC.first = QLLIXNULL and QLHEADC.last = QLLIXNULL. If the list being checked is the free list, the audit goes on to reconstruct the list by linking all QLLCBs with QLLCB.network = SMQLPSNONETWK onto the list. QLLCBs which belong on the timing list are recovered by other checks in this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The list's first element (QLHEADC.first).
<i>Bad Data =</i>	The list's last element (QLHEADC.last).
<i>Good Data =</i>	AUTRUE indicates that QLlcbhc[QLFREELIST] was being checked. AUFALSE indicates that QLlcbhc[QLTIMELIST] was being checked.
<i>Logical Index =</i>	The default (null) network value (SMQLPSNONETWK).
<i>Dump =</i>	NA.

Table 2 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	A quad linked packet switched logical link control block (QLLCB) is idle (QLLCB.network = SMQLPSNONETWK), but is not correctly initialized. Specifically, one or more of the following is true: - QLLCB.state $\neq$ QLS_LD (the "link disconnect" state). - One or more entries of QLLCB.l2time[] are not idle (the QLLCB has a timer pending on it). - One or more of the QLLCB frame descriptor ownership fields is not null (f_ifd, l_ifd, ready or blocked are $\neq$ QLFDNULL).
Possible Error Effect:	Either the QLLCB is not usable, or is preventing other links from using the frame descriptors it owns. Either case can cause resource problems and so degrade the QLPS network.
Corrective Action Taken:	The audit corrects all of the QLLCB fields to the default values described in the error description. Lost frame descriptors are recovered by the FD audit [on the MH or quad gateway processor (QGP)] or the IFD audit [protocol handler (PH)].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	State of the QLLCB (QLLCB.state).
<i>Bad Data =</i>	Index of the lowest numbered non-idle timer (index into QLLCB.l2time[]).
<i>Good Data =</i>	The head of the frame descriptor list associated with QLLCB (QLLCB.f_ifd).
<i>Logical Index =</i>	QLLCB index (index into QLlcb[]).
<i>Dump =</i>	The QLLCB (entry in QLlcb[]).

Table 3 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The ttype and tidx fields failed the null value checks. If ttype is null, tidx should also be null and vice versa.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ttype field.
<i>Bad Data =</i>	The tidx field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 4 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between a quad linked packet switched logical link control block (QLLCB) and frame descriptors it appears to own is invalid. Specifically, one or more of the following is true: - For the frame descriptor indexed by QLLCB.f_ifd, either FD.llcb != the QLLCB's index or FD.back_ifd != QLFDNULL. - For the frame descriptor indexed by QLLCB.l_ifd, either FD.llcb != the QLLCB's index or FD.next_ifd != QLFDNULL. - For the frame descriptor indexed by QLLCB.ready, FD.llcb != the QLLCB's index. - For the frame descriptor indexed by QLLCB.blocked, FD.llcb != the QLLCB's index. In an MH or QGP, FDs are elements of CMfdusef[].
Possible Error Effect:	Frames on the link associated with this QLLCB are not transmitted properly.
Corrective Action Taken:	The audit sets the QLLCB FD fields (f_ifd , l_ifd , ready , and blocked) to QLFDNULL, and idles all FDs whose FD.llcb field = the QLLCB's index and (on MHs and QGPs) whose level 1 ownership bit is not set in FD.bf_owner.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Transmit queue count (QLLCB.xmit_q).
<i>Logical Index =</i>	QLLCB index (index into QLLcb[]).
<i>Dump =</i>	The QLLCB (entry in QLLcb[]).

Table 5 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	A QLLCB does not own any frame descriptors, but has a non-zero transmit queue count (QLLCB.xmit_q != 0).
Possible Error Effect:	QLPS frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the transmit queue count (QLLCB.xmit_q = 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The transmit queue count (QLLCB.xmit_q).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	QLLCB index (index into QLLcb[]).
<i>Dump =</i>	NA.

Table 6 Error Code: BDLINK

Error Code:	BDLINK

Error Description:	The linkage checks between MTLLCB and PCCB failed. The check is made between the MTLLCB index value and the sapi2llcb value in PCCB.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	The audit recovers this MTLLCB to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The chan field of MTLLCB.
<i>Bad Data =</i>	The teib_ix field of MTLLCB.
<i>Good Data =</i>	The sapi2llcb field of teib[] of PCCB.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 7 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	This QLLCB is being used for the QLPS application but one of the fields used for non QLPS application(MTLLCB), chan , teib_ix , link_id , sapi , ttype and tidx does not have default or null values.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The QLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ttype field.
<i>Bad Data =</i>	The link_id field.
<i>Good Data =</i>	The sapi field.
<i>Logical Index =</i>	The QLLCB index.
<i>Dump =</i>	The QLLCB being audited.

Table 8 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	The MTLLCB is idle, but it is not correctly initialized. One of the fields, state , f_ifd , l_ifd , ready , blocked , xmit_q , ttype , tidx and chan , does not have default value.
Possible Error Effect:	Possible resource failure.
Corrective Action Taken:	The audit corrects all the fields of the MTLLCB to the default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The link_id field.
<i>Bad Data =</i>	The chan field.
<i>Good Data =</i>	The state field.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 9 Error Code: IDLE_CODE

Error Code:	IDLE_CODE
Error Description:	One of the QLLCB fields, network , far_end , peerbusy_time and l2time does not have default values when it is used as MTLLCB for the second MH.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The network field.
<i>Bad Data =</i>	The far_end field.
<i>Good Data =</i>	The l2time_idx field.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 10 Error Code: INV

Error Code:	INV
Error Description:	The contents of the level 2 profile data (QLprof[]) are not correct.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit invokes level 2 functionality to calculate the correct level 2 profile

	values and correct them.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The profile structure (QLprof[]).

Table 11 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	For MH0, a QLLCB is in an invalid state (QLLCB.state is $\geq$ QLS_MAX), is in a state which requires a timer but does not have one (QLLCB.l2time[index] is idle), or is in a state which should not have a timer but does. For second MH, similar checks apply. Audit found the MTLLCB has a state field that indicates it has a timer but the ttype is not valid or the state field indicates there is no timer pending for this MTLLCB, but the ttype field is not null.
Possible Error Effect:	The endpoint associated with this QLLCB may be torn down, lose traffic, or lose capacity. For second MH, possible message failure on one logical link between SMP of SM-2000 and DNU-S.
Corrective Action Taken:	If QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmb[QLLCB.network.llcb_ix[QLLCB.far_end]] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished. For second MH, the MTLLCB will be recovered to the free pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Network associated with the QLLCB (QLLCB.network). For second MH, ttype field of MTLLCB.
<i>Bad Data</i> =	Endpoint associated with the QLLCB (QLLCB.far_end). For second MH, state field of MTLLCB.
<i>Good Data</i> =	State of the QLLCB (QLLCB.state). For second MH, chan field of MTLLCB.
<i>Logical Index</i> =	QLLCB index (index into QLlcb[]). For second MH, the MTLLCB index.
<i>Dump</i> =	The QLLCB (entry in QLlcb[]). For second MH, the MTLLCB being audited.

Table 12 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-idle QLLCB (QLLCB.network != SMQLPSNONETWK) does not have an active or expired timer (QLLCB.l2time[]), but appears to be linked in the QLLCB timeout list (that is, one or both of QLLCB.forw_llcb and QLLCB.back_llcb are not QLLIXNULL).
Possible Error Effect:	If the QLLCB is linked in the timeout list (owned by QLlcbhc[QLTIMELIST]) without a timer, operational code may assert when processing level 2 timeouts.
Corrective Action Taken:	The audit sets QLLCB.forw_llcb and QLLCB.back_llcb to QLLIXNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Network associated with the QLLCB (QLLCB.network).
<i>Bad Data</i> =	Endpoint associated with the QLLCB (QLLCB.far_end).
<i>Good Data</i> =	Forward QLLCB linkage in the QLLCB (QLLCB.forw_llcb).
<i>Logical Index</i> =	QLLCB index (index into QLlcb[]).

<i>Dump</i> =	The QLLCB (entry in QLLcb[]).

Table 13 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One or more of a QLLCB's frame descriptor ownership fields (f_ifd , l_ifd , ready , or blocked) contained an out-of-range value ($\neq$ QLFDNULL) and either > CM_NUMBUF (on an MH or QGP) or > PSRIFDS (on a PH).
Possible Error Effect:	Frames from or to the endpoint associated with the QLLCB are not transmitted properly.
Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to QLFDNULL. Any lost frame descriptors are picked up by the FD audit on an MH or QGP and by the IFD audit on a PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Transmit queue count (QLLCB.xmit_q).
<i>Logical Index</i> =	QLLCB index (index into QLLcb[]).
<i>Dump</i> =	The QLLCB (entry in QLLcb[]).

Table 14 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	One or more of the following is true. Note that "list" refers to the QLLCB free list (QLlcbhc[QLFREELIST]) if QLLCB.network = SMQLPSNONETWK. Otherwise, "list" refers to the QLLCB timeout list (QLlcbhc[QLTIMELIST]). - The QLLCB.forw_llcb is null ($\neq$ QLLIXNULL), but the QLLCB is not the tail of the list (QLHEADC.last $\neq$ QLLCB index). - The QLLCB.back_llcb is null ($\neq$ QLLIXNULL), but the QLLCB is not the head of the list (QLHEADC.first $\neq$ QLLCB index). - QLLCB.forw_llcb is not null ($\neq$ QLLIXNULL), and is either out of range ($\geq$ QLNLLCBS) or indexes a QLLCB which does not "point back" (QLlcb[QLLCB.forw_llcb].back_llcb $\neq$ QLLCB index). - QLLCB.back_llcb is not null ($\neq$ QLLIXNULL), and is either out of range ($\geq$ QLNLLCBS) or indexes a QLLCB which does not "point back" (QLlcb[QLLCB.back_llcb].forw_llcb $\neq$ QLLCB index). - The list is owned by QLlcbhc[QLFREELIST], and the QLLCB is linked to another QLLCB by either QLLCB.forw_llcb or QLLCB.back_llcb, but the two QLLCBs have different network values.
Possible Error Effect:	If the QLLCB is idle (QLLCB.network = SMQLPSNONETWK), the QLLCB free pool is broken. Link setup may fail due to a lack of resources. Otherwise, the QLLCB timeout list is broken. Level 2 timers may not be processed properly.
Corrective Action Taken:	If QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmbd[QLLCB.network.llcb_ix[QLLCB.far_end]] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	First element of the list (QLHEADC.first).
<i>Bad Data =</i>	Last element of the list (QLHEADC.first).
<i>Good Data =</i>	Network associated with the QLLCB (QLLCB.network).
<i>Logical Index =</i>	QLLCB index (index into QLlcb[]).
<i>Dump 1 =</i>	The QLLCB (entry in QLlcb[]).
<i>Dump 2 =</i>	The QLLCB indexed by QLLLCB.forw_llcb if forw_llcb is less than QLNLLCBS. Otherwise nothing. Note that the error address in the dump message will equal 1.
<i>Dump 3 =</i>	The QLLCB indexed by QLLLCB.back_llcb if back_llcb is less than QLNLLCBS. Otherwise nothing. Note that the error address in the dump message will equal 2.

Table 15 Error Code: OUTFRANGE

Error Code:	OUTFRANGE
Error Description:	Any one of the chan , teib_ix , link_id , tidx , sapi , ttype and state fields of the MTLLCB failed the range check.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ttype field.
<i>Bad Data =</i>	The link_id field.
<i>Good Data =</i>	The sapi field.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 16 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A non-idle QLLCB (QLLCB.network != SMQLPSNONETWK) is not linked to an LMDB. That is, one or more of the following is true: • QLLLCB.network > SMQLPSNUMNETS. • QLLLCB.far_end < SMQL1STENDPT or QLLCB.far_end > SMQLLSTENDPT. • QLlmbd[QLLCB.network].llcb_ix[QLLCB.far_end] != QLLCB index.
Possible Error Effect:	If there is a link associated with the QLLCB, the link is not able to transmit or receive frames. Otherwise, the QLLCB is a lost resource. If enough QLLCBs became lost, link set up could fail due to a lack of QLLCB resources.
Corrective Action Taken:	If QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmbd[QLLCB.network].llcb_ix[QLLCB.far_end] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Network associated with the QLLCB (QLLCB.network).
<i>Good Data =</i>	Endpoint associated with the QLLCB (QLLCB.far_end).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The QLLCB (entry in QLlcb[]).

Table 17 Error Code: SAPIBAD

Error Code:	SAPIBAD
Error Description:	The logical link ID, link_id , field of the MTLLCB is used to get the logical link address, tei and sapi values, and the sapi value of the MTLLCB is checked against this sapi value but the check failed.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The link_id field.
<i>Bad Data =</i>	The sapi field.
<i>Good Data =</i>	The local value of sapi obtained by using link_id .
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 18 Error Code: STATE

Error Code:	STATE
Error Description:	For MH0, there is an inconsistency among SMqpipestate[network], QLlmbd[network], SMqllstate[network][endpoint], and the QLLCB associated with the network and endpoint. The error report includes the contents of all of these structures. To determine the exact error, the data dumps must be examined. For second MH, the point to point check for logical link ID between PCCB and MTLLCB failed. Using chan and link_id fields of MTLLCB, audit obtained the PCCB and entry into link_stat[] array. A check between state field of MTLLCB and status field of link_stat[] array entry failed.
Possible Error Effect:	Varies according to the particular error. Some possibilities are: maintenance failures (that is, failures to remove or restore networks and/or endpoints), lost QLPS network traffic, asserts.
Corrective Action Taken:	The audit invokes the QLPS error handler for this network and endpoint with an appropriate error code (which is given in the good data field of the audit error report). For second MH, recover the MTLLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	QLLCB.network. For second MH, status field of link_stat[] array.
<i>Bad Data =</i>	QLLCB.far_end. For second MH, state field of MTLLCB.
<i>Good Data =</i>	The error type to be passed to the error handler for recovery. For second MH, chan field of MTLLCB.
<i>Logical Index =</i>	The QLLCB index (index into QLlcb[]). For second MH, the MTLLCB index.
<i>Dump 1 =</i>	The QLLCB (entry in QLlcb[]). For second MH, the MTLLCB being audited.
<i>Dump 2 =</i>	The link manager database (LMDB) (entry in QLlmbd[]), except for QLLMDB.lcb_ix[].

Table 19 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The audit failed the point to point back linkage checks between MTLLCB and the timer element. For the non null tidx value of MTLLCB, the timer element accessed by this tidx value does not point back to this MTLLCB.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The sidx field.
<i>Bad Data =</i>	The tidx field.
<i>Good Data =</i>	The stype field.

<i>Logical Index</i> =	The MTLLCB index.
<i>Dump</i> =	The MTLLCB being audited.

Table 20 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out of range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	For MH0, if the audit had been checking a valid QLLCB at the time of the interrupt, if QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmbd[QLLCB.network[.llcb_ix[QLLCB.far_end] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished. For second MH, if the MTLLCB index that was being audited at the time is in range, it will be recovered to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	In an MH or QGP, this is the value of SMmhrunning telling which MH process is running. In a PH, it is NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Internal record of the last QLLCB or MTLLCB index the audit was checking prior to the interrupt.
<i>Dump</i> =	NA.

QLLCB-B**Software Release:** 5E12 and later**1. QLPS LOGICAL LINK CONTROL BLOCK (QLLCB) AUDIT**

QLLCBs (QLPS logical link control blocks, elements of `QLlcb[ ]`) are used to represent layer 2 protocol information for a quad linked packet switch (QLPS) network endpoint. QLLCBs also own lists of frames associated at protocol layer 2 with the endpoint they represent.

The QLLCB audit also audits the message transport logical link control block (MTLLCB) in the second message handler (MH) of the digital networking unit-synchronous optical network (DNU-S). The DNU-S is a new peripheral for the 5ESS[®] switch that provides a cost of goods (COGS) reduction and synchronous optical network (SONET) capabilities. The second message handler provides all of the MH resident operational and maintenance software for establishing and maintaining communication paths using the peripheral control and timing (PCT) link between the SMP of SM-2000 and the DNU-S. Link access protocol for D-channel (LAPD) will be used to communicate to the DNU-S using the PCT links over three fixed timeslots as control to each common control (CC) of DNU-S.

The QLLCB audit runs in a segmented mode and is part of the MH0 and second MH base level loop.

2. DATA STRUCTURES

The following data is used for the QLLCB audit:

`QLlcb[ ]` = QLPS logical link control blocks

`QLlmbd[ ]` = QLPS link manager data blocks

`SMqpipestate[ ][ ]` = QPIPE state per network

In the second MH, the QLLCB audit checks the `MTlcb[ ]` array made up of MTLLCB data structure elements. Some of the fields audited out of this data structure are:

chan = Channel number.

link_id = Logical link ID.

sapi = SAPI value.

teib_ix = TEIB index.

tidx = Timer index.

ttype = Timer type.

3. AUDIT CHECKS

The following checks are performed by the QLLCB audit:

`BAD_HEAD_TAIL` = Check for QLLCB corrupted head cell.

`BADIDLE` = Check for uninitialized idle QLLCB.

`BADIDLE2` = Check `ttype` and `tidx` fields.

BADLINK	= Check for bad linkage between QLLCB and owned FD.
BDCOUNT	= Check QLLCB transmit queue count.
BDLINK	= Check linkage to PCCB.
BUSYIDLE	= Check for null MTLLCB fields.
CORRUPT_IDLE	= Check for default values for idle MTLLCB.
IDLE_CODE	= Check for null fields used by MH0 for QLPS.
INV	= Check contents of level 2 profile data.
INV_TIMER	= Check QLLCB (or MTLLCB) in second MH state against existence of timer.
NONNULL	= Check for non-idle QLLCB without an active or expired timer.
NULL3LINK	= Check QLLCB FD ownership fields for out-of-range values.
ONE2ONE	= Check linkages associated with the QLLCB.
OUTOFRANGE	= Range check the fields used by MTLLCB.
OWNLINK	= Verify that non-idle QLLCB is linked to an LMDB.
SAPI_BAD	= Check sapi field.
STATE	= Check for state consistency amongst internal data.
TIMERLNK	= Check linkage to timer element.
ZAP	= A reset was taken during the audit.

4. ERROR HANDLING

Different recovery is taken depending upon the error reported. Refer to the "Corrective Action Taken" section of the individual error.

The audit produces the following error code(s).

Table 1 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	A QLLCB head cell (quad linked packet switched network logical link control block, of type QLHEADC) is corrupted. Good data indicates which list was being checked. Specifically, it is not the case that both QLHEADC.first and QLHEADC.last are null (= QLLIXNULL), and one or more of the following is true: • Either the first or last field is null (= QLLIXNULL), but the other field is not. • One or both of the first and last fields are out-of-range (> QLLNLLCBS). • The back_llcb field of the QLLCB indexed by QLHEADC.first is not null (!= QLLIXNULL).

	The forw_llcb field of the QLLCB indexed by QLHEADC.last is not null ($\neq$ QLLIXNULL).
Possible Error Effect:	Good data indicates whether the list being checked was the QLLCB timing list or QLLCB free list. If the list being checked is the QLLCB timing list, level 2 timers on the quad linked packet switched (QLPS) links may not be processed properly. If the list being checked is the QLLCB free list, level 2 linkages may not be set up because of a resource failure. In either case, the QLPS network may be degraded.
Corrective Action Taken:	The audit sets QLHEADC.first = QLLIXNULL and QLHEADC.last = QLLIXNULL. If the list being checked is the free list, the audit goes on to reconstruct the list by linking all QLLCBs with QLLCB.network = SMQLPSNONETWK onto the list. QLLCBs which belong on the timing list are recovered by other checks in this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The list's first element (QLHEADC.first).
<i>Bad Data =</i>	The list's last element (QLHEADC.last).
<i>Good Data =</i>	AUTRUE indicates that QLLcbhc[QLFREELIST] was being checked. AUFALSE indicates that QLLcbhc[QLTIMELIST] was being checked.
<i>Logical Index =</i>	The default (null) network value (SMQLPSNONETWK).
<i>Dump =</i>	NA.

Table 2 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	A quad linked packet switched logical link control block (QLLCB) is idle (QLLCB.network = SMQLPSNONETWK), but is not correctly initialized. Specifically, one or more of the following is true: - QLLCB.state $\neq$ QLS_LD (the "link disconnect" state). - One or more entries of QLLCB.l2time[] are not idle (the QLLCB has a timer pending on it). - One or more of the QLLCB frame descriptor ownership fields is not null (f_ifd, l_ifd, ready or blocked are $\neq$ QLFDNULL).
Possible Error Effect:	Either the QLLCB is not usable, or is preventing other links from using the frame descriptors it owns. Either case can cause resource problems and so degrade the QLPS network.
Corrective Action Taken:	The audit corrects all of the QLLCB fields to the default values described in the error description. Lost frame descriptors are recovered by the FD audit [on the MH or quad gateway processor (QGP)] or the IFD audit [protocol handler (PH)].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	State of the QLLCB (QLLCB.state).
<i>Bad Data =</i>	Index of the lowest numbered non-idle timer (index into QLLCB.l2time[]).
<i>Good Data =</i>	The head of the frame descriptor list associated with QLLCB (QLLCB.f_ifd).
<i>Logical Index =</i>	QLLCB index (index into QLLcb[]).
<i>Dump =</i>	The QLLCB (entry in QLLcb[]).

Table 3 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The ttype and tidx fields failed the null value checks. If ttype is null, tidx should also be null and vice versa.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ttype field.
<i>Bad Data =</i>	The tidx field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 4 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between a quad linked packet switched logical link control block (QLLCB) and frame descriptors it appears to own is invalid. Specifically, one or more of the following is true: - For the frame descriptor indexed by QLLCB.f_ifd, either FD.llcb != the QLLCB's index or FD.back_ifd != QLFDNULL. - For the frame descriptor indexed by QLLCB.l_ifd, either FD.llcb != the QLLCB's index or FD.next_ifd != QLFDNULL. - For the frame descriptor indexed by QLLCB.ready, FD.llcb != the QLLCB's index. - For the frame descriptor indexed by QLLCB.blocked, FD.llcb != the QLLCB's index. In an MH or QGP, FDs are elements of CMfdusef[].
Possible Error Effect:	Frames on the link associated with this QLLCB are not transmitted properly.
Corrective Action Taken:	The audit sets the QLLCB FD fields (f_ifd , l_ifd , ready , and blocked) to QLFDNULL, and idles all FDs whose FD.llcb field = the QLLCB's index and (on MHs and QGPs) whose level 1 ownership bit is not set in FD.bf_owner.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Transmit queue count (QLLCB.xmit_q).
<i>Logical Index =</i>	QLLCB index (index into QLlcb[]).
<i>Dump =</i>	The QLLCB (entry in QLlcb[]).

Table 5 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	A QLLCB does not own any frame descriptors, but has a non-zero transmit queue count (QLLCB.xmit_q != 0).
Possible Error Effect:	QLPS frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the transmit queue count (QLLCB.xmit_q = 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The transmit queue count (QLLCB.xmit_q).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	QLLCB index (index into QLlcb[]).
<i>Dump =</i>	NA.

Table 6 Error Code: BDLINK

Error Code:	BDLINK

Error Description:	The linkage checks between MTLLCB and PCCB failed. The check is made between the MTLLCB index value and the sapi2llcb value in PCCB.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	The audit recovers this MTLLCB to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The chan field of MTLLCB.
<i>Bad Data =</i>	The teib_ix field of MTLLCB.
<i>Good Data =</i>	The sapi2llcb field of teib[] of PCCB.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 7 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	This QLLCB is being used for the QLPS application but one of the fields used for non QLPS application(MTLLCB), chan , teib_ix , link_id , sapi , ttype and tidx does not have default or null values.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The QLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The ttype field.
<i>Bad Data =</i>	The link_id field.
<i>Good Data =</i>	The sapi field.
<i>Logical Index =</i>	The QLLCB index.
<i>Dump =</i>	The QLLCB being audited.

Table 8 Error Code: CORRUPT_IDLE

Error Code:	CORRUPT_IDLE
Error Description:	The MTLLCB is idle, but it is not correctly initialized. One of the fields, state , f_ifd , l_ifd , ready , blocked , xmit_q , ttype , tidx and chan , does not have default value.
Possible Error Effect:	Possible resource failure.
Corrective Action Taken:	The audit corrects all the fields of the MTLLCB to the default values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The link_id field.
<i>Bad Data =</i>	The chan field.
<i>Good Data =</i>	The state field.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 9 Error Code: IDLE_CODE

Error Code:	IDLE_CODE
Error Description:	One of the QLLCB fields, network , far_end , peerbusy_time and l2time does not have default values when it is used as MTLLCB for the second MH.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The network field.
<i>Bad Data =</i>	The far_end field.
<i>Good Data =</i>	The l2time_idx field.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 10 Error Code: INV

Error Code:	INV
Error Description:	The contents of the level 2 profile data (QLprof[]) are not correct.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit invokes level 2 functionality to calculate the correct level 2 profile

	values and correct them.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The profile structure (QLprof[]).

Table 11 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	For MH0, a QLLCB is in an invalid state (QLLCB.state is $\geq$ QLS_MAX), is in a state which requires a timer but does not have one (QLLCB.l2time[index] is idle), or is in a state which should not have a timer but does. For second MH, similar checks apply. Audit found the MTLLCB has a state field that indicates it has a timer but the ttype is not valid or the state field indicates there is no timer pending for this MTLLCB, but the ttype field is not null.
Possible Error Effect:	The endpoint associated with this QLLCB may be torn down, lose traffic, or lose capacity. For second MH, possible message failure on one logical link between SMP of SM-2000 and DNU-S.
Corrective Action Taken:	If QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmbd[QLLCB.network[.llcb_ix[QLLCB.far_end]] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished. For second MH, the MTLLCB will be recovered to the free pool.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Network associated with the QLLCB (QLLCB.network). For second MH, ttype field of MTLLCB.
<i>Bad Data</i> =	Endpoint associated with the QLLCB (QLLCB.far_end). For second MH, state field of MTLLCB.
<i>Good Data</i> =	State of the QLLCB (QLLCB.state). For second MH, chan field of MTLLCB.
<i>Logical Index</i> =	QLLCB index (index into QLlcb[]). For second MH, the MTLLCB index.
<i>Dump</i> =	The QLLCB (entry in QLlcb[]). For second MH, the MTLLCB being audited.

Table 12 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	A non-idle QLLCB (QLLCB.network != SMQLPSNONETWK) does not have an active or expired timer (QLLCB.l2time[]), but appears to be linked in the QLLCB timeout list (that is, one or both of QLLCB.forw_llcb and QLLCB.back_llcb. are not QLLIXNULL).
Possible Error Effect:	If the QLLCB is linked in the timeout list (owned by QLlcbhc[QLTIMELIST]) without a timer, operational code may assert when processing level 2 timeouts.
Corrective Action Taken:	The audit sets QLLCB.forw_llcb and QLLCB.back_llcb to QLLIXNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Network associated with the QLLCB (QLLCB.network).
<i>Bad Data</i> =	Endpoint associated with the QLLCB (QLLCB.far_end).
<i>Good Data</i> =	Forward QLLCB linkage in the QLLCB (QLLCB.forw_llcb).
<i>Logical Index</i> =	QLLCB index (index into QLlcb[]).
<i>Dump</i> =	The QLLCB (entry in QLlcb[]).

Table 13 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One or more of a QLLCB's frame descriptor ownership fields (f_ifd , l_ifd , ready , or blocked) contained an out-of-range value (!= QLFDNULL and either > CM_NUMBUF (on an MH or QGP) or > PSRIFDS (on a PH)).
Possible Error Effect:	Frames from or to the endpoint associated with the QLLCB are not transmitted properly.
Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to QLFDNULL. Any lost frame descriptors are picked up by the FD audit on an MH or QGP and by the IFD audit on a PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Transmit queue count (QLLCB.xmit_q).
<i>Logical Index =</i>	QLLCB index (index into QLLcb[]).
<i>Dump =</i>	The QLLCB (entry in QLLcb[]).

Table 14 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	One or more of the following is true. Note that "list" refers to the QLLCB free list (QLlcbhc[QLFREELIST]) if QLLCB.network = SMQLPSNONETWK. Otherwise, "list" refers to the QLLCB timeout list (QLlcbhc[QLTIMELIST]). - The QLLCB.forw_llcb is null (= QLLIXNULL), but the QLLCB is not the tail of the list (QLHEADC.last != QLLCB index). - The QLLCB.back_llcb is null (= QLLIXNULL), but the QLLCB is not the head of the list (QLHEADC.first != QLLCB index). - QLLCB.forw_llcb is not null (!= QLLIXNULL), and is either out of range ($\geq$ QLNLLCBS) or indexes a QLLCB which does not "point back" (QLlcb[QLLCB.forw_llcb].back_llcb != QLLCB index). - QLLCB.back_llcb is not null (!= QLLIXNULL), and is either out of range ($\geq$ QLNLLCBS) or indexes a QLLCB which does not "point back" (QLlcb[QLLCB.back_llcb].forw_llcb != QLLCB index). - The list is owned by QLlcbhc[QLFREELIST], and the QLLCB is linked to another QLLCB by either QLLCB.forw_llcb or QLLCB.back_llcb, but the two QLLCBs have different network values.
Possible Error Effect:	If the QLLCB is idle (QLLCB.network = SMQLPSNONETWK), the QLLCB free pool is broken. Link setup may fail due to a lack of resources. Otherwise, the QLLCB timeout list is broken. Level 2 timers may not be processed properly.
Corrective Action Taken:	If QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmbd[QLLCB.network.llcb_ix[QLLCB.far_end]] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	First element of the list (QLHEADC.first).
<i>Bad Data =</i>	Last element of the list (QLHEADC.first).

<i>Good Data</i> =	Network associated with the QLLCB (QLLCB.network).
<i>Logical Index</i> =	QLLCB index (index into QLlcb[]).
<i>Dump 1</i> =	The QLLCB (entry in QLlcb[]).
<i>Dump 2</i> =	The QLLCB indexed by QLLLCB.forw_llcb if forw_llcb is less than QLNLLCBS. Otherwise nothing. Note that the error address in the dump message will equal 1.
<i>Dump 3</i> =	The QLLCB indexed by QLLLCB.back_llcb if back_llcb is less than QLNLLCBS. Otherwise nothing. Note that the error address in the dump message will equal 2.

Table 15 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	Any one of the chan , teib_ix , link_id , tidx , sapi , ttype and state fields of the MTLLCB failed the range check.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The ttype field.
<i>Bad Data</i> =	The link_id field.
<i>Good Data</i> =	The sapi field.
<i>Logical Index</i> =	The MTLLCB index.
<i>Dump</i> =	The MTLLCB being audited.

Table 16 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	A non-idle QLLCB (QLLCB.network != SMQLPSNONETWK) is not linked to an LMDB. That is, one or more of the following is true: • QLLLCB.network > SMQLPSNUMNETS. • QLLLCB.far_end < SMQL1STENDPT or QLLCB.far_end > SMQLLSTENDPT. • QLlmbd[QLLCB.network].llcb_ix[QLLCB.far_end] != QLLCB index.
Possible Error Effect:	If there is a link associated with the QLLCB, the link is not able to transmit or receive frames. Otherwise, the QLLCB is a lost resource. If enough QLLCBs became lost, link set up could fail due to a lack of QLLCB resources.
Corrective Action Taken:	If QLLCB.network is less than SMQLPSNUMNETS, and QLLCB.far_end is less than SMQLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmbd[QLLCB.network].llcb_ix[QLLCB.far_end] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Network associated with the QLLCB (QLLCB.network).
<i>Good Data</i> =	Endpoint associated with the QLLCB (QLLCB.far_end).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The QLLCB (entry in QLlcb[]).

Table 17 Error Code: SAPIBAD

Error Code:	SAPIBAD
Error Description:	The logical link id, link_id , field of the MTLLCB is used to get the logical link

	address, tei and sapi values, and the sapi value of the MTLLCB is checked against this sapi value but the check failed.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The link_id field.
<i>Bad Data =</i>	The sapi field.
<i>Good Data =</i>	The local value of sapi obtained by using link_id .
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 18 Error Code: STATE

Error Code:	STATE
Error Description:	For MH0, there is an inconsistency among SMqpipestate[network], QLlmbd[network], SMqllstate[network][endpoint], and the QLLCB associated with the network and endpoint. The error report includes the contents of all of these structures. To determine the exact error, the data dumps must be examined. For second MH, the point to point check for logical link ID between PCCB and MTLLCB failed. Using chan and link_id fields of MTLLCB, audit obtained the PCCB and entry into link_stat[] array. A check between state field of MTLLCB and status field of link_stat[] array entry failed.
Possible Error Effect:	Varies according to the particular error. Some possibilities are: maintenance failures (that is, failures to remove or restore networks and/or endpoints), lost QLPS network traffic, asserts. For second MH, possible message failure between SM and DNU-S.
Corrective Action Taken:	The audit invokes the QLPS error handler for this network and endpoint with an appropriate error code (which is given in the good data field of the audit error report). For second MH, recover the MTLLCB to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	QLLCB.network. For second MH, status field of link_stat[] array.
<i>Bad Data =</i>	QLLCB.far_end. For second MH, state field of MTLLCB.
<i>Good Data =</i>	The error type to be passed to the error handler for recovery. For second MH, chan field of MTLLCB.
<i>Logical Index =</i>	The QLLCB index (index into QLlcb[]). For second MH, the MTLLCB index.
<i>Dump 1 =</i>	The QLLCB (entry in QLlcb[]). For second MH, the MTLLCB being audited.
<i>Dump 2 =</i>	The link manager database (LMDB) (entry in QLlmbd[]), except for QLLMDB.lcb_ix[].

Table 19 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	The audit failed the point to point back linkage checks between MTLLCB and the timer element. For the non null tidx value of MTLLCB, the timer element accessed by this tidx value does not point back to this MTLLCB.
Possible Error Effect:	Possible message failure between SM-2000 SMP and DNU-S.
Corrective Action Taken:	The MTLLCB will be recovered and put on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The sidx field.
<i>Bad Data =</i>	The tidx field.
<i>Good Data =</i>	The stype field.
<i>Logical Index =</i>	The MTLLCB index.
<i>Dump =</i>	The MTLLCB being audited.

Table 20 Error Code: ZAP

Error Code:	ZAP

Error Description:	The audit takes this error as a result of a write protect violation, out of range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	For MH0, if the audit had been checking a valid QLLCB at the time of the interrupt, if QLLCB.network is less than SMLPSNUMNETS, and QLLCB.far_end is less than SMLNUMENDPTS, the audit takes the endpoint associated with this QLLCB OOS (out-of-service) by clearing the index in QLlmb[QLLCB.network[llcb_ix[QLLCB.far_end]] and releasing all resources associated with the QLLCB. The audit then sets all fields in the QLLCB to their default values and links it to the QLLCB free pool (owned by QLlcbhc[QLFREELIST]). Level 3 checks may later cause the endpoint to be reestablished. For second MH, if the MTLLCB index that was being audited at the time is in range, it will be recovered to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	In an MH or QGP, this is the value of SMMhrunning telling which MH process is running. In a PH, it is NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Internal record of the last QLLCB or MTLLCB index the audit was checking prior to the interrupt.
<i>Dump =</i>	NA.

QLNK

Software Release: 5E10 and later

1. QUAD-LINK PACKET SWITCH LINK (QLNK) AUDIT

The QLNK audit consists of checks which verify the information pertaining to the PHYSICAL QLNKs as well as information regarding the associated LOGICAL links. This audit is run in the SMKP environment and uses a two pass mechanism for all checks other than the internal state check.

2. DATA STRUCTURES

The QLNK audit makes reference to the following data structures:

SMqIm[][] = PCD information related to PHYSICAL QLNKs.

SMlIm[][] = PCD information related to LOGICAL links.

SMqImcst[] = Reflects number of ACT QLNKs.

CMnqIm[] = Reflects current link configuration.

RLMODATT = Stores MODATT static information per SM.

3. AUDIT CHECKS

The following checks are performed to audit the QLNK:

ZAP = A reset was taken during the audit.

CL_MODATT = DB failure on read of RLMODATT check.

UNEQUIP = Verify that a QLNK is truly OOSUNEQ with regards to its parents and the number of QGPs.

INT_STATE = Verify QLNK internal state is a valid one.

PARENT = Verify QLNK PCD state against its parent states.

LNK_CONFIG = Verify *nolink* bit is set properly in LLM entry.

ACT_PT_CL = Verify that SMqImcst[] value is correct and that physical link values point back properly in *lImbits*.

LEV2_QL = Verify that DDCPQL2 is set properly with respect to the states of the PHYSICAL QLNKs.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_PT_CL

Error Code:	ACT_PT_CL
Error Description:	Either the value in SMqImcst[] has been found to be inconsistent with the actual

	number of PHYSICAL QLNKs or the LLM entry is not pointing back to the correct PHYSICAL QLNK.
Possible Error Effect:	Incorrect QLNK information could be used.
Corrective Action Taken:	Either SMqlmcst[] is updated to the correct value or the LLM llmbits values are updated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0.
<i>Bad Data</i> =	SMqlmcst[] contents or Index into LLM entry.
<i>Good Data</i> =	Local calculation of SMqlmcst[] or 0.
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Audit's copy of the data read (QL).

Table 2 Error Code: CL_MODATT

Error Code:	CL_MODATT
Error Description:	DB failure on read of RLMODATT for this specified SM.
Possible Error Effect:	Data unavailable to perform QLNK checks.
Corrective Action Taken:	None.
Manual Action Required:	User needs to investigate the reason for DB read failure.
Dump Description:	
<i>Error Address</i> =	0.
<i>Bad Data</i> =	Database return value.
<i>Good Data</i> =	SM number used to read RLMODATT.
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Contents of buffer used to read RLMODATT.

Table 3 Error Code: INT_STATE

Error Code:	INT_STATE
Error Description:	The QLNK has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in switch maintenance kernel process (SMKP). Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the QLNK in a valid state and possibly requests a LEVEL2 restart of it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PHYSICAL QLNK number.
<i>Bad Data</i> =	QLNK state.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Audit's copy of the data read (QL).

Table 4 Error Code: LEV2_QL

Error Code:	LEV2_QL
Error Description:	The value of DDCPQL2 does not match the current states of the PHYSICAL QLNKs for a given SM.
Possible Error Effect:	Incorrect assumptions could be made using the DDCPQL2 data.
Corrective Action Taken:	The audit updates the DDCPQL2 key to be consistent with the QLNK states.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Either AU_DDOFFNORMAL (0xb) or AU_DDNORMAL (0xc).
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Audit's copy of the data read (QL).

Table 5 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The nolink bit in SMllm[] is incorrect for one of the following reasons: (1) Bit is not set for entries indexed greater than 192.

	(2) Bit is set when we have at least one ACT QLNK. (3) Bit is not set when we have no ACT QLNKs. (4) Bit is not set, but LLM entry is not pointing back to an ACT QLNK. (5) Bit is not set when SMqlmcsf[] is 0.
Possible Error Effect:	Incorrect use of links could occur causing failures.
Corrective Action Taken:	The audit will set the <i>nolink</i> bit to its correct value depending upon the conditions.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Link index.
<i>Good Data</i> =	AULLMGETS_NOLNK (4).
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Audit's copy of the data read (QL).

Table 6 Error Code: PARENT

Error Code:	PARENT
Error Description:	The QLNK has been found to be in an inconsistent state with regards to its parents. The parents of the QLNK are the MH QPIPE and QGP QPIPE.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the QLNK in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	PHYSICAL QLNK number.
<i>Bad Data</i> =	QLNK state.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Audit's copy of the data read (QL).

Table 7 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The QLNK was found to be in a state other than SMQLOOSUNEQ for one of the following cases: (1) Dependent upon the number of operational QGPs, the UNEQ'ed QLNKs can be determined. (2) If either of the parent MH or QGP QPIPEs are OOS-UNEQ-NULL, then the QLNK should be SMQLOOSUNEQ.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit updates the state of the QLNK to SMQLOOSUNEQ.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	QLNK index.
<i>Good Data</i> =	SMQLOOSUNEQ (6).
<i>Logical Index</i> =	SM number.
<i>Dump</i> =	Audit's copy of the data read (QL).

Table 8 Error Code: ZAP

Error Code:	ZAP

Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit clears SMAuditbit and returns.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUpcdpass.
<i>Bad Data</i> =	SMAuditbit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

QLPS

Software Release: 5E10 and later

1. QUAD-LINK PACKET SWITCH (QLPS) AUDIT

The QLPS audit consists of checks done for both the QLPS and QTMSLNK. This audit is run in the SMKP environment and uses a two pass mechanism for all checks other than the internal PCD state check.

2. DATA STRUCTURES

The QLPS audit makes reference to the following data structures:

SMdqtps[] = PCD information related to QLPS and QTMSLNK.

SMdlnk2pipe[][] = An array which tracks qtmslnk to pipe number translations.

RLQLPS = Stores QLPS and QTMSLNK static information.

3. AUDIT CHECKS

The following checks are performed to audit the QLPS:

ZAP = A reset was taken during the audit.

INT_STATE = Check for invalid QLPS PCD state with regards to QLPS equippage.

PARENT = Check for inconsistent QLPS state with parent ONTC.

MATE = Check for inconsistent QLPS state with its mate.

EXTER = Verify that if parent ONTC is ACT, then QLPS is not OOS-FE.

BADDATAR = Check for inconsistent *qtmslnk* values with static ODD.

HWSW_CONFIG = Verify SW state of QLPS with state as determined in HW regs.

The following checks are performed to audit the QTMSLNKs:

ZAP = A reset was taken during the audit.

OUTOFRANGE = Verify that qtmslnk number is within range for QTMSLNK.

BIT1MAP = Verify SMdlnk2pipe[] against QTMSLNK data.

INT_STATE = Check for invalid QTMSLNK PCD state with regards to QLPS equippage.

PARENT = Check for inconsistent QTMSLNK state with parent QLPS.

EXTER = Verify that if parent QLPS is ACT, then QTMSLNK is not OOS-FE.

INT_DATA_BAD = Verify QGL information stored in QTMSLNK PCD data.

HWSW_CONFIG = Verify SW QTMSLNK PCD state against CP and EQ table entries.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	The qtmslnk number in SMDqlps[][] was found to be incorrect with respect to static data in RLQLPS.
Possible Error Effect:	Incorrect link information could be used.
Corrective Action Taken:	The audit updates the qtmslnk to its' correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Index into <i>qtmslnk_num[]</i> in SMDqlps[][].
<i>Bad Data =</i>	QTMSLNK number as read from PCD.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	[ONTC side (0 or 1)]QLPS network (0 or 1)].
<i>Dump 1 =</i>	Audit's copy of the data read (ST).
<i>Dump 2 =</i>	Buffer containing contents of RLQLPS static data.

Table 2 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The bit in SMDlnk2pipe[] is set and should not be for this corresponding QTMSLNK.
Possible Error Effect:	Incorrect link information could be used.
Corrective Action Taken:	The audit will clear the appropriate bit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Flag indicating whether this QTMSLNK matches an entry in SMDqpipe[][] (DBYES or DBNO).
<i>Bad Data =</i>	Do both associated QPIPEs exist? (DBYES or DBNO)
<i>Good Data =</i>	Pipe value.
<i>Logical Index =</i>	Office network and timing complex (ONTC) side (0 or 1) or quad-link packet switch (QLPS) network (0 or 1).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 3 Error Code: EXTER

Error Code:	EXTER
Error Description:	If unit is QLPS, then it has been found to be out-of-service family of equipment (OOS-FE) when the parent ONTC is active (ACT). If unit is QTMSLNK, then it has been found to be OOS-FE when the parent QLPS is ACT.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in switch maintenance kernel process (SMKP). Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The state is written to OOS-AUTO-RMV for the QLPS, and a conditional restore of the QLPS is requested. If the error is related to the QTMSLNK, then the state is written to OOS-AUTO-RMV for the QTMSLNK, and an unconditional restore of the QTMSLNK is requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID (0x37 for QLPS and 0x39 for QTMSLNK).
<i>Bad Data =</i>	Unit's state (base[qual1]qual2).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 4 Error Code: HWSW_CONFIG

Error Code:	HWSW_CONFIG
Error Description:	The software (SW) state of the QLPS does not agree with the hardware (HW) state of the QLPS as reflected in the HW registers. For the QTMSLNK, the SW state of the QTMSLNK does not agree with the CP and EQ tables in the HW registers.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	A conditional restore is requested for either the QLPS or QTMSLNK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x37 for QLPS or 0x39 for QTMSLNK).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 5 Error Code: INT_DATA_BAD

Error Code:	INT_DATA_BAD
Error Description:	The QGL information associated to a particular QTMSLNK was found to be incorrect in SMDqlps[][].
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The QGL information is updated and an unconditional restore is requested for the QTMSLNK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x39 for QTMSLNK).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 6 Error Code: INT_STATE

Error Code:	INT_STATE
Error Description:	The QLPS or QTMSLNK has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x37 for QLPS, 0x39 for QTMSLNK).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 7 Error Code: MATE

Error Code:	MATE
Error Description:	The QLPS has been found to be in an inconsistent state with its mate QLPS.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x37 for QLPS).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 8 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The qtmslnk number retrieved from SMDqlps[][] was found to not be within the range of valid quad linked packet switch time multiplexed switch network control and timing I-link (QTMSLNK) numbers.
Possible Error Effect:	Incorrect information returned from SMddread().
Corrective Action Taken:	Related audits run to clean up this error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID [0x39 for QTMSLNK].
<i>Bad Data</i> =	Unit's state (base[qual1]qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 9 Error Code: PARENT

Error Code:	PARENT
Error Description:	The QLPS or QTMSLNK has been found to be in an invalid state with relation to its parents. For the QLPS, the parent is the ONTC. For the QTMSLNK, the parent is the QLPS.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the unit in a valid state with relation to its parent and mate unit (for QLPS).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID (0x37 for QLPS and 0x39 for QTMSLNK).
<i>Bad Data</i> =	Unit's state (base[qual1]qual2).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump</i> =	Audit's copy of the data read (ST).

Table 10 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit clears SMAuditbit and returns.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	AUpcdpass.
<i>Bad Data</i> =	SMAuditbit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

QMMLHG

Software Release: 5E10 and later

1. MULTI-LINE HUNT GROUP QUEUE MEMBER (QMMLHG) AUDIT

Modular queuing provides queuing for multi-line hunt groups and includes queuing for lines, call waiting lamp control, delay announcements, and queue overflow message registers. Lines may be queued to a multi-line hunt group when all the members of the hunt group are busy. The strategy used for queuing calls to a multi-line hunt group when all members of the hunt group are busy requires three audits to ensure the availability of the multi-line hunt group queues: the multi-line hunt group queue audit (QUOS), the multi-line hunt group queue owner audit (QOMLHG), and the multi-line hunt group queue member audit (QMMLHG). This audit, QMMLHG, checks the multi-line hunt group queue member relation (RLQM_MLHG) and ensures that tuples are available when needed.

The QMMLHG audit is part of the routing and terminal allocation (RTA) audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The audit checks the owner-member linkage between the dynamic multi-line hunt group queue owner relation (RLQO_MLHG) and the dynamic RLQM_MLHG. The RLQM_MLHG relation contains all the members that may be in a queue. Each member tuple has a key, a queuing terminal process identifier, quarantine state, and linkage fields. Figure 1 shows these data structures.

The members of the ISAT/LGISAT group are owned by the relation RLQO_ICI. The audit has to now deal with a shared linkage between RLQM_MLHG and either RLQO_MLHG or RLQO_ICI.

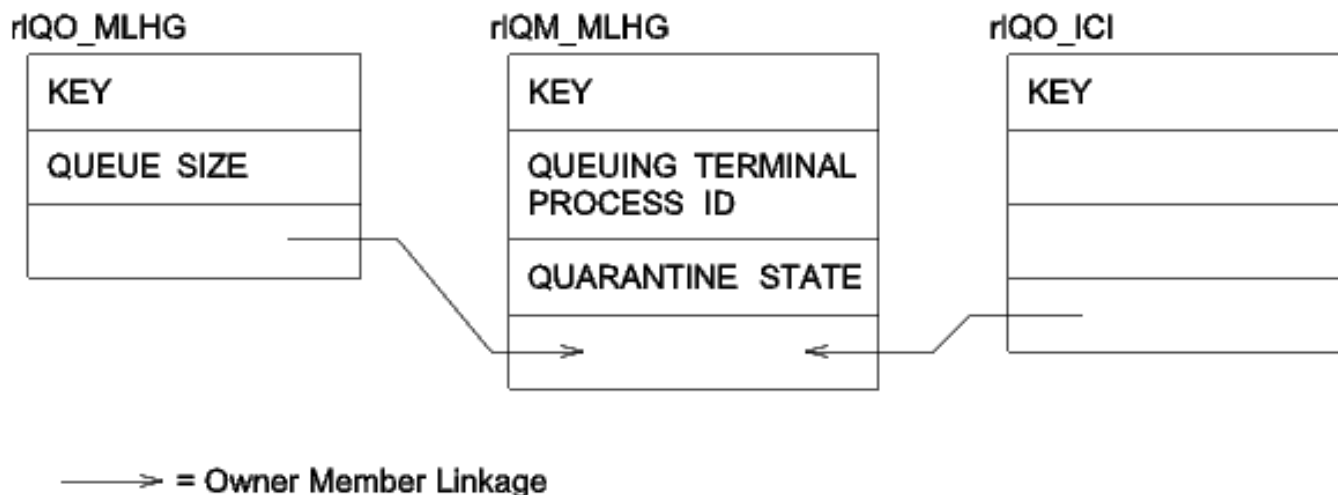


Figure 1 (QMMLHG-1) Data Structures of the QMMLHG Audit

3. AUDIT CHECKS

The audit invokes the indirect access sequential key driver (DRINASEK) to step through RLQM_MLHG. The audit performs existence checks on tuples 1 through N in RLQM_MLHG, where N is the value returned by the function RTqnummem. It corresponds to the number of tuples in RLQM_MLHG. For each non-vacant tuple, the audit performs syntactic checks by calling the AUTuple function. The checks include a check of the member-owner linkage between RLQM_MLHG and RLQO_MLHG.

If the syntactic checks pass, the semantic check function verifies that the queuing terminal process still exists for the queued tuple.

4. ERROR HANDLING

If the audit finds vacant tuples in RLQM_MLHG (gaps in the 1 through N sequence), it inserts them, links them to the idle list of RLQO_MLHG, and schedules the QU_OS and QO_MLHG audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The QMMLHG tuple is supposed to be in the idle list but the linkage has an error.
Possible Error Effect:	The QMMLHG tuple can not be used for call processing.
Corrective Action Taken:	The QMMLHG tuple is reconstructed and be put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the QMMLHG tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the QMMLHG tuple.
<i>Dump =</i>	Dump of the QMMLHG tuple.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	A queued QMMLHG tuple should be in the QO_MLHG owner linklist but is instead in the QO_ICI owner linklist.
Possible Error Effect:	The tuple remains queued to the wrong owner. The call will not be set up properly.
Corrective Action Taken:	The tuple will be put on the correct owner's list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the QMMLHG tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the QMMLHG tuple.
<i>Dump =</i>	Dump of the QMMLHG tuple.

Table 3 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A queued QMMLHG tuple is supposed to be idle but is not linked to the idle list.
Possible Error Effect:	The tuple QMMLHG is in limbo condition which is unusable by call processing.
Corrective Action Taken:	The tuple will be put on the correct owner's list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the QMMLHG tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the QMMLHG tuple.
<i>Dump =</i>	Dump of the QMMLHG tuple.

Table 4 Error Code: QUE

Error Code:	QUE
Error Description:	A queued QMMLHG tuple does not have an associated queuing terminal process.
Possible Error Effect:	The tuple remains queued until the resource being queued for becomes available.
Corrective Action Taken:	The tuple is idled.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the QMMLHG tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	riQM_MLHG tuple.

Table 5 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	A queued QMMLHG tuple should be in the QO_ICI owner linklist but is instead in the QO_MLHG owner linklist.
Possible Error Effect:	The tuple remains queued up to the wrong owner. The call will not be set up properly.
Corrective Action Taken:	The tuple will be put on the correct owner's list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the QMMLHG Tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the QMMLHG tuple.
<i>Dump</i> =	Dump of the QMMLHG tuple.

QMOGT

Software Release: 5E10 and later

1. OUTGOING TRUNK QUEUE MEMBER (QMOGT) AUDIT

The modular queuing capability supports offhook and ringback queuing for outgoing private facilities. Private facilities refer to either simulated facility groups (SFGs) or physical trunk groups (TGs). Requests for private facilities are assigned tuples in the trunk queue outgoing trunk member relation (RLQM_OGT). The QMOGT audit ensures that the resources of RLQM_OGT are not lost to call processing.

The QMOGT audit is part of the Feature Control audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The audit confines its checks to the RLQM_OGT relation. This data structure is shown in Figure 1.

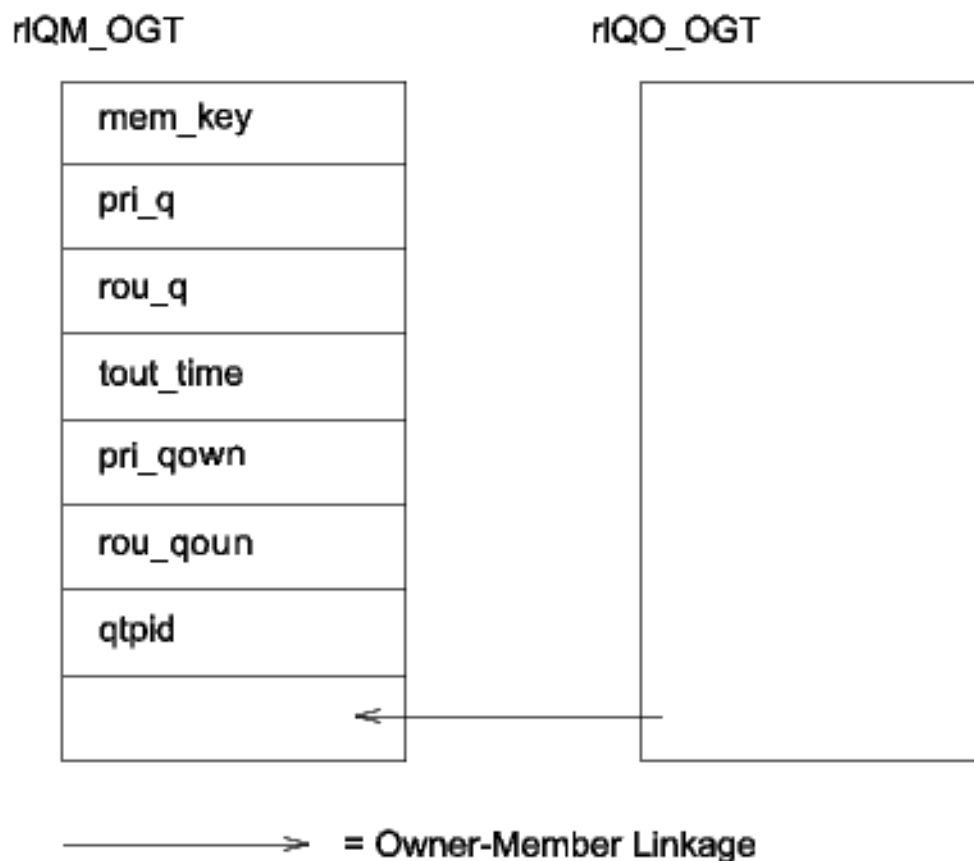


Figure 1 (QMOGT-1) Data Structures of the QMOGT Audit

3. AUDIT CHECKS

The audit invokes the indirect access sequential key driver (DRINASEK) to step through RLQM_OGT to perform existence checks, recover missing tuples, and remove nonvacant tuples with a key of zero. The syntactic check function ensures that tuples one through N exist, where "N" is a value obtained from the routing and terminal allocation function RTqnomem. For each nonvacant tuple found, the audit performs member linkage checks and invokes the semantic check function.

The semantic check function verifies the timer and priority fields, and ensures that offhook tuples on the queue have associated terminal processes. The semantic check function determines whether a tuple has timed out waiting for an outgoing trunk. It checks the routine and priority set owner linkages (ro_qown and pri_qown). When the routine set owner is on the idle list, the priority set owner linkage must be null.

4. ERROR HANDLING

If a syntactic check fails, the tuple is placed on the rIQM_OGT idle list. If a semantic check fails, the tuple is recovered and linked to the idle list.

The audit schedules the QUOGTOS and QOOGT audits as related audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: INV

Error Code:	INV
Error Description:	Routine set owner linkage is the idle list and the priority set owner linkage is not null.
Possible Error Effect:	Resources lost.
Corrective Action Taken:	Dynamic member tuple in rIQM_OGT is zeroed out and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of dynamic tuple.
<i>Bad Data =</i>	Member tuple's priority set owner linkage value.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQM_OGT tuple.

Table 2 Error Code: QMPRIORITY

Error Code:	QMPRIORITY
Error Description:	Routine or priority set owner linkage is invalid. The first three invalid cases deal with the member tuple linked to the priority queue. A priority error occurs when the tuple's priority set owner linkage is linked to the idle list or null, or when the tuple's routine set owner linkage is not null. The remaining cases deal with the member tuple linked to the routine queue. A priority error occurs when the tuple's priority set owner linkage is not null, or when the tuple's routine set owner linkage is null.
Possible Error Effect:	Resources lost.
Corrective Action Taken:	Dynamic member tuple in rIQM_OGT is zeroed out and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of dynamic tuple.
<i>Bad Data =</i>	Valid values are: 1 = A priority member tuple. 0 = A routine member tuple.
<i>Good Data =</i>	Member tuple's priority set owner linkage value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQM_OGT tuple.

Table 3 Error Code: QOFFHKTP

Error Code:	QOFFHKTP
Error Description:	Dynamic member tuple in rIQM_OGT does not have an associated terminal

	process.
Possible Error Effect:	The resource is stuck until dequeued.
Corrective Action Taken:	Dynamic member tuple in rIQM_OGT is zeroed out and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of dynamic tuple.
<i>Bad Data</i> =	Process number of member tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rIQM_OGT tuple.

Table 4 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	Dynamic member tuple in rIQM_OGT has been timed out for at least five minutes past timeout time.
Possible Error Effect:	Resources lost.
Corrective Action Taken:	Dynamic member tuple in rIQM_OGT is zeroed out and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of dynamic tuple.
<i>Bad Data</i> =	Member tuple timeout time in milliseconds plus five minutes.
<i>Good Data</i> =	Current time in milliseconds.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rIQM_OGT tuple.

QOICI

Software Release: 5E10 and later

1. OWNER RELATION FOR ISAT SUBQUEUES (QOICI) AUDIT

The QOICI audit performs existence checks between the static RElation rLICX_ICI and the dynamic relation RLQO_ICI. The audit also performs owner-member linkage checks between rIQO_ICI and rIQM_MLHG tuples. The QOICI uses the driver, DRINDEX, to step through the dynamic relation RLQO_ICI.

The dynamic relation RLQO_ICI is an index relation, keyed by isatici and contains backwards and forwards pointer for each ISAT subqueue. It also contains total number of calls de-queued for 30 minutes interval, total number of calls abandoned for 30 minutes interval, longest and average time a call on queue before being served and abandoned. For this capability, some of the queue cells in RIQM_MLHG are now linked to the RLQO_ICI relation instead of RLQO_MLHG.

2. DATA STRUCTURES

The RLQO_ICI relation is the owner relation for ISAT subqueues.

The QOICI audit is part of the AURtaque module. QOICI runs in the segmented mode and operate in the OSDS-M environment.

3. AUDIT CHECKS

The objective of the QOICI audit is to verify that for every tuple in the dynamic relation RLQO_ICI a corresponding tuple exists in the static relation rLICX_ICI. Any surplus rIQO_ICI tuples are deleted. The DRINDEX driver performs the key check, the quarantine bit check and the owner member linkage check.

4. ERROR HANDLING

When the audit encounters an error, an error message is printed on the ROP and an attempt is made to recover from an error.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an extra rIQO_ICI dynamic tuple. A relationship exists such that for every rIQO_ICI tuple there should be a corresponding rLICX_ICI static tuple where (key) rLICX_ICI.isat.idx = (key) rIQO_ICI.icx_ici.
Possible Error Effect:	Malfunction of ISAT MLHGs which are hunted and queued may take place.
Corrective Action Taken:	The extra rIQO_ICI dynamic tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Logical address of the rIQO_ICI tuple in error.
<i>Bad Data =</i>	The logical key of RLQO_ICI (isat_idx).
<i>Good Data =</i>	NA
<i>Logical Index =</i>	The logical key of RLQO_ICI (isat_idx).
<i>Dump =</i>	The contents of the rIQO_ICI tuple.

QOMLHG

Software Release: 5E10 and later

1. MULTI-LINE HUNT GROUP QUEUE OWNER (QOMLHG) AUDIT

Modular queuing provides queuing for multi-line hunt groups and includes queuing for lines, call waiting lamp control, delay announcements, and queue overflow message registers. Lines may be queued to a multi-line hunt group when all the members of the hunt group are busy. The strategy used for queuing calls to a multi-line hunt group when all members of the hunt group are busy requires three audits to ensure the availability of the multi-line hunt group queues: the multi-line hunt group queue audit (QUOS), the multi-line hunt group queue owner audit (QOMLHG), and the multi-line hunt group queue member audit (QMMLHG). This audit, QOMLHG, checks the multi-line hunt group queue owner relation (RLQO_MLHG) and ensures that hunt groups can be accessed when they are needed.

The QOMLHG audit is part of the routing and terminal allocation audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The audit performs existence checks between the static multi-line hunt group queue relation (RLQU_OS) and the dynamic RLQO_MLHG. The RLQO_MLHG relation is the owner relation for each assigned group which, when linked to the multi-line hunt group queue member relation (RLQM_MLHG), forms a queue. Each owner tuple has a key (the multi-line hunt group number plus one), the basic queue size, the extended queue size, the quarantine state, and linkage fields. Figure 1 shows these data structures.

RLQU_OS

KEY
BASIC QUEUE SIZE
EXTENDED QUEUE SIZE

RLQO_MLHG

KEY
BASIC QUEUE SIZE
EXTENDED QUEUE SIZE
QUARANTINE STATE

Figure 1 (QOMLHG-1) Data Structures of the QOMLHG Audit

3. AUDIT CHECKS

The audit invokes the index driver function (DRINDEX) to perform dynamic-to-static existence checks between the RLQO_MLHG and the RLQU_OS relations. These are the checks it makes:

- (a) If a tuple in RLQO_MLHG has a corresponding tuple in RLQU_OS, the audit proceeds to the next tuple.
- (b) If for a tuple in RLQO_MLHG the audit finds no corresponding tuple in RLQU_OS and recent change is not active on RLQU_OS, an error is printed and the tuple is deleted.

The semantic check function verifies that the basic queue size and the extended queue size in the rIQO_MLHG match those in the static relation rIQO_OS. This function corrects the queue size if they don't match.

4. ERROR HANDLING

If the audit finds extra tuples in RLQO_MLHG, it deletes them and schedules the QMMLHG audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BQSIZE

Error Code:	BQSIZE
Error Description:	Basic queue size in multi-line hunt group queue owner relation RLQO_MLHG contains incorrect value.
Possible Error Effect:	Multi-line hunt group queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Basic queue size is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad basic queue size value.
<i>Good Data</i> =	Good basic queue size value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rIQO_MLHG tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of queue monitor consoles in a tuple of the multi-line hunt group queue owner relation (RLQO_MLHG) contains an incorrect value.
Possible Error Effect:	A multi-line hunt group can be monitored by up to 3 queue monitor consoles. The queue threshold warning lamps for multi-line hunt groups have to be turned on or off on every monitor console monitoring the multi-line hunt group. If the number of queue monitor consoles is incorrect, threshold warning lamps for multi-line hunt groups would not be turned on or off on monitor consoles monitoring the multi-line hunt group.
Corrective Action Taken:	The number of queue monitor consoles is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad number of queue monitor consoles.
<i>Good Data</i> =	Good number of queue monitor consoles.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rIQO_MLHG tuple.

Table 3 Error Code: EXTQSIZE

Error Code:	EXTQSIZE
Error Description:	Extended queue size in multi-line hunt group queue owner relation rIQO_MLHG contains value.
Possible Error Effect:	Multi-line hunt group queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Extended queue size is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad extended queue size value.
<i>Good Data</i> =	Good extended queue size value.
<i>Logical Index</i> =	Key of tuple.

<i>Dump</i> =	rlQO_MLHG tuple.

Table 4 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	An MLHG queue is not monitored by any queue monitor console but the queue status counts are non-zero. The queue status counts are only used by queue monitor consoles. The queue status counts are tqcs (total queued calls served), tqca (total queued calls abandoned), ltoqbs (longest time on queue before being served), ltoqba (longest time on queue before being abandoned), ttoqbs (total time on queue before being served), and ttoqba (total time on queue before being abandoned).
Possible Error Effect:	MLHG queue status counts in RLQO_MLHG should be pegged only if the MLHG is monitored by at least one queue monitor console (num_qmon is non-zero). The queue status counts are invalid if they are pegged when the MLHG queue is not monitored by a queue monitor console.
Corrective Action Taken:	The queue status counts are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Number of queue monitor consoles.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_MLHG tuple.

Table 5 Error Code: WARNTHRESH

Error Code:	WARNTHRESH
Error Description:	The warning threshold value in multi-line hunt group queue owner relation RLQO_MLHG contains an incorrect value.
Possible Error Effect:	The queue threshold warning lamp for a multi-line hunt group is turned on if the number of calls in queue for multi-line hunt is greater than the value of warning threshold. If this value is in error then the queue threshold warning lamp is not turned on and off correctly.
Corrective Action Taken:	Warning threshold value is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad warning threshold value.
<i>Good Data</i> =	Good warning threshold value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_MLHG tuple.

QOOGT

Software Release: 5E10 and later

1. OUTGOING TRUNK QUEUE OWNER (QOOGT) AUDIT

The modular queuing capability supports offhook and ringback queuing for outgoing private facilities. Private facilities refer to either simulated facility groups (SFGs) or physical trunk groups (TGs). The QOOGT audit is needed to ensure that the static and dynamic data used in modular queuing are consistent.

The QOOGT audit is part of the Feature Control audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The QOOGT audit uses two relations: the outgoing trunk queue static relation (RLQU_OGTOS) and the outgoing trunk queue owner relation (RLQO_OGT). These data structures are shown in Figure 1.

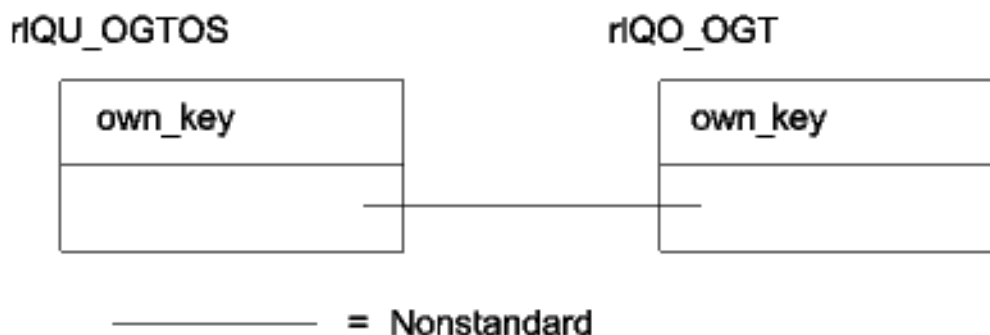


Figure 1 (QOOGT-1) Data Structures of the QOOGT Audit

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) to step through the tuples of `RLQO_OGT`. A dynamic-to-static existence check verifies that each tuple in `RLQO_OGT` has a corresponding tuple in `RLQU_OGTOS`.

If the syntactic checks pass, the audit's semantic check function checks the routine and priority queue sizes, priority flag, routine section type, post queue treatment, timeout limits, and routine and priority counts in `RLQO_OGT` against the data in `RLQU_OGTOS`. The routine queue count (`rq_count`) and priority queue count (`pg_count`) are checked against the number of members on each queue and against the number of members allowed. If recent changes are being made on `RLQU_OGTOS` when the audit runs, the audit does not report any semantic errors.

4. ERROR HANDLING

If there is not a corresponding static tuple in `RLQU_OGTOS`, the audit deletes the tuple in `RLQO_OGT`.

The QMOGT audit is scheduled as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP

Error Description:	Static data indicates the outgoing trunk queue owner relation RLQO_OGT should not have contained this tuple.
Possible Error Effect:	When assigning a outgoing trunk queue, an assert will fire.
Corrective Action Taken:	Dynamic tuple in RLQO_OGT is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of dynamic tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of dynamic tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 2 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	Group type and/or number in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the group type and number in the static relation RLQU_OGTOS.
Possible Error Effect:	Outgoing trunk queue member may be unnecessarily denied access to this queue.
Corrective Action Taken:	Group type and/or number in outgoing trunk queue owner relation RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect group type.
<i>Good Data</i> =	Correct group type.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 3 Error Code: POSTQ_TREAT

Error Code:	POSTQ_TREAT
Error Description:	Post queue treatment in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the post queue treatment in the static relation RLQU_OGTOS.
Possible Error Effect:	Outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Post queue treatment in outgoing trunk queue owner relation RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect post queue treatment value.
<i>Good Data</i> =	Correct post queue treatment value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 4 Error Code: PQCNT

Error Code:	PQCNT
Error Description:	Priority queue count in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the number of priority queue members counted.
Possible Error Effect:	Inability to queue a priority outgoing trunk queue member on the indicated outgoing trunk queue.
Corrective Action Taken:	Priority queue count in the outgoing trunk queue owner relation is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect priority queue count value.
<i>Good Data</i> =	Number of priority members counted.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 5 Error Code: PQMAXCNT

Error Code:	PQMAXCNT
Error Description:	Number of priority members counted on the outgoing trunk queue exceeds the priority queue size.
Possible Error Effect:	Priority outgoing trunk queue members may be unnecessarily access to this queue.
Corrective Action Taken:	Initialize and recover rIQO_OGT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect priority queue count value.
<i>Good Data =</i>	Maximum priority queue size value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 6 Error Code: PQSIZE

Error Code:	PQSIZE
Error Description:	Priority queue size in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the priority queue size in RLQU_OGTOS.
Possible Error Effect:	Priority outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Priority queue size in the outgoing trunk queue owner relation is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect priority queue size value.
<i>Good Data =</i>	Correct priority queue size value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 7 Error Code: PQTIMEOUT

Error Code:	PQTIMEOUT
Error Description:	Priority queue timeout limit in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the priority queue timeout limit in the static relation RLQU_OGTOS.
Possible Error Effect:	Outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Priority queue timeout limit in outgoing trunk queue owner relation RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect priority queue timeout limit value.
<i>Good Data =</i>	Correct priority queue timeout limit value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 8 Error Code: QOPRIORITY

Error Code:	QOPRIORITY
Error Description:	Priority in outgoing trunk queue owner relation RLQO_OGT does not correspond with the priority in the static relation RLQU_OGTOS.
Possible Error Effect:	Priority outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Priority in outgoing trunk queue owner relation RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect priority value.
<i>Good Data =</i>	Correct priority value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 9 Error Code: RETRYTIME

Error Code:	RETRYTIME
Error Description:	Retry timeout limit in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the retry timeout limit in the static relation RLQU_OGTOS.
Possible Error Effect:	Outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Retry timeout limit in outgoing trunk queue owner relation rIQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect retry timeout limit value.
<i>Good Data =</i>	Correct retry timeout limit value.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 10 Error Code: RQBUSYCNT

Error Code:	RQBUSYCNT
Error Description:	Routine queue busy count in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the number of busy queue members counted.
Possible Error Effect:	Inability to queue a routine outgoing trunk queue member on the indicated outgoing trunk queue.
Corrective Action Taken:	Routine queue busy count in the outgoing trunk queue owner relation is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect routine queue busy count value.
<i>Good Data =</i>	Number of busy routine members counted.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 11 Error Code: RQCNT

Error Code:	RQCNT
Error Description:	Routine queue count in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the number of queue members counted.
Possible Error Effect:	Inability to queue a routine outgoing trunk queue member on the indicated outgoing trunk queue.
Corrective Action Taken:	Routine queue count in the outgoing trunk queue owner relation is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect routine queue count value.
<i>Good Data =</i>	Number of routine members counted.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

Table 12 Error Code: RQMAXCNT

Error Code:	RQMAXCNT
Error Description:	Number of routine members counted on the outgoing trunk queue exceeds the routine queue size.
Possible Error Effect:	Routine outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Initialize and recover rIQO_OGT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address of tuple in error.
<i>Bad Data =</i>	Incorrect routine queue count value.
<i>Good Data =</i>	maximum routine queue size value.

<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 13 Error Code: RQSIZE

Error Code:	RQSIZE
Error Description:	Routine queue size in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the routine queue size in RLQU_OGTOS.
Possible Error Effect:	Routine outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Routine queue size in the outgoing trunk queue owner relation is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect routine queue size value.
<i>Good Data</i> =	Correct routine queue size value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 14 Error Code: RQTIMEOUT

Error Code:	RQTIMEOUT
Error Description:	Routine queue timeout limit in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the routine queue timeout limit in the static relation RLQU_OGTOS.
Possible Error Effect:	Outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Routine queue timeout limit in outgoing trunk queue owner relation RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect routine queue timeout limit value.
<i>Good Data</i> =	Correct routine queue timeout limit value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 15 Error Code: RSECTYP

Error Code:	RSECTYP
Error Description:	Routine section type in outgoing trunk queue owner relation RLQO_OGT does not correspond with the routine section type in the static relation RLQU_OGTOS.
Possible Error Effect:	Routine outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Routine section type in outgoing trunk queue owner relation RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect routine section type value.
<i>Good Data</i> =	Correct routine section type value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

Table 16 Error Code: SPTIME

Error Code:	SPTIME
Error Description:	Service protection time delay in the outgoing trunk queue owner relation RLQO_OGT does not correspond with the service protection time delay in the static relation RLQU_OGTOS.
Possible Error Effect:	Outgoing trunk queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Service protection time delay in outgoing trunk queue owner relation

	RLQO_OGT is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Incorrect service protection time delay value.
<i>Good Data</i> =	Correct service protection time delay value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_OGT tuple.

28. QP - RS

QPIPE

Software Release: 5E10 and later

1. QUAD-LINK PACKET SWITCH PIPES (QPIPE) AUDIT

The QPIPE audit consists of checks done for both the MH and QGP QPIPEs. This audit is run in the SMKP environment and uses a two pass mechanism for all checks other than the internal PCD state check.

2. DATA STRUCTURES

The QPIPE audit makes reference to the following data structures:

SMdqppe[][]	= PCD information related to MH and QGP QPIPEs.
SMdlnk2pipe[][]	= An array which tracks qtmslnk and tmslnk to pipe number translations.
RLTMSPIPE relation	= stores QPIPE static information.
RLSPC relation	= stores QGP static information.

3. AUDIT CHECKS

The following checks are performed to audit the MH QPIPEs:

BADDATAR	= Verify data in SMdqppe[][] against the same in RLTMSPIPE.
BIT1MAP	= Verify SMdlnk2pipe[][] mapping with TMSLNK number in SMdqppe[][].
BIT2MAP	= Verify SMdlnk2pipe[][] mapping with QTMSLNK number in SMdqppe[][].
EXTER	= Verify QPIPE PCD state against DDRQISOL.
EXTER2	= Verify QPIPE PCD state against DDSQPIPE.
HWSW_3CONFIG	= Verify Timeslot mapping for QPIPE.
INT_STATE	= Verify QPIPE internal state is a valid one.
OUTOFRANGE	= Verify that corresponding QTMSLNK number is within range.
PCD_STATIC	= Verify MH pipe equipage.
PARENT	= Verify QPIPE PCD state against its' parent states.
RANGE	= Verify that channel for MH PIPE is not 0 or 31.
STATE	= Verify QPIPE state against RLTMSPIPE existence.
UNEQUIP	= Verify that QPIPE is truly in an OOS-UNEQ state.
ZAP	= A reset was taken during the audit.

The following checks are performed to audit the QGP QPIPEs:

BADDATAR	= Verify QPIPE data with regards to QGP equipage.

BIT1MAP	= Verify SMdlnk2pipe[][] mapping for the QTMSLNK.
EXTER	= Verify QPIPE PCD state against that in the QGP HW for special OOS-AUTO-FE case.
EXTER2	= Verify QPIPE PCD state against that in the QGP HW for all other cases where recovery consists of sending message to QLFR SMER.
HWSW_2CONFIG	= Verify QPIPE endpoint mapping.
HWSW_CONFIG	= Verify QPIPE PCD state against that in the QGP HW for all other cases where recovery consists of secondarily implicating the QGP.
INT_STATE	= Verify QPIPE internal state is a valid one.
OUTOFRANGE	= Verify QTMSLNK number is within range.
PARENT	= Verify QPIPE PCD state against its' parent states.
ZAP	= A reset was taken during the audit.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADDATAR

Error Code:	BADDATAR
Error Description:	For MH QPIPEs, one of the following data items, qtmslnk tmslnk hdchan pipetype qts.qtsnum , in SMdqppe[][] was found to be incorrect with respect to static data in RLTMSPIPE. For QGP QPIPEs, depending upon the QGP equipage, the values of one of the following data items, type channelno qtmslnknum , were found to be incorrect.
Possible Error Effect:	Incorrect pipe information could be used.
Corrective Action Taken:	The audit updates the above mentioned fields to their correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	0 or 1 (QGP QPIPEs) or unit's state (base qual1 qual2) (MH QPIPEs only).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	Office network and timing complex (ONTC) side (0 or 1) or quad-link packet switch (QLPS) network (0 or 1).
<i>Dump 1 =</i>	Audit's copy of the data read (ST).
<i>Dump 2 =</i>	Contents of SMdqppe[][] for this Endpoint and QLPS network.
<i>Dump 3 =</i>	Contents of static RLTMSPIPE entry (MH QPIPEs only).

Table 2 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	For MH QPIPE, the bit in SMdlnk2pipe[] is not set and should not be for the corresponding TMSLNK. For QGP QPIPE, the bit in SMdlnk2pipe[] is not set and should not be for the corresponding QTMSLNK.
Possible Error Effect:	Incorrect pipe information could be used.
Corrective Action Taken:	The audit will set the appropriate bit in SMdlnk2pipe[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).

<i>Bad Data</i> =	TMSLNK number for MH QPIPE and QTMSLNK number for QGP QPIPE.
<i>Good Data</i> =	Endpoint number.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1</i> =	Contents of SMDqpipe[] for this endpoint and QLPS network.
<i>Dump 2</i> =	Audit's copy of the data read (ST).

Table 3 Error Code: BIT2MAP

Error Code:	BIT2MAP
Error Description:	For MH QPIPE, the bit in Smdlnk2pipe[] is not set and should not be for the corresponding QTMSLNK.
Possible Error Effect:	Incorrect pipe information could be used.
Corrective Action Taken:	The audit will set the appropriate bit in Smdlnk2pipe[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID of QPIPE (0x3a).
<i>Bad Data</i> =	QTMSLNK number.
<i>Good Data</i> =	Endpoint number.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1</i> =	Contents of SMDqpipe[] for this endpoint and QLPS network.
<i>Dump 2</i> =	Audit's copy of the data read (ST).

Table 4 Error Code: EXTER

Error Code:	EXTER
Error Description:	For MH QPIPEs, the state of the QPIPE does not agree with the current value of DDISOL and DDRQISOL. For QGP QPIPEs, the QGP HW state of the QPIPE is ACT, but the SW state of the QPIPE is OOS-AUTO-FE (All other QGP QPIPE errors of this type are covered under the EXTER2 error).
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in switch maintenance kernel process (SMKP). Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	For MH QPIPEs, the state of the QPIPE will be updated to match the current value of DDRQISOL and/or DDISOL. For QGP QPIPEs, related audits will run to clean this error up.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID for QPIPE(0x3a).
<i>Bad Data</i> =	Unit's state (base qual1 qual2).
<i>Good Data</i> =	Endpoint number.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1</i> =	Contents of SMDqpipe[] for this endpoint and QLPS network.
<i>Dump 2</i> =	Audit's copy of the data read (ST).

Table 5 Error Code: EXTER2

Error Code:	EXTER2
Error Description:	For MH QPIPEs, the state of the QPIPE does not agree with the current value of DDSQPIPE. For QGP QPIPEs, the QGP HW state of the QPIPE does not agree with the SW state of the QPIPE. This error does not cover the case where HW is ACT but SW is OOS-AUTO-FE (this is covered under EXTER error).
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	A message is sent to SMER to perform either a PIPEUP or PIPEFLT (near or far) on the QPIPE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Unit ID for QPIPE(0x3a).
<i>Bad Data</i> =	Value of DDSQPIPE for MH QPIPE. Unit's state (base qual1 qual2) for QGP QPIPE.
<i>Good Data</i> =	Endpoint number.
<i>Logical Index</i> =	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1</i> =	Contents of SMDqpipe[] for this endpoint and QLPS network.

<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 6 Error Code: HWSW_CONFIG

Error Code:	HWSW_CONFIG
Error Description:	The QGP HW state of the QGP QPIPE does not agree with the SW state of the QPIPE. This error does not cover the case where HW is ACT but SW is OOS-AUTO-FE (this is covered under EXTER error).
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The QGP is secondarily implicated since it appears to be a problem with the QGP itself. This will bring the HW and SW QPIPE states in synch.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID for QPIPE(0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Contents of SMdqppef [j] for this endpoint and QLPS network.
<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 7 Error Code: HWSW_2CONFIG

Error Code:	HWSW_2CONFIG
Error Description:	For QGP QPIPE, the timeslot mapping check failed with respect to TMSCON's internal audit. If the timeslot mapping check passed, then the endpoints do not match.
Possible Error Effect:	Incorrect QPIPE endpoint mapping information.
Corrective Action Taken:	A message is sent to secondarily implicate the QLPS which will in turn restore the QGP QPIPE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID for QPIPE(0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Contents of SMdqppef [j] for this endpoint and QLPS network.
<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 8 Error Code: HWSW_3CONFIG

Error Code:	HWSW_3CONFIG
Error Description:	For MH QPIPEs, the channel maps and timeslot mapping does not agree with the HW.
Possible Error Effect:	Incorrect QPIPE information being used.
Corrective Action Taken:	A message is sent to secondarily implicate the QLPS and get the QPIPE restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID for QPIPE(0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Contents of SMdqppef [j] for this endpoint and QLPS network.
<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 9 Error Code: INT_STATE

Error Code:	INT_STATE
Error Description:	The QPIPE has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.

Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The (q)tmlnlnk number retrieved from SMDqpipe[][] was found to not be within the range of valid (Q)TMLSLNK numbers.
Possible Error Effect:	Incorrect information returned from SMddread().
Corrective Action Taken:	Related audits run to clean up this error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Contents of SMDqpipe[][] for this endpoint and QLPS network.
<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 11 Error Code: PCD_STATIC

Error Code:	PCD_STATIC
Error Description:	An operational SM-2000 has been found with no MH QPIPES (that is, no entries in rITMSPIPE where rITMSPIPE.qendpt = SM number) but QLPS is equipped for the office (Glgpseq = DBYES).
Possible Error Effect:	Without MH QPIPES the SM may go into CTS overload and may initialize because of the overload.
Corrective Action Taken:	None.
Manual Action Required:	Technicians must perform MH QPIPE growth procedures for this SM as defined in the Routine Operations and Maintenance Procedures manual (235-102-210).
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	SM number.
<i>Dump 1 =</i>	The rIMODATT tuple where rIMODATT.module = SM number.
<i>Dump 2 =</i>	Internal data used by the audit.

Table 12 Error Code: PARENT

Error Code:	PARENT
Error Description:	The QPIPE has been found to be in an invalid state with relation to its parents. For the MH QPIPE, the parents are the QTMSLNK and the TMSLNK. For the QGP QPIPE, the parent is the QGL.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit places the unit in a valid state with relation to its parent(s).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump =</i>	Audit's copy of the data read (ST).

Table 13 Error Code: RANGE

Error Code:	RANGE

Error Description:	For MH QPIPE, the hdchan field in RLTMSPICE is set to either 0 or 31. This is not allowed for MH QPIPEs.
Possible Error Effect:	Incorrect pipe information could be used.
Corrective Action Taken:	None.
Manual Action Required:	Static data RLTMSPICE needs to be corrected.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	hdchan field in RLTMSPICE.
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Audit's copy of the data read (ST).
<i>Dump 2 =</i>	Contents of SMdqppe[][] for this endpoint and QLPS network.

Table 14 Error Code: STATE

Error Code:	STATE
Error Description:	For MH QPIPE, we are unable to read the RLTMSPICE static relation, therefore the QPIPE state should be OOS-UNEQ-NULL. The QPIPE state is NOT OOS-UNEQ-NULL.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit will update the state of the QPIPE to OOS-UNEQ-NULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2).
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Contents of SMdqppe[][] for this endpoint and QLPS network.
<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 15 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	For MH QPIPE, either a growth situation exists where the QPIPE should be OOS-UNEQ-NULL and it isn't or DDSQPIPE reflects a UNEQ situation but the QPIPE state does not show this.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	The audit will update the state of the QPIPE to OOS-UNEQ-NULL for the growth situation. For the DDSQPIPE check the audit will only report this error since there is most likely a problem in the SM and SM audits will clean the inconsistency up.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Unit ID of QPIPE (0x3a).
<i>Bad Data =</i>	Unit's state (base qual1 qual2) for growth case. Value of DDSQPIPE for all other cases.
<i>Good Data =</i>	Endpoint number.
<i>Logical Index =</i>	ONTC side (0 or 1) or QLPS network (0 or 1).
<i>Dump 1 =</i>	Contents of SMdqppe[][] for this Endpoint and QLPS network.
<i>Dump 2 =</i>	Audit's copy of the data read (ST).

Table 16 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit clears SMAuditbit and returns.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	AUpcdpass.
<i>Bad Data</i> =	SMAuditbit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

QRYDB

Software Release: 5E10 and later

1. QUERY DATA BLOCK (QRYDB) AUDIT

The query data block (QRYDB) audit verifies the consistency of the query data block relation.

The audit will run in the routine and elevated segmented modes.

2. DATA STRUCTURES

The audit structure for QRYDB is shown in Figure 1 .

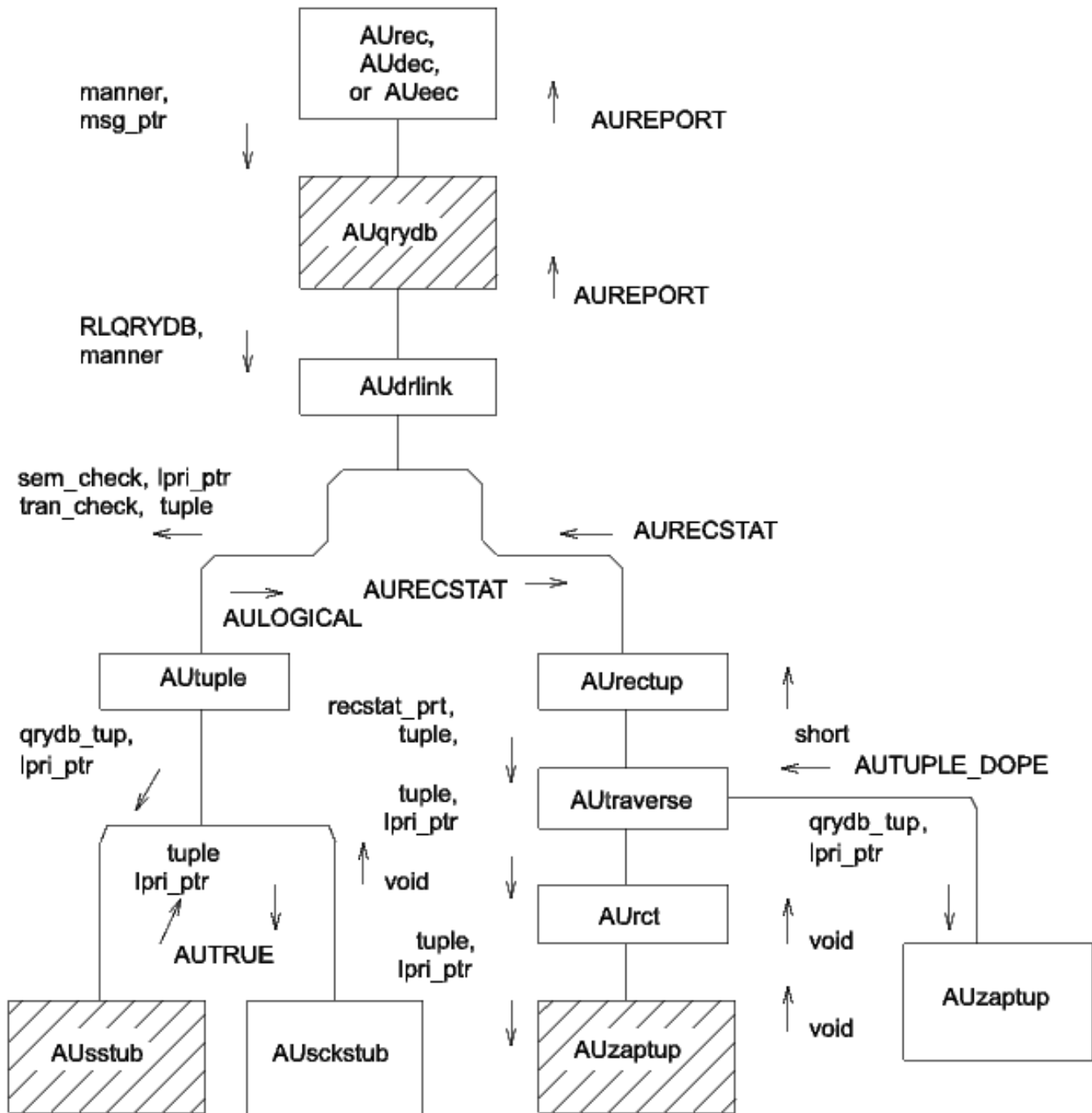


Figure 1 (QRYDB-1) QRYDB Audit Structure Chart

3. AUDIT CHECKS

The QRYDB audit calls the driver to step through the rIQRYDB tuples. The driver performs syntactic checks on the key field, the quarantine field and any linkage fields.

When an error is detected in a QRYDB tuple, the audit will recover the bad tuple and schedule the PTRDB audit to be run in the elevated mode.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes.

The audit produces the following error code(s).

Table 1 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	An idle rIQUERYDB tuple does not have all of its linkages marked idle. For idle rIQUERYDB tuples (tuples linked to the RLDB_HEADC idle list head cell tuple, where $rIQUERYDB.qry\_hdc.st\_ownk == DBQRYIDLE + DBSFDB\_HEA$), all other linkages should be marked to zero.
Possible Error Effect:	Incorrect billing and measurements for OSPS call operations.
Corrective Action Taken:	The rIQUERYDB tuple is recovered to the idle list (linked to the RLDB_HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIQUERYDB tuple.
<i>Bad Data =</i>	The contents of the particular linkage that was found to be marked to something other than zero.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIQUERYDB tuple (rIQUERYDB.procnum).
<i>Dump =</i>	The contents of the rIQUERYDB tuple.

Table 2 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	A rIQUERYDB tuple was found in an inconsistent state. If the tuple is idle, it should be linked to the RLDB_HEADC idle list head cell (rIQUERYDB.qry_hdc.st_ownk == DBQRYIDLE + DBSFDB_HEA). If the tuple is busy, this linkage should be zero, and there should be a one-to-one linkage with a rIPTRDB tuple (rIPTRDB.procnum == rIQUERYDB.qrptl.link - DBSFPTRDB). A tuple was found that does not meet either of these conditions.
Possible Error Effect:	Incorrect billing and measurements for OSPS call operations.
Corrective Action Taken:	The rIQUERYDB tuple is recovered to the idle list (linked to the RLDB_HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIQUERYDB tuple.
<i>Bad Data =</i>	The contents of the owner linkage to RLDB_HEADC (rIQUERYDB.qry_hdc.st_ownk - DBSFDB_HEA).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rIQUERYDB tuple (rIQUERYDB.procnum).
<i>Dump =</i>	The contents of the rIQUERYDB tuple.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A rIQUERYDB tuple was found marked busy, but had no linkage to a rIPTRDB tuple. If the rIQUERYDB tuple is idle, it should be linked to the RLDB_HEADC idle list head cell (rIQUERYDB.qry_hdc.st_ownk == DBQRYIDLE + DBSFDB_HEA). If the tuple is busy, this linkage should be zero, and there should be a one-to-one linkage with a rIPTRDB tuple (rIPTRDB.procnum == rIQUERYDB.qrptl.link - DBSFPTRDB). A tuple was found that does not meet either of these conditions.
Possible Error Effect:	Incorrect billing and measurements for OSPS call operations.
Corrective Action Taken:	The rIQUERYDB tuple is recovered to the idle list (linked to the RLDB_HEADC idle list head cell).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIQUERYDB tuple.
<i>Bad Data =</i>	The contents of the owner linkage to RLDB_HEADC (rIQUERYDB.qry_hdc.st_ownk

	- DBSFDB HEA).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rIQRYDB tuple (rIQRYDB.procnum).
<i>Dump</i> =	The contents of the rIQRYDB tuple.

QUOGTOS

Software Release: 5E10 and later

1. OUTGOING TRUNK QUEUE (QUOGTOS) AUDIT

The modular queuing capability supports offhook and ringback queuing for outgoing private facilities. Private facilities refer to either simulated facility groups (SFGs) or physical trunk groups (TGs). The QUOGTOS audit is needed to ensure that the static and dynamic data used in modular queuing are consistent.

The QUOGTOS audit is part of the feature control audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The relations the audit checks are the outgoing trunk queue relation (RLQU_OGTOS), a static relation, and the outgoing trunk queue owner relation (RLQO_OGT). These data structures are shown in Figure 1 .

rLQU_OGTOS

own_key
portgrp
rou_q
pri_q
rqsize
pgsize
rq_timeout
pq_timeout
retry_time
sp_time
post_q_treat

rLQO_OGT

own_key
portgrp
rou_q
pri_q
rqsize
pgsize
rq_timeout
pq_timeout
retry_time
sp_time
post_q_treat
pq_count
rq_count

———— = Nonstandard Linkage

Figure 1 (QUOGTOS-1) Data Structures of the QUOGTOS Audit

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) driver to step through the tuples of RLQU_OGTOS. The audit performs a static-to-dynamic existence check on each nonvacant tuple to ensure that there is a corresponding tuple in the dynamic rLQO_OGT relation. If a tuple does not exist in RLQO_OGT (that is, if the existence check fails), one is built and inserted in RLQO_OGT.

4. ERROR HANDLING

If there is not a corresponding dynamic tuple in RLQO_OGT for the static tuple in RLQU_OGTOS, the audit builds and inserts a tuple in RLQO_OGT.

The QMOGT audit is scheduled as a related audit.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Static data assignment indicates the outgoing trunk or facility queue owner relation RLQO_OGT should have an associated dynamic tuple.
Possible Error Effect:	Inability to queue a outgoing trunk queue member on the indicated outgoing trunk queue.
Corrective Action Taken:	The dynamic tuple in rIQO_OGT is created.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Absolute address where tuple should have been.
<i>Bad Data =</i>	DB return code from attempt to insert tuple, success = 0; failure < 0.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of tuple.
<i>Dump =</i>	rIQO_OGT tuple.

QUOS

Software Release: 5E10 and later

1. MULTI-LINE HUNT GROUP QUEUE (QUOS) AUDIT

Modular queuing provides queuing for multi-line hunt groups and includes queuing for lines, call waiting lamp control, delay announcements, and queue overflow message registers. Lines may be queued to a multi-line hunt group when all the members of the hunt group are busy. The strategy used for queuing calls to a multi-line hunt group when all members of the hunt group are busy requires four audits to ensure the availability of the multi-line hunt group queues: the multi-line hunt group queue audit (QUOS), the multi-line hunt group queue owner audit (QOMLHG), the multi-line hunt group queue member audit (QMMLHG), and the isat subqueue owner relation audit (QOICI). This audit, QUOS, checks the multi-line hunt group queue relation (RLQU_OS) and verifies the accuracy of the information in the dynamic multi-line hunt group queue owner relation (RLQO_MLHG), and the isat subqueue owner relation (RLQO_ICI).

The QUOS audit is part of the routing and terminal allocation audits subgroup. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The audit performs existence checks between the static RLQU_OS and the dynamic RLQO_MLHG. It also performs owner-member linkage checks between RLQO_MLHG and RLQM_MLHG. The RLQU_OS relation is keyed by the multi-line hunt group plus one; it includes the basic queue size and the extended queue size of the hunt group. Figure 1 shows these data structures.

Some queue calls are linked to the RLQO_ICI relation rather than the RLQO_MLHG relation. The QUOS audit performs existence checks between RLQU_OS and either RLQO_MLHG or RLQO_ICI relation.

rlQU_OS	rlQO_MLHG	rlQO_ICI
KEY	KEY	KEY
BASIC QUEUE SIZE	BASIC QUEUE SIZE	QUARANTINE STATE
EXTENDED QUEUE SIZE	EXTENDED QUEUE SIZE	
	QUARANTINE STATE	

Figure 1 (QUOS-1) Data Structures of the QUOS Audit

3. AUDIT CHECKS

With the new relation RLQO_ICI, the audit now has to first check the isat_idx in rLRT_MHG to see whether a rlQO_MLHG tuple should be the corresponding tuple or rlQO_ICI. If the isat_idx is non-zero then there should be a corresponding QOICI tuple.

- (a) If a tuple in RLQU_OS does not have a corresponding tuple in RLQO_ICI and recent change is not active on RLQU_OS, the audit reports an error and the tuple is inserted. The same is true for rlQO_MLHG tuples.

4. ERROR HANDLING

If when stepping through RLQU_OS the audit does not find a corresponding tuple in RLQO_MLHG, it creates one and schedules the QMMLMG audit.

If a semantic check fails, an error message is printed and the size is corrected from the static data.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BQSIZE

Error Code:	BQSIZE
Error Description:	Basic queue size in multi-line hunt group queue owner relation RLQO_MLHG contains incorrect value.
Possible Error Effect:	Multi-line hunt group queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Basic queue size is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad basic queue size value.
<i>Good Data</i> =	Good basic queue size value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_MLHG tuple.

Table 2 Error Code: EXTQSIZE

Error Code:	EXTQSIZE
Error Description:	Extended queue size in multi-line hunt group queue owner relation RLQO_MLHG contains incorrect value.
Possible Error Effect:	Multi-line hunt group queue members may be unnecessarily denied access to this queue.
Corrective Action Taken:	Extended queue size is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Absolute address of tuple in error.
<i>Bad Data</i> =	Bad extended queue size value.
<i>Good Data</i> =	Good extended queue size value.
<i>Logical Index</i> =	Key of tuple.
<i>Dump</i> =	rlQO_MLHG tuple.

Table 3 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	Static data assignment indicates the multi-line hunt dynamic relation RLQO_MLHG is missing a tuple, or dynamic relation RLQO_ICI is missing a tuple. For every tuple in static relation RLQU_OS there should exist a corresponding tuple in RLQO_MLHG where (key) rlQU_OS.grpnum = (key) rlQO_MLHG.mlhgnum. For every tuple in static relation RLICX_ICI, there should exist a corresponding tuple in RLQO_ICI, where (rlICX_ICI.ici[0 - 15] is not ICI_NULL, and rlICX_ICI.isat_idx << 4 & 0x3f0 X (where X is the index in rlICX_ICI.ici[X])) is equal to (key) rlQO_ICI.isatici. The relation ID for the missing tuple is dumped as bad data.
Possible Error Effect:	Inability to queue a multi-line hunt group member on the indicated multi-line hunt group's queue.
Corrective Action Taken:	The missing tuple is created.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Logical address where the missing tuple should have been.

<i>Bad Data</i> =	The relation ID of the missing tuple's relation (RLQO_MLHG, or RLQO_ICI).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the missing tuple.
<i>Dump</i> =	NA.

Table 4 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	Static data inconsistency between rIRT_MHG. Static data inconsistency between rICX_ICI.
Possible Error Effect:	Loss of resources and path hunt will be affected.
Corrective Action Taken:	None.
Manual Action Required:	User has to find nature of inconsistency and correct it. This error will continue until corrected.
Dump Description:	
<i>Error Address</i> =	(1) Logical address of the rIRT_MHG tuple. (2) Logical address of the rICX_ICI tuple.
<i>Bad Data</i> =	(1) Relation ID for rIRT_MHG. (2) Relation ID for rICX_ICI.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	(a) Key for RTRT_MHG relation. (b) Key for RLICX_ICI relation.
<i>Dump</i> =	NA.

QVSTATE

Software Release: 5E11 and later

1. QUEUING AND VECTORING STATE (QVSTATE) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. The queuing and vectoring data is stored in the RLQV_STATE (queuing and vectoring state) relation. Data in this relation is audited by the QVSTATE audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLQV_STATE relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The audit found a busy rIQV_STATE tuple, but the tuple is not linked to the proper DB head cell busy list.
Possible Error Effect:	This may result in call processing resources being lost.
Corrective Action Taken:	The rIQV_STATE tuple is recovered to the idle list defined in the RLDB_HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIQV_STATE tuple.
<i>Bad Data</i> =	The tuple's busy/idle owner (rISRVQDB.busyidle.st_ownk - DBSFDB_HEADC).
<i>Good Data</i> =	The value of the busy DB head cell owner (DBQVSTBUSY).
<i>Logical Index</i> =	The logical key of the rIQV_STATE tuple.
<i>Dump</i> =	The rIQV_STATE tuple.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The audit found an idle rIQV_STATE tuple, but the tuple is not linked to the idle DB head cell list.
Possible Error Effect:	This may result in call processing resources being lost.
Corrective Action Taken:	The rIQV_STATE tuple is recovered to the idle list defined in the RLDB_HEADC relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rIQV_STATE tuple.
<i>Bad Data</i> =	The tuple's busy/idle owner (rISRVQDB.busyidle.st_ownk - DBSFDB_HEADC).

<i>Good Data</i> =	The value of the idle DB head cell owner (DBQVSTIDLE).
<i>Logical Index</i> =	The logical key of the rQV_STATE tuple.
<i>Dump</i> =	The rQV_STATE tuple.

RCACSRQ

Software Release: 5E10 and later

1. RECENT CHANGE AUTOMATIC CUSTOMER STATION REARRANGEMENT QUEUE (RCACSRQ) AUDIT

The RCACSRQ tuple is used when a new request for automatic customer station rearrangement (ACSR) is received by the system. The RCACSRQ audit verifies the missing tuples in dynamic relation RLACSR_QC. The audit also verifies the dynamic global data structure RCacsr_jobs, and runs segmented in operational kernel process (OKP) environment.

2. DATA STRUCTURES

RLRCACSR_Q

RCACSR_JOBS

3. AUDIT CHECKS

The RCACSRQ audit steps through the RLRCACSR_Q relation, verifying missing tuples in the RLACSR_QC relation. At the end of the tuple level checking, the audit verifies the global dynamic data RCacsr_jobs.

4. ERROR HANDLING

If any missing tuples are found in the RLACSR_QC relation, those tuples are inserted. If any of the field is incorrect in RCacsr_jobs with the calculated value, it is corrected. A related audit ACSRQC is run to recover other data structures.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ACCPTR

Error Code:	ACCPTR
Error Description:	The nxt_num field (or the enqueueing field) of the global data structure RCacsr_jobs does not match the same value calculated by the audit in the special case when we have acsr request from anywhere in the queue till the end of the queue.
Possible Error Effect:	The effect of this error could be that the recent change OKP process that handles the enqueueing of the ACSR requests may fail to do so and will take a DB assert or error or both.
Corrective Action Taken:	The audit will report the error and will correct the field by its own calculated value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data =</i>	The value of the nxt_num found in the RCacsr_jobs.
<i>Good Data =</i>	The value of the nxt_num calculated by audit.
<i>Logical Index =</i>	0. There is only one such global dynamic data.
<i>Dump =</i>	None.

Table 2 Error Code: ACPTRBAD

Error Code:	ACPTRBAD
Error Description:	The nxt_num field (or the enqueueing field) of the global data structure RCacsr_jobs does not match the same value calculated by the audit.

Possible Error Effect:	The effect of this error could be that the recent change OKP process that handles the enqueueing of the ACSR requests may fail to do so and will take a DB assert or error or both.
Corrective Action Taken:	The audit will report the error and will correct the field by its own calculated value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data =</i>	The value of the nxt_num found in the RCacsr_jobs.
<i>Good Data =</i>	The value of the nxt_num calculated by the audit.
<i>Logical Index =</i>	0. There is only one such global dynamic data.
<i>Dump =</i>	None.

Table 3 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	The audit found the DDACSRQFULL key to be having a "normal" value when the value of audit running count showed that the queue was full.
Possible Error Effect:	The RC OKP process that enqueues the ACSR requests will try to enqueue an ACSR requests when it receives one even though the queue is full. This will result in error (DB error).
Corrective Action Taken:	The audit will report the error and will write the correct value of the DDACSRQFULL key.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. There is no address to be dumped.
<i>Bad Data =</i>	The return value of the DDREAD of the DDACSRQFULL key.
<i>Good Data =</i>	DDOFFNORMAL. This should have been the value of the DDACSRQFULL key.
<i>Logical Index =</i>	This field is used to dump the value of running count kept by the audits, AUtotal_jobs, as there is no key to be dumped.
<i>Dump =</i>	None.

Table 4 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The nxt_job field (or the dequeuing field) of the global data structure RCacsr_jobs was found to be out of range, that is having a value more than the maximum value allowed in the case where we have the ACSR queue full that is the value of the running count of the audit is equal to the value of the maximum number of the requests allowed.
Possible Error Effect:	The next time when RC UNIX [®] process tries to dequeue an ACSR request, it will encounter an DB error.
Corrective Action Taken:	The audit will report the error and will set the nxt_job value to the value of nxt_num after checking it to be within range.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data =</i>	The value of the nxt_job found in the RCacsr_jobs.
<i>Good Data =</i>	0. Any value between 0 through 120 would have been a good value. This field is used here to dump the value of the nxt_num field.
<i>Logical Index =</i>	This field is used to dump the value of running count kept by the audit.
<i>Dump =</i>	None.

Table 5 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The nxt_num field (or the dequeuing field) of the global data structure RCacsr_jobs was found to be out of range, that is having a value more than the maximum value allowed in the case where we have the ACSR queue full that is the value of the running count of the audit is equal to the value of the maximum number of the requests allowed. It should be noted here that the value of nxt_job was already found out of range before this error.

Possible Error Effect:	The next time when RC OKP process tries to enqueue an ACSR request, it will encounter a DB error.
Corrective Action Taken:	The audit will report the error and will set the <code>nxt_job</code> and the <code>nxt_num</code> values to be equal to 1.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the global dynamic data structure <code>RCacsr_jobs</code> .
<i>Bad Data =</i>	The value of the <code>nxt_num</code> found in the <code>RCacsr_jobs</code> .
<i>Good Data =</i>	0. Any value between 0 through 120 would have been a good value. This field is used here to dump the value of the <code>nxt_job</code> field.
<i>Logical Index =</i>	This field is used to dump the value of running count kept by the audit.
<i>Dump =</i>	None.

Table 6 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The <code>num_jobs</code> field of the global data structure <code>RCacsr_jobs</code> did not match the audit running count value.
Possible Error Effect:	The next time when RC OKP process tries to enqueue an ACSR request, it may or may not enqueue the request depending upon the false value of the <code>num_jobs</code> . If the false value of <code>num_jobs</code> is the maximum number of the ACSR requests allowed, then the <code>DDACSRQFULL</code> key may be set incorrectly.
Corrective Action Taken:	The audit will report the error and will set the <code>num_jobs</code> value equal to the value of running count of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the global dynamic data structure <code>RCacsr_jobs</code> .
<i>Bad Data =</i>	The value of the <code>num_jobs</code> found in the <code>RCacsr_jobs</code> .
<i>Good Data =</i>	<code>AUtotal_jobs</code> . The value of the audit's running count for total number of the ACSR requests in the queue.
<i>Logical Index =</i>	0. There is no key to be dumped.
<i>Dump =</i>	None.

Table 7 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	INValid STATE The audit found <code>DDACSRQFULL</code> key set even though the audit running count showed that the queue is not full.
Possible Error Effect:	No new ACSR requests will be allowed to put in the queue assuming the queue is full and they will be asked to try later.
Corrective Action Taken:	The audit will report the error and will set the <code>DDACSRQFULL</code> key to <code>DDNORMAL</code> value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. No address to be dumped.
<i>Bad Data =</i>	The return value of <code>DDREAD</code> of <code>DDACSRQFULL</code> key (should be <code>OFFNORMAL</code> or 1).
<i>Good Data =</i>	0. The good value of the <code>DDACSRQFULL</code> key, <code>DDNORMAL</code> .
<i>Logical Index =</i>	This field is used to dump the value of running count kept by the audit.
<i>Dump =</i>	None.

Table 8 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit did not find a dynamic tuple of type <code>rlACSR_QC</code> when it tried to access that tuple from the static <code>rlRCACSR_Q</code> tuple.
Possible Error Effect:	The result of the status query of an ACSR request may be erroneously replied as no request of that type exists.
Corrective Action Taken:	The audit will report the error and will insert the missing tuple by filling in the appropriate information from the static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the static tuple buffer.
<i>Bad Data =</i>	0. We do not know what caused the missing tuple.

<i>Good Data</i> =	epdn field of the static tuple which is the key for the dynamic tuple if it had existed.
<i>Logical Index</i> =	req_number. Key of the static tuple.
<i>Dump</i> =	1) The static tuple of type rIRCACSR_Q whose corresponding dynamic tuple is missing.

Table 9 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The audit found the value of <i>nxt_num</i> and <i>nxt_job</i> not equal when the value of <i>audit running count</i> showed that the queue was full.
Possible Error Effect:	The RC OKP process that enqueues the ACSR requests may try to enqueue an ACSR requests when it receives one even though the queue is full. This will result in error (DB error).
Corrective Action Taken:	The audit will report the error and will set the <i>nxt_num</i> value equal to the <i>nxt_job</i> value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the global data structure RCacsr_jobs.
<i>Bad Data</i> =	Any value, either <i>nxt_num</i> or the <i>nxt_job</i> is bad. We will dump the value of <i>nxt_job</i> here and the value of the <i>nxt_num</i> will be dumped in the <i>good data</i> field.
<i>Good Data</i> =	<i>nxt_num</i> field of the RCacsr_jobs data structure is dumped here. As long as both <i>nxt_job</i> and <i>nxt_num</i> have the same value, we should not see this error.
<i>Logical Index</i> =	This field is used to dump the value of running count kept by the audits, <i>AUtotal_jobs</i> , as there is no key to be dumped.
<i>Dump</i> =	None.

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The <i>nxt_job</i> field (or the dequeuing field) of the global data structure RCacsr_jobs was found to be out of range, that is having a value more than the maximum value allowed in the case where we have no ACSR requests pending to be served.
Possible Error Effect:	The next time when RC OKP process tries to enqueue an ACSR request, it will encounter a DB error.
Corrective Action Taken:	The audit will report the error and will correct both the <i>nxt_job</i> and the <i>nxt_num</i> fields to their initial values that is 0 and 1 respectively.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data</i> =	The value of the <i>nxt_job</i> found in the RCacsr_jobs.
<i>Good Data</i> =	0. Any value between 0 through 120 would have been a good value.
<i>Logical Index</i> =	This field is used to dump the value of <i>nxt_num</i> when we found the out of range <i>nxt_job</i> value as there is no key value to be dumped.
<i>Dump</i> =	None.

Table 11 Error Code: PTR1

Error Code:	PTR1
Error Description:	When there are no jobs in the queue, the <i>nxt_job</i> and <i>nxt_num</i> fields can have two different sets of values. If the audit is running right after any type of initialization, the value of <i>nxt_job</i> should be 0 and the value of <i>nxt_num</i> should be 1. But if at least a single request has been honored, both values will be pointing to the same value. When audit ran, it did not find either of the two cases.
Possible Error Effect:	There is a possibility of assert from the recent change if the pointers are not correct.
Corrective Action Taken:	The audit will report the error and will correct both the fields to its initialization values, that is 0 and 1.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data</i> =	The value of the nxt_num found in the RCacsr_jobs.
<i>Good Data</i> =	This field is used to dump the value of AUtotal_jobs which should be zero.
<i>Logical Index</i> =	This field is used to dump the value of nxt_job.
<i>Dump</i> =	None.

Table 12 Error Code: PTR3

Error Code:	PTR3
Error Description:	The nxt_job field (or the dequeuing field) of the global data structure RCacsr_jobs does not match the same value calculated by the audit in special cases like wrap around case and the case where no acsr requests were found after the nxt_num field value is found (or set).
Possible Error Effect:	The effect of this error could be that the recent change <i>UNIX</i> [®] process that handles the enqueueing of the ACSR requests may fail to do so and will take a DB assert or error or both.
Corrective Action Taken:	The audit will report the error and will correct the field by its own value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data</i> =	The value of the nxt_job found in the RCacsr_jobs.
<i>Good Data</i> =	The value of the nxt_job calculated by audit.
<i>Logical Index</i> =	0. There is only one such global dynamic data.
<i>Dump</i> =	None.

Table 13 Error Code: RPTRBAD

Error Code:	RPTRBAD
Error Description:	The nxt_job field (or the dequeuing field) of the global data structure RCacsr_jobs does not match the same value calculated by the audit.
Possible Error Effect:	The effect of this error could be that the recent change <i>UNIX</i> [®] process that handles the enqueueing of the ACSR requests may fail to do so and will take a DB assert or error or both.
Corrective Action Taken:	The audit will report the error and will correct the field by its own value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the global dynamic data structure RCacsr_jobs.
<i>Bad Data</i> =	The value of the nxt_job found in the RCacsr_jobs.
<i>Good Data</i> =	The value of the nxt_job calculated by audit.
<i>Logical Index</i> =	0. There is only one such global dynamic data.
<i>Dump</i> =	None.

RCATPS

Software Release: 5E10 and later

1. REMOTE MEASUREMENT SYSTEM CONTROL AREA TEST POSITION STATUS (RCATPS) AUDIT

The remote measurement system control area test position status (RCATPS) audit audits dynamic relation RLR_CATPS. RLR_CATPS is the owner of the RLR_TPSTAT relation.

2. DATA STRUCTURES

The RCATPS audit checks and verifies the RLR_CATPS dynamic relation.

3. AUDIT CHECKS

The linkage driver checks linkage and restores the key attribute and nulls out the tuple if linkage problems are found.

The semantic check performed involves determining if the value of the attended test position attribute is consistent with the RLR_TPSTAT relation and correcting the attribute, if necessary.

4. ERROR HANDLING

If there are linkage problems, a function is invoked to calculate the values for the set head key attribute and set tail key attribute in the set owner portion of the linkage area, and rewrite the tuple.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NCATP

Error Code:	NCATP
Error Description:	An rIR_CATPS tuple was found to have an incorrect value in the "attended test positions" attribute.
Possible Error Effect:	The tuple contains the wrong information.
Corrective Action Taken:	The attribute is updated with correct information.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_CATPS tuple.
<i>Bad Data</i> =	Actual value of "attended test positions" attribute.
<i>Good Data</i> =	Corrected value for same attribute.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_CATPS tuple.

Table 2 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	An rIR_CATPS tuple was found to have an incorrect value in the "master control area owner" attribute (catps_ptr→r_mcatps.st_ownk).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	Correct tuple information is restored via a recovery routine.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_CATPS tuple.
<i>Bad Data</i> =	Actual value of "master control area owner" attribute (catps_ptr→r_mcatps.st_ownk).
<i>Good Data</i> =	Corrected value for the same attribute described in Bad Data ((catps_ptr→ca_id / MCATPS NUCA) + DBSFR MCATP).

<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rlR CATPS tuple.

RCGTMR

Software Release: 5E11 and later

1. ROUTE CONGESTION TIMER LIST (RCGTMR) AUDIT

RCGTMR audit step through the RLRCGTMR (rout congestion timer list) relation. For each tuple, the audit checks the linkage integrity between the RLRCGTMR and RLDPC_STAT (destination point code status) relations and verifies how many destination point code (DPC) tuples are linked to RLRCGTMR.

2. DATA STRUCTURES

For details on the fields contained in the RLRCGTMR relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: COUNT_BAD

Error Code:	COUNT_BAD
Error Description:	The count in the current rIRCGTMR tuple of the rIDPC_STAT tuples owned by the rIRCGTMR tuple does not equal the count taken by the audit.
Possible Error Effect:	If rIRCGTMR.count is smaller than the actual number of DPCs, the DPC remains congested. If rIRCGTMR.count is larger than the actual number of DPCs, common channel signaling 7 (CCS7) code asserts because it is not able to find a DPC.
Corrective Action Taken:	The audit corrects the count in the current rIRCGTMR tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRCGTMR tuple.
<i>Bad Data</i> =	rIRCGTMR.count.
<i>Good Data</i> =	Count of member tuples taken by the audit.
<i>Logical Index</i> =	rIRCGTMR.rcgnum.
<i>Dump</i> =	NA.

RDTADATA

Software Release: 5E10 and later

1. REMOTE DIGITAL TEST ACCESS DATA (RDTADATA) AUDIT

The remote protocol access (RPA) feature facilitates the maintenance of ISDN interfaces in the independent and Bell Operating Company (BOC) environments, as well as the AT&T leased network environment. The feature provides the capability to monitor and interact with raw data streams. Using the RPA feature, these data streams can be remoted to a central location where sophisticated performance analysis, surveillance, and in some cases, participation in protocol exchanges may take place. The feature consists of two basic components:

- (1) A non-interruptive protocol monitoring function which enables service providers to non-intrusively view actual raw data bit streams from one or more basic rate interface (BRI) or primary rate interface (PRI) channels between the 5ESS[®] switch and customer premise equipment (CPE); and,
- (2) A split test access function which provides remote interaction with protocol exchanges on the PRI connections between CPE and 5ESS[®]/5ESS[®] switch, as well as on ISDN interswitch connections.

2. DATA STRUCTURES

The RLRDTA_DATA relation stores remote digital test access (RDTA) data in the operations kernel processor (OKP) pertaining to active sessions which are currently taking place in switching modules (SMs). This includes gport which is the port under test (key), and session which is the session number assigned to the test session (0 to TMMXSESRDTA). The rLRDTA_DATA tuples are created as testing sessions in office SMs are started, and are deleted as these sessions are stopped. The maximum number of rLRDTA_DATA tuples which may exist is equal to the maximum number of sessions which may be up simultaneously in an office (TMMXASESRDTA).

The TMrdta_ports[] global array is indexed by session number, and is used to map session numbers to ports. Since there may be up to TMMXSESRDTA different session numbers in an office, the array contains a maximum of TMMXSESRDTA + 1 entries with a maximum of TMMXASESRDTA entries populated at any given time. There exists a one-to-one correspondence between the RLRDTA_DATA relation and TMrdta_ports[] global array such that for rLRDTA_DATA.session = X in TMrdta_ports[X], rLRDTA_DATA.gport = TMrdta_ports[X].gport. These data structures are shown in Figure 1 .

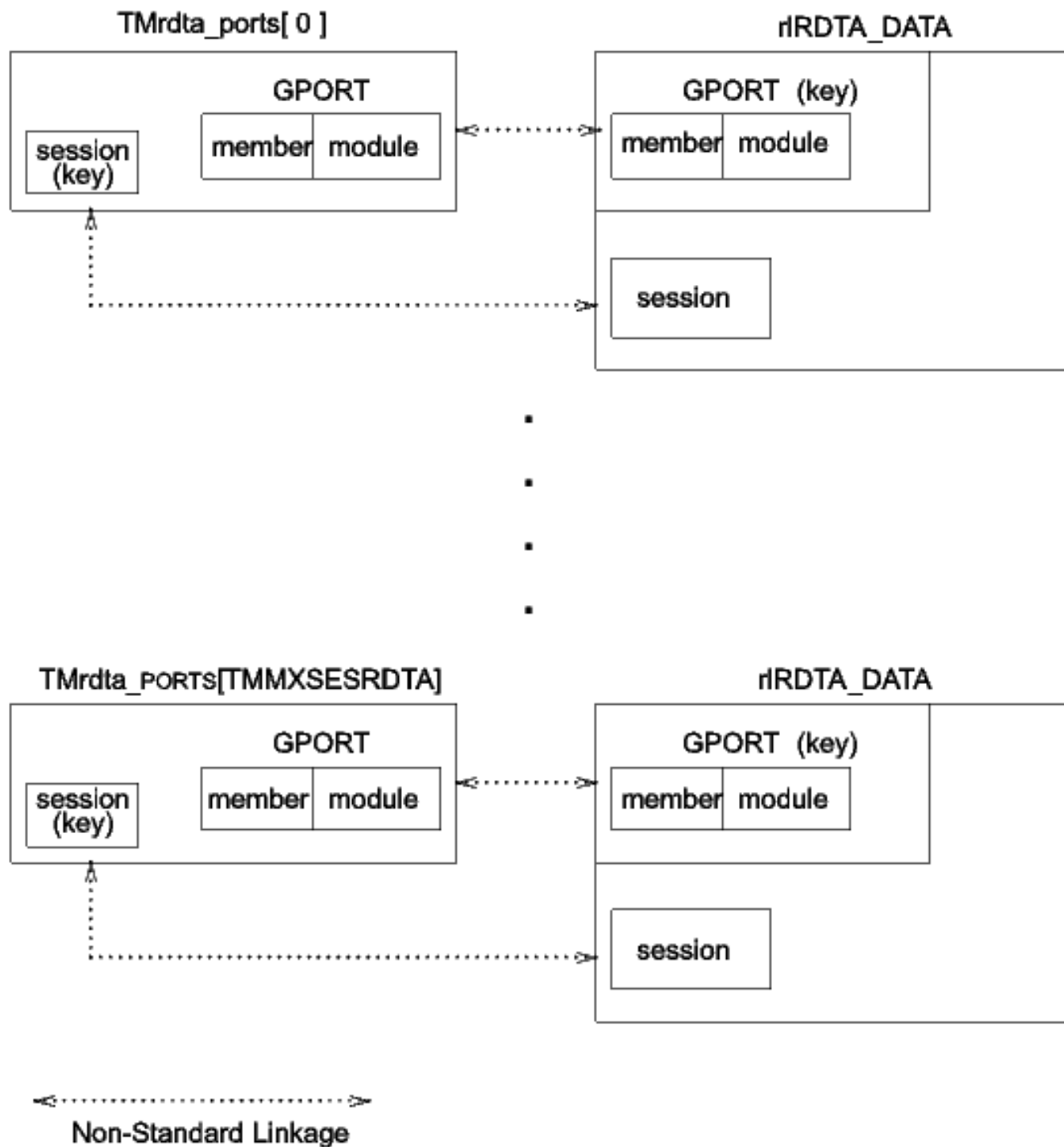


Figure 1 (RDTADATA-1)

3. AUDIT CHECKS

The RDTADATA audit uses the DRSTATIC driver, and makes these checks:

For every rRDTA_DATA tuple found:

GPORT = Check that the gport in the TMrda_ports[] array entry indexed by rIRDTA_DATA.session is equal to the session in the rIRDTA_DATA tuple (rIRDTA_DATA.gport).

OUTOFRANGE = Check that the session (rIRDTA_DATA.session) is validly in range.

For every entry in TMrda_ports[] up through TM_MXRDTASES in which TMrda_ports[].gport is not zero:

INDEX = Check that the session in the rIRDTA_DATA tuple (rIRDTA_DATA.session) equals the index of the TMrda_ports[] array entry currently under audit.

MISSTUP = Check that an rIRDTA_DATA tuple exists for this array entry, where TMrda_ports[].gport = (key) rIRDTA_DATA.gport.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: GPORT

Error Code:	GPORT
Error Description:	The gport (key) in an rIRDTA_DATA tuple did not match the gport in the corresponding TMrda_ports[] array entry. For the session in RLRDTA_DATA, when used as the index for the TMrda_ports[] array, there should exist a relationship such that rIRDTA_DATA.gport (key) equals TMrda_ports[session].gport, where session = rIRDTA_DATA.session, and is dumped as bad data.
Possible Error Effect:	The rIRDTA_DATA tuple no longer accurately represents the actual testing session taking place in the SM, if there is one, because the tuple may no longer be accessed by terminal maintenance RDTA activities without an array entry for the tuple also present to map a session number to a port under test.
Corrective Action Taken:	The rIRDTA_DATA tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIRDTA_DATA tuple.
<i>Bad Data =</i>	The session (rIRDTA_DATA.session) in the tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The first two bytes of the rIRDTA_DATA tuple's key, which contains the port that the invalid session was monitoring (gport.member). For the SM that the session was up on, refer to the second two bytes of the dump of the rIRDTA_DATA tuple.
<i>Dump 1 =</i>	The contents of the rIRDTA_DATA tuple.
<i>Dump 2 =</i>	The contents of the corresponding entry in TMrda_ports[session] where session equals rIRDTA_DATA.session.

Table 2 Error Code: INDEX

Error Code:	INDEX
Error Description:	The index (session number) for the TMrda_ports[] array entry under audit did not match the session field in the corresponding rIRDTA_DATA tuple. There exists a relationship such that, for all populated array entries (gport not equal to 0) in TMrda_ports[], session in TMrda_ports[session] equals rIRDTA_DATA.session for the rIRDTA_DATA tuple where TMrda_ports[session].gport equals rIRDTA_DATA.gport (key). Session is dumped as the logical key.
Possible Error Effect:	Up to TMMXASESRDTA session numbers should be available for assignment to test sessions. The session associated with the index to the array is unavailable for Remote Digital Test Access activities.

Corrective Action Taken:	The TMrda_ports[] array entry indexed by rIRDTA_DATA.session will be zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TMrda_ports[] array entry.
<i>Bad Data =</i>	The index (session) into the array.
<i>Good Data =</i>	The session in the rIRDTA_DATA tuple (rIRDTA_DATA.session). rIRDTA_DATA tuple (DBNO MATCH).
<i>Logical Index =</i>	The index (session) into the array.
<i>Dump 1 =</i>	The contents of the TMrda_ports[] global array.
<i>Dump 2 =</i>	The contents of the rIRDTA_DATA tuple where X in TMrda_ports[X] equals rIRDTA_DATA.session and TMrda_ports[X].gport equals rIRDTA_DATA.gport (key).

Table 3 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	A populated TMrda_ports[] array entry was found for which there was no corresponding rIRDTA_DATA tuple. For every populated TMrda_ports[] array entry (where TMrda_ports[].gport is not 0), there should exist a corresponding rIRDTA_DATA tuple, where TMrda_ports[].gport equals (key) rIRDTA_DATA.gport.
Possible Error Effect:	Up to TM_MXRDTASES session numbers should be available for assignment to test sessions. The session associated with the index to the array is unavailable for remote digital test access activities.
Corrective Action Taken:	The TMrda_ports[] array entry indexed by rIRDTA_DATA.session will be zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TMrda_ports[] array entry.
<i>Bad Data =</i>	The index (session) into the array.
<i>Good Data =</i>	The return from DB when an attempt was made to access a corresponding rIRDTA_DATA tuple (DBNO MATCH).
<i>Logical Index =</i>	The index (session) into the array.
<i>Dump =</i>	The contents of the TMrda_ports[] global array.

Table 4 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The field in an rIRDTA_DATA tuple which contains the session number corresponding to an RPA session (rIRDTA_DATA.session) was found out of range. The legal range for session numbers is 0 through TMMXSESRDTA.
Possible Error Effect:	The user will be unable to perform remote digital test access procedures on the port under test as specified by the rIRDTA_DATA tuple's key rIRDTA_DATA.gport, as long as the tuple exists, since tuple no longer corresponds to an valid test access session.
Corrective Action Taken:	The rIRDTA_DATA tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIRDTA_DATA tuple which contained the invalid session.
<i>Bad Data =</i>	The rIRDTA_DATA.session field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The first two bytes of the rIRDTA_DATA tuple's key, which contains the port that the invalid session was monitoring (gport.member). For the SM that the session was up on, refer to the second two bytes of the dump of the rIRDTA_DATA tuple.
<i>Dump =</i>	The contents of the rIRDTA_DATA tuple.

RDTCPDAT

Software Release: 5E12 and later

1. REMOTE DATA TERMINAL CALL PROCESSING DATA (RDTCPDAT) AUDIT

The RLRDTCPDAT relation contains information related to DS0/DS1 path hunting for TR-303 RDT. The RDTCPDAT audit verifies this data is consistent between circuit status, process status, and timeslot status for call processing use.

2. DATA STRUCTURES

For details on the fields contained in the RLRDTCPDAT relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The rLRDTCPDAT tuple is idle (linked to an idle rIPTSB tuple), but the data contained with the tuple indicates at least one timeslot is busy.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The busy timeslots are idled (set to 1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The DS1 number found to be in error.
<i>Bad Data =</i>	The DS0 found to be busy.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rLRDTCPDAT tuple (RDT unit number).
<i>Dump 1 =</i>	Dynamic rLRDTCPDAT tuple under audit.
<i>Dump 2 =</i>	rSMEST tuple.
<i>Dump 3 =</i>	rIPTSB tuple.

Table 2 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The rLRDTCPDAT tuple is busy (linked to an busy rIPTSB tuple), but the data contained with the tuple indicates at least one timeslot is idle OR the maintenance bit is set for the timeslot.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The idle timeslots are busied (set to 0). The maintenance bit is cleared (set to 1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The DS1 number found to be in error.
<i>Bad Data =</i>	The DS0 found to be idle.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rLRDTCPDAT tuple (RDT unit number).
<i>Dump 1 =</i>	Dynamic rLRDTCPDAT tuple under audit.
<i>Dump 2 =</i>	rSMEST tuple.
<i>Dump 3 =</i>	rIPTSB tuple.

Table 3 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	A static tuple read failed for either RLCKTDATA, RLF_VPIDB, RLLNK_PLI, RLTSP2APP, RLDS1_PT, or RLUNIT2RDT.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	NA.
Manual Action Required:	The missing tuples need to be inserted in the database.
Dump Description:	
<i>Error Address</i> =	The relation ID of the static relation accessed.
<i>Bad Data</i> =	The DS1 under audit.
<i>Good Data</i> =	The associated circuit name.
<i>Logical Index</i> =	Key of the rIRDTCPCDAT tuple (RDT unit number).
<i>Dump</i> =	Dynamic rIRDTCPCDAT tuple under audit.

Table 4 Error Code: P_ALWOOS

Error Code:	P_ALWOOS
Error Description:	The rIRDTCPCDAT tuple is busy (linked to an OOS rIPTSB tuple), but the data contained with the tuple indicates at least one timeslot is idle or its maintenance bit is not set.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The idle timeslots are busied (set to 0) and the maintenance bit is set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The DS1 number found to be in error.
<i>Bad Data</i> =	The DS0 found to be corrupted.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of the rIRDTCPCDAT tuple (RDT unit number).
<i>Dump 1</i> =	Dynamic rIRDTCPCDAT tuple under audit.
<i>Dump 2</i> =	rSMEST tuple.
<i>Dump 3</i> =	rIPTSB tuple.

Table 5 Error Code: RESL_ERROR

Error Code:	RESL_ERROR
Error Description:	The rIRDTCPCDAT tuple under audit does not contain the correct ds0_reserve value as derived from rIDS1_PT.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The correct value is written to ds0_reserve.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The DS1 number found to be in error.
<i>Bad Data</i> =	The ds0_reserve field found to be corrupted.
<i>Good Data</i> =	The derived ds0_reserve field value.
<i>Logical Index</i> =	Key of the rIRDTCPCDAT tuple (RDT unit number).
<i>Dump</i> =	Dynamic rIRDTCPCDAT tuple under audit.

Table 6 Error Code: SALW

Error Code:	SALW
Error Description:	The rIRDTCPCDAT tuple under audit is associated with an out-of-service DS1, but the associated ds1_alw bit in the tuple indicates this DS1 is allowed (1).
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The ds1_alw field is reset to inhibit (0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The DS1 number found to be in error.
<i>Bad Data</i> =	The ds1_alw bit found to be corrupted.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of the rIRDTCPCDAT tuple (RDT unit number).
<i>Dump 1</i> =	Dynamic rIRDTCPCDAT tuple under audit.
<i>Dump 2</i> =	Dynamic rSMEST tuple for the associated DS1.

Table 7 Error Code: SINH

Error Code:	SINH
Error Description:	The rIRDTCPDAT tuple under audit is associated with an active DS1, but the associated ds1_alw bit in the tuple indicates this DS1 is inhibited (0).
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The ds1_alw field is reset to allow (1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The DS1 number found to be in error.
<i>Bad Data =</i>	The ds1_alw bit found to be corrupted.
<i>Good Data =</i>	1.
<i>Logical Index =</i>	Key of the rIRDTCPDAT tuple (RDT unit number).
<i>Dump 1 =</i>	Dynamic rIRDTCPDAT tuple under audit.
<i>Dump 2 =</i>	Dynamic rSMEST tuple for the associated DS1.

Table 8 Error Code: SMEST

Error Code:	SMEST
Error Description:	An attempt to read rSMEST for a DS1 circuit failed.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	NA.
Manual Action Required:	Verify values if rIRT2DS1 are correct for DS1 circuit names.
Dump Description:	
<i>Error Address =</i>	The DS1 number found to be in error.
<i>Bad Data =</i>	The DS1 circuit name used to read rSMEST.
<i>Good Data =</i>	The associated RDT circuit name.
<i>Logical Index =</i>	Key of the rIRDTCPDAT tuple (RDT unit number).
<i>Dump =</i>	Dynamic rIRDTCPDAT tuple under audit.

Table 9 Error Code: TSI3_OPTS

Error Code:	TSI3_OPTS
Error Description:	The rIRDTCPDAT tuple under audit contains invalid values for TSI link or VPIDB.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The tsi_link and vpidb fields are set to their correct values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The DS1 number found to be in error.
<i>Bad Data =</i>	The value in the tsi_link field.
<i>Good Data =</i>	The tsi_link value derived from rLNK_PLI.
<i>Logical Index =</i>	Key of the rIRDTCPDAT tuple (RDT unit number).
<i>Dump 1 =</i>	Dynamic rIRDTCPDAT tuple under audit.
<i>Dump 2 =</i>	rLNK_PLI tuple.
<i>Dump 3 =</i>	rIF_VPIDB tuple.

Table 10 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The rIRDTCPDAT tuple under audit does not reflect the proper state for unequipped DS1 circuits. There could be a missing tuple in RLRT2DS1, or an existing tuple contains a null entry in rtds1, but the associated bits in rIRDTCPDAT are not set properly. Either the ds1_alw, ds0_status, or ds0_mtce bit are not set properly.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The ds1_alw, ds0_status, and ds0_mtce bits are set to correct values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	If missing tuple in RLRT2DS1, address of rIRDTCPDAT tuple. If null entry in rtds1, "1".
<i>Bad Data =</i>	The value in the ds1_alw field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIRDTCPDAT tuple (RDT unit number).
<i>Dump =</i>	Dynamic rIRDTCPDAT tuple under audit.

RELQ

Software Release: 5E10 and later

1. RELATION QUEUE (RELQ) AUDIT

The database manager program provides concurrency among transactions through multi-level locking (MLL) which includes relation level locking (RLL) and tuple level locking (TLL). Read-only transactions are not queued. Update transactions are queued if they have to access the same set of relations as other update transactions in RLL; or, the same set of tuples in TLL. The relation queue dictionary (RLDB_RELQ) has one entry for every relation in the system. When a transaction starts, the database manager creates an open dictionary entry for each relation needed by that transaction under RLL. The database manager also creates a database lock dictionary entry for each relation needed by that transaction. The open dictionary entry and the database lock dictionary entry, associated with the same relation and the same transaction, are linked together with a one-to-one link. While the transaction is in progress, if it needs some other relation, the database manager creates a database lock dictionary entry for that relation under TLL. In this case, there is no open dictionary entry created. The database lock dictionary entries are stored in a list for the needed relation. All database lock dictionary lists are owned by the relation queue dictionary. The relation queue audit (RELQ) checks the linkages between the relation queue dictionary and the database lock dictionary. BASE is the feature package name.

The RELQ audit is part of the database audits subgroup. This audit runs in the operational kernel process (OKP) in the administrative module (AM), in the operational system for distributed switching-switching module (OSDS-SM), and in the communications module processor (CMP) environments in the segmented mode.

2. DATA STRUCTURES

The RELQ audit checks the owner-member linkages between the relation queue dictionary and the database lock dictionary (RLDB_LOCK). Refer to Figure 1.

The system allocates one database lock dictionary block for each relation (under RLL or TLL) associated with a database transaction. When two or more transactions require access to the same relation, the database lock dictionary blocks are put in a list. The owner of the list is a tuple in RLDB_RELQ. Figure 1 shows that blocks 1 and 4 need access to r12; blocks 2, 5, and 7 need access to r18; blocks 3 and 6 need access to r15. It also shows that blocks 2, 4, and 7 are under RLL, which have associated open dictionary entries (r1DB_OPNDC); and, blocks 1, 3, 5 and 6 are under TLL.

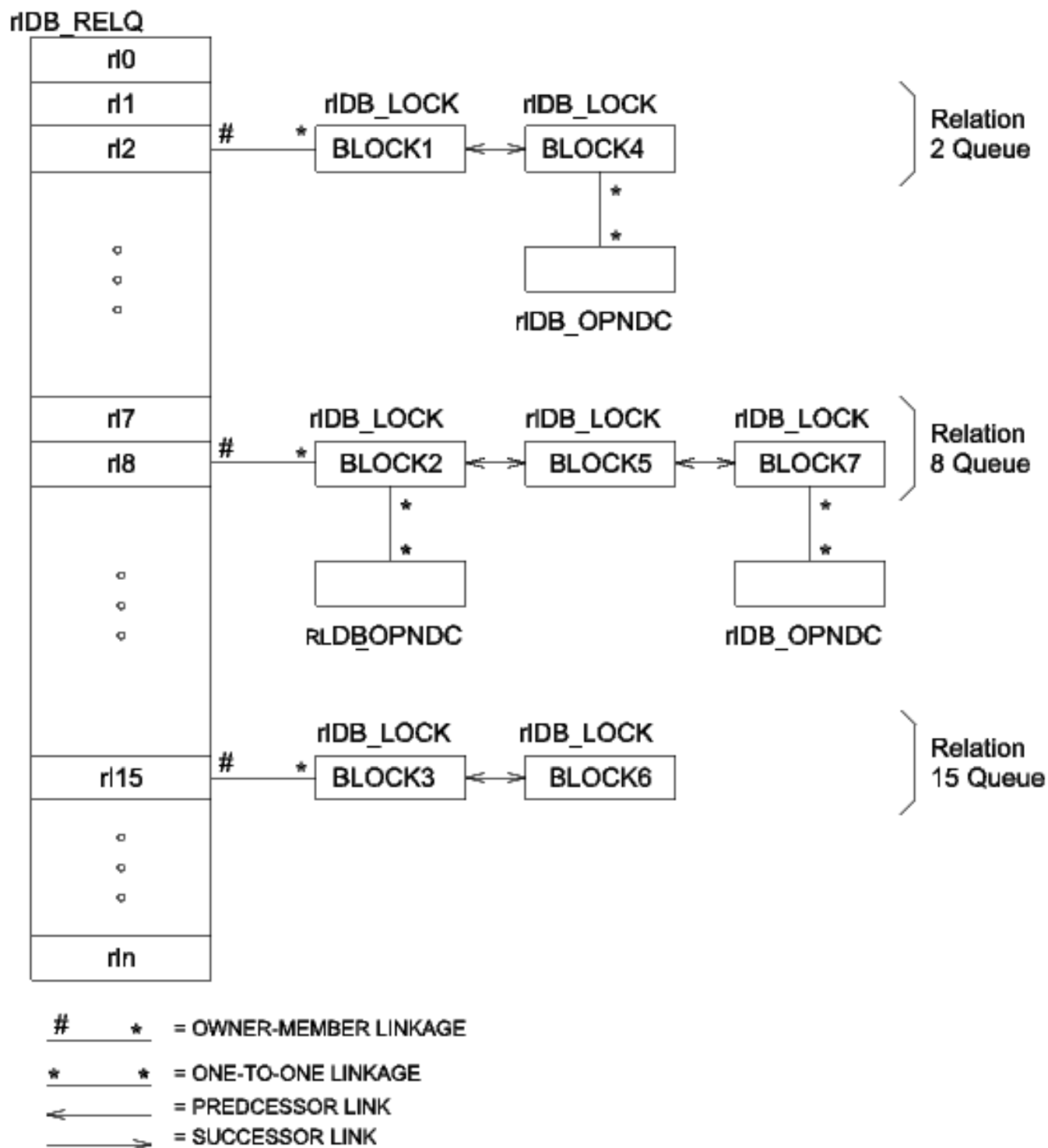


Figure 1 (RELQ-1) Data Structures of the RELQ Audit

3. AUDIT CHECKS

The RELQ audit calls the linkage driver (DRLINK) to check the linkages between the tuples of the relation queue dictionary and the database lock dictionary blocks that are owned by them. The audit performs no semantic checks.

4. ERROR HANDLING

If the audit detects any broken linkages on the active relation queue list, all the transactions are aborted and all the related resources (open dictionary entries, database lock dictionary entries, head tables, and data pages) are reclaimed.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes.

REMACS

Software Release: 5E10 and later

1. REMOTE ACCESS SUBSYSTEM (REMACS) AUDIT

This is a routine audit that is run automatically. However, it can also be run on demand through the maintenance terminal.

The purpose of this audit is to assure consistency between the data in the remote nodes and the data in the administrative module. The data is divided into "windows" (a window is a consecutive area of memory consisting of one or more bytes). REMACS provides an audit of each window. This audit compares the data in the administrative module with the data in the nodes on the basis of byte-to-byte comparison. If the comparison does not match, the data in the nodes is audited a second time. If the second audit fails, the data in the nodes is updated. The erroneous data in the nodes and the correct data in the administrative module are printed on the receive-only printer (ROP). If the data comparison finds no data discrepancies, the audit is then performed on the next consecutive chunk of memory within the window. This check is performed on all auditable windows each time the audit is invoked. All of the windows for all nodes are audited sequentially to complete one audit cycle.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

REPLYBLF

Software Release: 5E10 and later

1. SWITCHING MODULE LINK HEADER BUSY LAMP FIELD (REPLYBLF) AUDIT

The tuples in the RLREPLYBLF relation indicate the DN idle/busy status for a particular hundreds group. This information is stored in the hundreds group routing module (HGRM). Integrated services digital network (ISDN) attendant consoles send messages to the HGRM using a busy lamp field system process (BLFSP) which causes corresponding rIREPLYBLF tuples go from idle to busy. When an rIREPLYBLF tuple becomes busy, the timer field counts down to 0, and at 0, the tuple goes from busy back to idle unless the HGRM receives an update message from the Attendant Console. An HGRM can request the DN busy/idle status of a particular SM, and the rIREPLYBLF counter field counts the number of SMs which have responded to such a request.

The switching module link header busy lamp field audit (REPLYBLF) verifies the RLDBHEADC idle and busy list linkages for the RLREPLYBLF relation. The REPLYBLF audit is part of the attendant features audits subgroup. It runs in the OSDS-M environment in routine and elevated modes.

2. DATA STRUCTURES

The RLREPLYBLF relation uses RLDBHEADC as a headcell for its busy and idle lists. If the RLREPLYBLF tuple is on the RLDBHEADC busy list, and the hundreds group controller is in use, there will be a one-to-one linkage between RLREPLYBLF and RLMODH_CLF.

3. AUDIT CHECKS

The audit calls the indirect access sequential key driver (DRINASEK) to step through the RLREPLYBLF relation and perform linkage, quarantine state, and key checks for each tuple. The audit then checks the rIREPLYBLF tuple's timer field to determine whether or not the tuple should be linked to the RLDBHEADC idle or busy list and whether, based on the status of the hundreds group controller, there should be a one-to-one linkage between this rIREPLYBLF tuple and a rIMODH_BLF tuple.

4. ERROR HANDLING

rIREPLYBLF tuples found to be incorrectly linked to the rIDBHEADC busy or idle lists, or found to be incorrectly linked to rIMODH_BLF tuples, are recovered to the rIDBHEADC idle list to preserve resources.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ALINK

Error Code:	ALINK
Error Description:	The hundred's group controller was using an rIREPLYBLF tuple but this tuple was found not to have a one-to-one linkage with a corresponding rIMODH_BLF tuple, which should be present under these conditions.
Possible Error Effect:	The hundred's group controller will not receive proper update messages from ISDN attendant console equipped RSMs.
Corrective Action Taken:	The audit driver DRINASEK will zero the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIREPLYBLF tuple.
<i>Bad Data =</i>	0 since there is no one-to-one linkage between rIREPLYBLF and MODH_BLF.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rIREPLYBLF tuple.
<i>Dump =</i>	The contents of the rIREPLYBLF tuple.

Table 2 Error Code: BLINK

Error Code:	BLINK
Error Description:	The hundred's group controller was NOT using an rIREPLYBLF tuple but this tuple was found to have a one-to-one linkage with a corresponding rMODH BLF tuple, which should NOT be present under these conditions.
Possible Error Effect:	The hundred's group controller will not receive proper update messages from ISDN attendant console equipped RSMs.
Corrective Action Taken:	The audit driver DRINASEK will zero the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIREPLYBLF tuple.
<i>Bad Data</i> =	The one-to-one linkage that was found between rIREPLYBLF and MODH BLF.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rIREPLYBLF tuple.
<i>Dump</i> =	The contents of the rIREPLYBLF tuple.

Table 3 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	An idle tuple was found linked to the busy list.
Possible Error Effect:	Unless corrected, the idle tuple will forever remain on the busy list resulting in a waste of resources.
Corrective Action Taken:	The audit driver DRINASEK will zero the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIREPLYBLF tuple.
<i>Bad Data</i> =	Owner linkage of the rIREPLYBLF tuple.
<i>Good Data</i> =	Key of the headcell of the busy list.
<i>Logical Index</i> =	Logical key of the rIREPLYBLF tuple.
<i>Dump</i> =	The contents of the rIREPLYBLF tuple.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A busy tuple was found linked to the idle list.
Possible Error Effect:	The busy tuple will be seen as an available resource when, in fact, it is not.
Corrective Action Taken:	The audit driver DRINASEK will zero the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIREPLYBLF tuple.
<i>Bad Data</i> =	Owner linkage of the REPLYBLF tuple.
<i>Good Data</i> =	Key of the headcell of the idle list.
<i>Logical Index</i> =	Logical key of the rIREPLYBLF tuple.
<i>Dump</i> =	The contents of the rIREPLYBLF tuple.

Table 5 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An rIREPLYBLF tuple was found in an idle state but some of its linkages not involved in the idle list mapping are non-zero.
Possible Error Effect:	Resources are linked to an rIREPLYBLF tuple on the idle list, and may not be usable by call processing.
Corrective Action Taken:	The audit driver DRINASEK will zero the tuple along with the linkages involved and places it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIREPLYBLF tuple.
<i>Bad Data</i> =	The linkage that was found.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the rIREPLYBLF tuple.
<i>Dump</i> =	NA.

RFD

Software Release: 5E10 and later

1. RECEIVE FRAME DESCRIPTOR (RFD) AUDIT

Receive frame descriptors (RFD) are used by the direct memory access (DMA) processor when the processor receives frames. It then seizes a RFD from the free list, puts the RFD into the channel queue, and notifies the port processor (PP). The port processor removes the RFD from the queue, copies the RFD information into a frame structure, and passes the structure to layer 2. The port processor then returns the RFD to the RFD free list.

2. DATA STRUCTURES

PSS_RFD.

3. AUDIT CHECKS

The RFD audit performs audits of the RFD free list, per channel RFD lists, collects lost RFDs, and puts them back on to the free list.

4. ERROR HANDLING

The error handling of this audit involves setting the head and tail pointers of free and channel RFD queues; checking the total number of RFDs per channel; handling merged and circular lists; and returning the lost RFDs to the free list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The audit is auditing the RFD channel list. It found the "nrfd" field of the first RFD to be null and the tail offset does not point to this RFD.
Possible Error Effect:	This error will result in loss of frames on that channel as tail offset does not point properly to the tail of the list. This can result in interrupts from operational code.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to the first RFD in the channel list which should be equal to the head offset here.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the channel control block of that channel whose list we are auditing.
<i>Bad Data =</i>	The value of tail offset of the channel list.
<i>Good Data =</i>	The address (offset) of the first RFD in the channel list. Here it will be equal to head offset.
<i>Logical Index =</i>	The number of the channel whose RFD list we are auditing.
<i>Dump =</i>	NA.

Table 2 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The audit is auditing the RFD free list. It found the "nrfd" field of the first RFD to be null and the tail offset does not point to this RFD.
Possible Error Effect:	This error will result in running out of RFDs as the RFDs will be returned at the tail of the free list and the tail offset of the free list does not point to the end of the free list.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to the first RFD in

	the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the last RFD in the free list.
<i>Bad Data =</i>	The value of tail offset of the free list.
<i>Good Data =</i>	The address (offset) of the first RFD in the free list. Here it will be equal to head offset.
<i>Logical Index =</i>	This value is "PSMAXCHN + 2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	NA.

Table 3 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	While going through the RFD list for the channel, audit found the owner field of a RFD other than the first RFD, not equal to the channel number that is, bad value of owner field. This means either some other channel RFD list has merged with this list or the free list has merged with this list or the operational code did not update the owner when it put this RFD on the channel list or there is wild write.
Possible Error Effect:	This error may result in interrupts from operational code.
Corrective Action Taken:	The audit will report the error and will set the tail offset to previous RFD, will update the "rfdcnt" field and set the "nrfd" field of the previous RFD to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the current RFD where audit found bad owner value.
<i>Bad Data =</i>	The value of "owner" field of the current RFD.
<i>Good Data =</i>	The channel number of the RFD list being audited as that should be the value of the owner field.
<i>Logical Index =</i>	The channel number whose RFD channel list is being audited.
<i>Dump =</i>	NA.

Table 4 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The owner field of the first RFD in the RFD free list has bad value of owner (other than free list owner value). This is possible if any of the channel RFD queue has merged with free list or the value of owner field was not updated correctly when put back to free list or the value of owner field had wild write.
Possible Error Effect:	This error may result in a resource problem of RFD and that will be reported by operational code (DMAP). This can also result in loss of frames on a channel if channel queue has merged with the free list because the RFD will be taken off the channel list rather than the free list. Operational code may take interrupts for more than one reason.
Corrective Action Taken:	The audit will report the error and will set the head offset to point to tail offset or a valid RFD if tail offset is also bad. All the RFDs that the head was pointing to, would be put back to free list reporting them as "LOST" RFDs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the first RFD in the free list.
<i>Bad Data =</i>	The value of owner field of the first RFD in the free list.
<i>Good Data =</i>	PSMAXCHN, the owner value when RFD is on the free list.
<i>Logical Index =</i>	This value is "PSMAXCHN + 2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	NA.

Table 5 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	While going through the RFD list for the channel, audit reached the snap shot value of the tail offset and compared the snap shot value of the RFD count with the running count kept by the audit and they do not match.
Possible Error Effect:	This error may result in interrupts from operational code. We may have either

	more or less number of RFDs on the channel queue than the count shows. Some of the frames will be lost which will be resent.
Corrective Action Taken:	The audit will report the error and will set the RFD count value equal to the running count value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	This field is used here to dump the current value of the RFD count which can be different from the snap shot value or the running count value.
<i>Bad Data =</i>	The snap shot value of the RFD count.
<i>Good Data =</i>	The RFD running count value kept by audit.
<i>Logical Index =</i>	The channel number whose RFD channel list is being audited.
<i>Dump =</i>	NA.

Table 6 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The tail offset of the RFD free list does not point to the RFD whose "nrfd" field has null value.
Possible Error Effect:	We may have a loss of resource problem as the RFDs are returned at the tail offset and tail offset does not point to the actual tail of the RFD free list.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to this RFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the variable which holds the value of the tail offset of RFD free list.
<i>Bad Data =</i>	The value of the tail offset of RFD free list.
<i>Good Data =</i>	The offset value of the address of the current RFD.
<i>Logical Index =</i>	This value is "PSMAXCHN + 2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	NA.

Table 7 Error Code: BLINK

Error Code:	BLINK
Error Description:	The head offset of this channel's RFD queue has null value and audit found that tail offset does not have null value which it should have.
Possible Error Effect:	This may result in no frames being transferred on that channel as DMAP is putting RFDs at the end and PP thinks there are no RFDs to transfer on this channel. The operational code may take interrupts.
Corrective Action Taken:	The audit will report the error and will set the value of the tail offset to null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the channel control block for this channel.
<i>Bad Data =</i>	The value of "rqli", tail offset, of this channel.
<i>Good Data =</i>	PSNULLOFF, the null value of the tail offset.
<i>Logical Index =</i>	The number of the channel whose tail offset has bad value.
<i>Dump =</i>	NA.

Table 8 Error Code: BUSYCHAN

Error Code:	BUSYCHAN
Error Description:	While auditing the RFD free list, audit found the value of head and tail offset to be out of range and it did not find a single RFD whose owner field has a value of free list owner that is, it is attached to the free list. This condition is very rare and we should never see this error because finding this error means all the RFDs are currently in use and the value of head and tail offsets of the RFD free list have bad value.
Possible Error Effect:	This error represents an emergency situation as there are no more RFDs left on the RFD free list. The operational code will report various error reports and interrupts including resource error.
Corrective Action Taken:	The audit will report the error and will take the very first RFD no matter where it is by going through index and set the head and tail of the RFD free list to point to this RFD.

Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. NA.
<i>Bad Data =</i>	0. NA.
<i>Good Data =</i>	PSMAXCHN. The value of the RFD free list owner. This value is put here to say that no RFD of this owner was found which should not have happened.
<i>Logical Index =</i>	PSMAXCHN +2. Representing that RFD free list is being audited at that point.
<i>Dump =</i>	NA.

Table 9 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The head offset of this channel's RFD queue has null value and audit found a non-zero value to the "rfdcnt" field which represents total number of RFDs currently on the RFD channel queue.
Possible Error Effect:	No significant effect other than if the value is not zero, it might get updated to a wrong value when RFDs are attached to that list which in turn may lead to false value of RFDs being allowed on that list.
Corrective Action Taken:	The audit will report the error and will set the value of the "rfdcnt" to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the channel control block for this channel.
<i>Bad Data =</i>	The value of "rfdcnt", RFD count, of this channel.
<i>Good Data =</i>	0. No RFDs on this channel.
<i>Logical Index =</i>	The number of the channel whose "rfdcnt" had bad value.
<i>Dump =</i>	NA.

Table 10 Error Code: HEAD

Error Code:	HEAD
Error Description:	The audit started auditing the RFD free list and the value of head offset (the pointer to the first RFD of the RFD free list) failed the range and boundary check. This means it is not pointing to an address which is a valid starting address of any of the RFDs.
Possible Error Effect:	This error will immediately result in a resource problem of RFDs and will be reported by operational code (DMAP). If not corrected, this audit will be called in direct mode to correct the problem.
Corrective Action Taken:	The audit will report the error and will set the head offset to point to tail offset or a valid RFD if tail offset is also bad. All the RFDs that the head was pointing to, would be put back to free list reporting them as "LOST" RFDs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. The head offset which is an address in itself is bad and is dumped in Bad data field.
<i>Bad Data =</i>	The value of head offset.
<i>Good Data =</i>	0. The value chosen here is zero but any value which is a valid beginning address of the RFD is a good data.
<i>Logical Index =</i>	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	NA.

Table 11 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	The head offset of this channel's RFD queue failed the range and boundary check. That is it is not pointing to an address which is the starting address of any of the RFDs.
Possible Error Effect:	This may result in no frames being processed at this channel as PP processor will always think that there is no RFD in the queue.
Corrective Action Taken:	The audit will report the error, will set the value of the head and tail offset to null value.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The address of the channel control block for this channel.
<i>Bad Data</i> =	The value of "rqhd", head offset, of this channel.
<i>Good Data</i> =	PSNULLOFF, the null value of the head offset. Also any valid RFD offset can be a good data.
<i>Logical Index</i> =	The number of the channel whose head offset has bad value.
<i>Dump</i> =	NA.

Table 12 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	While going through the RFD list for the channel, audit found the "nrfd" field of the current RFD it is auditing to be null and the tail offset of this RFD channel queue not to be pointing to this RFD, which it should be.
Possible Error Effect:	This error may result in interrupts from operational code. We may lose some frames as the tail offset has bad value and any RFDs attached to this frame will be lost.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to this RFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of this channel's RFD queue tail offset.
<i>Bad Data</i> =	The value of the tail offset.
<i>Good Data</i> =	The offset value of the current RFD audit is auditing.
<i>Logical Index</i> =	The channel number whose RFD channel list is being audited.
<i>Dump</i> =	NA.

Table 13 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	While going through the RFD free list, the value of head pointer or the value of owner field of the first RFD in the free list has bad value. So when audit tried to check the tail offset value of the RFD free list, it was also found to have the bad value that is, its value did not pass the range and boundary checks.
Possible Error Effect:	This error may result in interrupts from operational code. It can also cause the resource error by the operational code.
Corrective Action Taken:	The audit will report the error and will set the head and tail offset to a valid RFD whose owner field says it is on the free list while going through RFD list by index.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. As we have found out that the value of tail offset is out of range anyway.
<i>Bad Data</i> =	The value of the tail offset.
<i>Good Data</i> =	0. The value is chosen zero but the valid value should be any starting address value of the RFD in the form of offset.
<i>Logical Index</i> =	PSMAXCHN+2. Representing that the RFD free list is being audited.
<i>Dump</i> =	NA.

Table 14 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit has found a RFD whose "touch" field is not set that is not equal to 1 when it was going through all the RFDs by index again at the end of the audit. This means that neither audit nor the operational code came across this RFD during the current run of this audit and therefore it is "LOST". One of the possible causes of "LOST" RFD could be that it is linked to the list and the "nrfd" field of the previous RFDs was bad. So the audit updated the list making that or the previous RFD in the list and now all the RFDs following that RFD are "LOST". This can happen in free list or channel list.
Possible Error Effect:	This error has no immediate effect on the system but if it is not detected and if RFD is not put back to proper place (free list) it may result in resource problem of RFDs.
Corrective Action Taken:	The audit will report the error and will put this "LOST" RFD back to free list.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the "LOST" RFD.
<i>Bad Data</i> =	The value of "owner" field of this RFD.
<i>Good Data</i> =	PSMAXCHN. The value of the owner field when on free list. It is assumed here that this RFD, should at least be on the free list.
<i>Logical Index</i> =	The RFD number which is found "LOST".
<i>Dump</i> =	The RFD which found to be lost.

Table 15 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The owner field of the first RFD in the RFD channel list has bad value of owner (other than channel number).
Possible Error Effect:	This can result in loss of frames on the channel or channels if the channel list has merged with this channel's list. It can also cause resource problem if RFD free list has merged with this channel's list as we may run out of RFDs or operational code may complain about those RFDs as certain fields would not have been set.
Corrective Action Taken:	The audit will report the error and will set the head and tail offset to null value and the "rfdcnt" to zero value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the first RFD in the channel list.
<i>Bad Data</i> =	The value of owner field of the first RFD in this channel's RFD list.
<i>Good Data</i> =	The channel number whose list is being audited as that is the owner value for all RFDs on that list.
<i>Logical Index</i> =	The channel number whose list is being audited.
<i>Dump</i> =	NA.

Table 16 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The audit found the "nrfd" field of the very first RFD in the RFD free list pointing to the head offset. Thus the RFD free list has become a circular list of one member.
Possible Error Effect:	This error may result in a resource problem of RFD if operational code cannot find more RFDs on the free list.
Corrective Action Taken:	The audit will report the error, will set the "nrfd" field to null value and set the tail offset to point to this RFD which will be equal to the head offset here.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the first RFD in the free list.
<i>Bad Data</i> =	The value of "nrfd" field of the first RFD on the free list.
<i>Good Data</i> =	PSNULLOFF. The value chosen here is null but any value which is a valid beginning address of the RFD is a good data.
<i>Logical Index</i> =	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump</i> =	NA.

Table 17 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The audit found the "nrfd" field of the very first RFD in the RFD channel list pointing to the head offset. Thus the RFD channel list has become a circular list of one member.
Possible Error Effect:	This error may result in losing some of the frames on that channel and the operational code may take an interrupt.
Corrective Action Taken:	The audit will report the error, will set the head and tail offset to null value and the "rfdcnt" to zero value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the first RFD in the channel list.
<i>Bad Data</i> =	The value of "nrfd" field of the first RFD on the channel list.

<i>Good Data</i> =	PSNULLOFF. The value chosen here is null but any value which is a valid beginning address of the RFD is a good data.
<i>Logical Index</i> =	The value of the channel whose RFD list is being audited.
<i>Dump</i> =	NA.

Table 18 Error Code: NULL5LINK

Error Code:	NULL5LINK
Error Description:	While going through the RFD list for the channel, audit found the running count of the total number of RFDs on this channel exceeded by the maximum value allowed. This means that either there are more number of RFDs on this channel than allowed or the queue has become a circular queue.
Possible Error Effect:	This error may result in interrupts from operational code.
Corrective Action Taken:	The audit will report the error and will set both the tail and head offset to null and RFD count field to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. As we do not know where and which error occurred.
<i>Bad Data</i> =	0. NA.
<i>Good Data</i> =	The null value of "nrfd" field as we should have encountered that condition in any of the RFD before we reached at this point.
<i>Logical Index</i> =	The channel number whose RFD channel list is being audited.
<i>Dump</i> =	NA.

Table 19 Error Code: ORCOUNTS

Error Code:	ORCOUNTS
Error Description:	Audit first calculated the number of free RFDs by subtracting the existing value of number of RFDs seized by IOP/DMAPI processor from the RFD free list from total number of RFDs. Then it calculated this count again while auditing the RFD free list. If these two counts do not match with a tolerance value of AURFDTOLERANCE, it results in this error.
Possible Error Effect:	This error, if left undetected, may potentially result in a PH being in overload condition forever.
Corrective Action Taken:	The audit will report the error and will correct the subgate RFD returned count.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The tolerance value for these counts, AURFDTOLERANCE.
<i>Bad Data</i> =	The initial count of the free RFDs.
<i>Good Data</i> =	The count of free RFDs calculated by audit.
<i>Logical Index</i> =	The value of PSMAXCHAN. It has no real significance for this error.
<i>Dump</i> =	NA.

Table 20 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit has found a RFD whose "nrfd" field neither passed the range and boundary checks nor it is null when audit was traversing this RFD by index at the beginning of the audit.
Possible Error Effect:	This error could also have been detected by operational code resulting in an assert.
Corrective Action Taken:	The audit will report the error and will set the "nrfd" field to a null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the RFD whose "nrfd" field was not good.
<i>Bad Data</i> =	The value of "nrfd" field of this RFD.
<i>Good Data</i> =	PSNULLOFF. The value chosen to dump here is null value but along with this null value, any value which satisfies the range and boundary check is a good value. By satisfying range and boundary checks means this RFD is pointing to a RFD address which is the starting address of any one of the total RFDs.
<i>Logical Index</i> =	The RFD number whose "nrfd" field is bad.
<i>Dump</i> =	NA.

Table 21 Error Code: OWNER

Error Code:	OWNER
Error Description:	The owner field of the current RFD audit is visiting in the RFD free list has bad value of owner (other than free list owner value). This is possible if any of the channel RFD queue has merged with free list or the value of owner field was not updated correctly when put back to free list or the value of owner field had wild write.
Possible Error Effect:	This can result in loss of frames on a channel if channel queue has merged with the free list. Because the RFD will be taken off the channel list rather than the free list. The operational code may take interrupts.
Corrective Action Taken:	The audit will report the error and will set the "nrfd" field of the RFD to null and set the tail offset to point to this RFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the current RFD audit is visiting in the free list.
<i>Bad Data =</i>	The value of owner field of the RFD audit is visiting in the free list.
<i>Good Data =</i>	PSMAXCHN, the owner value when RFD is on the free list.
<i>Logical Index =</i>	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	NA.

Table 22 Error Code: PQCNT

Error Code:	PQCNT
Error Description:	When audit found that the "nrfd" field of the first RFD of the RFD channel list has null value, the value of running count and the "rfdcnt" did not match, that is, the value of "rfdcnt" was not one.
Possible Error Effect:	This error may result in an interrupt by operational code.
Corrective Action Taken:	The audit will report the error and will set the value of the "rfdcnt" equal to one.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the channel control block of the channel whose RFD channel list is being audited.
<i>Bad Data =</i>	The value of "rfdcnt" field of the channel control block for this channel.
<i>Good Data =</i>	rng_cnt, running count value which should be 1.
<i>Logical Index =</i>	The channel number whose RFD channel list is being audited.
<i>Dump =</i>	NA.

Table 23 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The running count of the RFDs for the free list exceeded the total number of the RFDs per channel. Hence the RFD free list has become a circular list.
Possible Error Effect:	This can result in loss of resource of RFDs since all the RFDs are returned at the tail and the tail is not pointing where it is supposed to point (at the end).
Corrective Action Taken:	The audit will report the error, will set the "nrfd" field of the current RFD to null and set the tail offset to point to this RFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. We do not have any idea of where in the free list we found this to be the circular so we cannot dump the address of this RFD.
<i>Bad Data =</i>	0. We do not know which RFD's "nrfd" field is bad.
<i>Good Data =</i>	PSNULLOFF, null value of the "nrfd" field. This could have prevented the free list being a circular list.
<i>Logical Index =</i>	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	NA.

Table 24 Error Code: RQCNT

Error Code:	RQCNT
Error Description:	While going through the RFD list for the channel, audit found the "nrfd" field of

	the current RFD it is auditing to be null and the value of RFD count does not match the value of the running count kept by the audit.
Possible Error Effect:	This error may result in interrupts from operational code. We may have more or less than allowable number of RFDs on the queue or the rfdcnt variable has a wild write.
Corrective Action Taken:	The audit will report the error and will set the RFD count equal to the running count kept by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of this channel's RFD queue's RFD count.
<i>Bad Data</i> =	The value of the RFD count.
<i>Good Data</i> =	The running count value kept by audit.
<i>Logical Index</i> =	The channel number whose RFD channel list is being audited.
<i>Dump</i> =	NA.

Table 25 Error Code: ZAP

Error Code:	ZAP
Error Description:	The RFD audit entered in the ZAP mode. This means that when the processor (PP) took a non-returning error, Audit Control decided that RFD audit was running at the time.
Possible Error Effect:	The error is printed to notify that RFD audit was entered in the ZAP mode. It causes resets and asserts.
Corrective Action Taken:	The audit will decide from the value of AUcount, semi-global variable in au/AUups module where audit was working when it got zapped and accordingly it will take appropriate action.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA. Audit does not know the address where the processor took error.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcount. The value of AUcount variable shows where audit was when it got zapped. The following is the list of possible values of AUcount and what is the meaning of it: 0-127 = The audit was auditing the RFD channel queue for channel equal to AUcount. Make sure that the write protection of this channel is turned off. 128 = The audit was in the middle of going from one task to other. Do nothing. 129 = The audit was zapped while going through the RFDs by its index value. Nothing to recover. 130 = The audit was zapped while it was auditing the RFD free list. Nothing to recover. 131 = The audit was zapped while returning "lost" RFDs. Return them here. 132 and higher = This is an invalid value. Turn off write protection of all the RFD channel queues for precaution.
<i>Dump</i> =	NA.

RGBUS

Software Release: 5E10 and later

1. RINGING GENERATOR BUS (RGBUS) AUDIT

The integrated service line unit (ISLU) Z-interface (ISLU-Z) converts digital signals to analog. The ISLU-Z is used by ringing generators (RGs) and other metallic access network (MAN) circuits. Several audits check the data structures related to the ISLU-Z circuits: the ringing generator bus (RGBUS) audit, the ringing generator occupancy table (RGOCUP) audit, the ringing generator task data block (RGTDDB) audit, the channel control block user (CCBUSR) audit, the ISLU-Z access network buses (ISANBUS) audit, and the spare line care (SPRCRD) audit. The RGBUS audit verifies data reflecting the connections of the ringing generator, ring bus, and line cards.

The RGBUS audit is part of the peripheral control audits subgroup. It is part of the ISLU application loadable package (ISLUAPP). It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The RGBUS audit uses these data structures: the peripheral control ringing bus relation (RLRGBUS), the channel control block user relation (RLCCBUSR), the switch maintenance equipment status relation (RLSMEST), the circuit data relation (RLCKTDATA), and the channel data block common data relation (RLCDBCOM).

3. AUDIT CHECKS

The RGBUS audit calls the linkage driver (DRLINK) to perform point-to/point-back checks between rIRGBUS and rICCBUSR. When the linkage checks pass, the semantic check function makes the following checks of tuples in the RLRGBUS relation:

TYPE	= Verify rIRGBUS Line Unit type.
UNEQUIP	= If associated circuit is unequipped, rIRGBUS should be empty.
CKTNAME	= Verify existence of circuit from rIRGBUS.
MC_NOMTCH	= Bus circuit does not exist in RLCKTDATA.
READSMEST	= Unable to read RLSMEST for circuit.
OOSQUAL	= Verify rIRGBUS to rISMEST maintenance state for BUS.
OOS_BLINK	= Verify rIRGBUS to rISMEST maintenance state for MAN.
BAS_STAT	= Verify MAN/BUS/RG maintenance states with rIRGBUS.
BAS_EQPG	= MAN and BUS are in-service, RGs are out-of-service, but rIRGBUS does not reflect this.
QUAL_2	= MAN is in-service/degraded, but rIRGBUS does not reflect this.
NOTBUSY	= The maintenance and queue bits are not set, but the associated rICDBCOM is not idle.
LOSTCDB	= Can not access associated rICDBCOM.
LPSTATE	= The maintenance bit is set, but the associated rICBCOM's lpstate is not maintenance.
CQBCDB	= The queue bit is set but no rICQB linkage exists.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BAS_EQPG

Error Code:	BAS_EQPG
Error Description:	The audit has found the rdbus being checked to be active, the associated manboard active and both ringing generators not active, but the rg field of the rdbus being checked is set to something other than "no rg."
Possible Error Effect:	A rdbus may be used for ringing when it should not be.
Corrective Action Taken:	Set the rg field to "no rg," and send a request to remove/restore the associated manboard to get the correct ringing generator connected to the bus being checked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rg field of rdbus being checked.
<i>Good Data =</i>	smest circuit name for the associated manboard.
<i>Logical Index =</i>	smest bas_stat for the associated manboard.
<i>Dump 1 =</i>	smest tuple for the rdbus tuple being checked.
<i>Dump 2 =</i>	smest tuple for ringing generator 0.
<i>Dump 3 =</i>	smest tuple for ringing generator 1.

Table 2 Error Code: BAS_STAT

Error Code:	BAS_STAT
Error Description:	The audit has found the rdbus being checked to be active, the associated manboard active and at least one active ringing generator, but the rg field of the rdbus being checked is set to "no rg."
Possible Error Effect:	A rdbus may not be able to be used for ringing.
Corrective Action Taken:	Zero out the rdbus data, restore the key and rdbus fields, set the rg field to "no rg," and send a request to remove/restore the associated manboard to get the correct ringing generator connected to the bus being checked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rg field of rdbus being checked.
<i>Good Data =</i>	smest circuit name for the associated manboard.
<i>Logical Index =</i>	smest bas_stat for the associated manboard.
<i>Dump 1 =</i>	smest tuple for the rdbus tuple being checked.
<i>Dump 2 =</i>	smest tuple for ringing generator 0.
<i>Dump 3 =</i>	smest tuple for ringing generator 1.

Table 3 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	The audit has detected an inconsistency between the circuit name in the rdbus tuple and the circuit name in rCKTDATA.
Possible Error Effect:	An invalid circuit may be selected to do ringing.
Corrective Action Taken:	The rdbus circuit name will be updated to agree with that in cktdata.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rdbus data being checked.
<i>Bad Data =</i>	Circuit name found in rdbus field of rdbus tuple being checked.
<i>Good Data =</i>	Circuit name from cktdata.
<i>Logical Index =</i>	Key of the rdbus tuple being checked.
<i>Dump 1 =</i>	rdbus tuple being checked.
<i>Dump 2 =</i>	cktdata tuple for the rdbus being checked.

Table 4 Error Code: CQBCDB

Error Code:	CQBCDB
Error Description:	The audit has found that the rbus queue field indicates that queuing action for this rbus is pending yet the cdbcom owner is set to null.
Possible Error Effect:	An rbus may not be able to be used for ringing.
Corrective Action Taken:	Set the rbus queue field to indicate no queuing action.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of the rbus queue field. (Not an address.)
<i>Bad Data =</i>	Value of associated cdbcom owner field. (Not bad data per se.)
<i>Good Data =</i>	DBNULL.
<i>Logical Index =</i>	Value of rbus maintenance field.
<i>Dump 1 =</i>	Tuple for the rbus tuple being checked.
<i>Dump 2 =</i>	Tuple for the associated cdbcom.

Table 5 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	The audit cannot access the data for the rCCBCOM associated with the rbus being checked.
Possible Error Effect:	A rbus may not be able to be used for ringing.
Corrective Action Taken:	Quarantine the cdbcom tuple and schedule the CKTDATA audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA
<i>Bad Data =</i>	Circuit name of rbus being checked. (Not bad data per se.)
<i>Good Data =</i>	Return code from trying to access the ccbcom data.
<i>Logical Index =</i>	Name field of the associated cdbcom tuple.
<i>Dump 1 =</i>	Tuple for the rbus tuple being checked.
<i>Dump 2 =</i>	Tuple for the associated cdbcom.
<i>Dump 3 =</i>	Tuple for the associated ccbcom.

Table 6 Error Code: LPSTATE

Error Code:	LPSTATE
Error Description:	The audit has found that the associated ccbcom lpstate field is not set to indicate maintenance action when it should be.
Possible Error Effect:	An rbus may not be able to be used for ringing.
Corrective Action Taken:	Quarantine the ccbcom tuple and schedule the CKTDATA audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	smest bas_stat for the rbus being checked. (Not an address.)
<i>Bad Data =</i>	Value of associated ccbcom lpstate field.
<i>Good Data =</i>	Value the lpstate should be.
<i>Logical Index =</i>	Owner field of the associated cdbcom tuple.
<i>Dump =</i>	Tuple for the rbus tuple being checked.

Table 7 Error Code: MC_NOMTCH

Error Code:	MC_NOMTCH
Error Description:	The audit has detected a non-zeroed rbus tuple for a rbus not found in cktdata.
Possible Error Effect:	An invalid rbus may be selected to do ringing.
Corrective Action Taken:	The rbus circuit name will be zeroed and the key field restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rbus data being checked.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Circuit name for the rbus being checked.
<i>Logical Index =</i>	Key of the rbus tuple being checked.
<i>Dump =</i>	rbus tuple being checked.

Table 8 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The audit has found the rCDBCOM associated with the rbus being checked

	not linked to the idle list when it should be.
Possible Error Effect:	An rdbus may not be able to be used for ringing.
Corrective Action Taken:	Quarantine the rCDBCOM tuple and schedule the cktdata audit to relink to correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of the rdbus maintenance bit. (Not an address.)
<i>Bad Data =</i>	Owner of the cdbcom associated with the rdbus being checked.
<i>Good Data =</i>	Idle list the cdbcom should be linked to.
<i>Logical Index =</i>	Value of the rdbus queue bit.
<i>Dump 1 =</i>	Tuple for the rdbus tuple being checked.
<i>Dump 2 =</i>	Tuple for the associated cdbcom.

Table 9 Error Code: OOS_BLINK

Error Code:	OOS_BLINK
Error Description:	The audit has found the associated manboard to be out of service in rSMEST while the rg field of the rdbus being checked is not set to "no rg".
Possible Error Effect:	A rdbus may be used for ringing when it should not be.
Corrective Action Taken:	Set the rg field to "no rg".
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; the rdbus field of the rdbus tuple being checked.
<i>Bad Data =</i>	rg field of rdbus being checked.
<i>Good Data =</i>	rSMEST bas_stat of the associated manboard.
<i>Logical Index =</i>	rSMEST unit_id of the associated manboard.
<i>Dump 1 =</i>	smest tuple for the associated manboard.
<i>Dump 2 =</i>	rdbus tuple being checked.
<i>Dump 3 =</i>	smest tuple for the rdbus tuple being checked.

Table 10 Error Code: OOSQUAL

Error Code:	OOSQUAL
Error Description:	The audit has found the rdbus being checked to be out of service in smest and the rdbus rg field not set to "no rg" while the maintenance bit for this rdbus is not set.
Possible Error Effect:	An invalid rdbus may be selected to do ringing.
Corrective Action Taken:	Zero out the rdbus data, restore the key and rdbus fields, set the rg field to "no rg", set the maintenance bit, and send a request to remove/restore the associated manboard to get the correct ringing generator connected to the bus being checked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rdbus tuple being checked.
<i>Bad Data =</i>	Ring generator this rdbus is supposedly connected to.
<i>Good Data =</i>	smest bas_stat for the rdbus being checked.
<i>Logical Index =</i>	Key of the rdbus tuple being checked.
<i>Dump 1 =</i>	rdbus tuple being checked.
<i>Dump 2 =</i>	smest tuple for rdbus being checked.

Table 11 Error Code: QUAL_2

Error Code:	QUAL_2
Error Description:	The audit has found the associated manboard to be active degraded in rSMEST while the rg field of the rdbus being checked is not set to "no rg".
Possible Error Effect:	A rdbus may be used for ringing when it should not be.
Corrective Action Taken:	Zero out the rdbus data, restore the key and rdbus fields, set the rg field to "no rg" and send a request to remove/restore the associated manboard to get the correct ringing generator connected to the bus being checked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; unit_id of rSMEST tuple for the associated manboard.
<i>Bad Data =</i>	rg field of rdbus being checked.
<i>Good Data =</i>	rSMEST qual_2 of the associated manboard.
<i>Logical Index =</i>	rdbus field of the rdbus tuple being checked.

<i>Dump 1 =</i>	smest tuple for the rgbus tuple being checked.
<i>Dump 2 =</i>	The rgbus tuple being checked.

Table 12 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The audit cannot access rSMEST data for the rgbus being checked.
Possible Error Effect:	An invalid rgbus may be selected to do ringing.
Corrective Action Taken:	The rSMEST and circuit data audits will be scheduled to determine if an entry should exist in rSMEST for this rgbus
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Return code from a read of rSMEST. (Not bad data.)
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Value of variable used in switch statement.
<i>Dump =</i>	None.

Table 13 Error Code: TYPE

Error Code:	TYPE
Error Description:	The line unit associated with rIRGBUS is of a type such that the rIRGBUS data should be zeroed. The audit has detected that this data is not zeroed.
Possible Error Effect:	An invalid rIRGBUS may be selected to do ringing.
Corrective Action Taken:	The rIRGBUS data will be zeroed and the key field restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIRGBUS data being checked.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Line unit type.
<i>Logical Index =</i>	Key of the rIRGBUS tuple being checked.
<i>Dump =</i>	RGBUS tuple being checked.

Table 14 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The audit has detected non-zero rIRGBUS data for a rIRGBUS that is unequipped in rCKTDATA.
Possible Error Effect:	An invalid rIRGBUS may be selected to do ringing.
Corrective Action Taken:	The rIRGBUS data will be zeroed and the key field restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIRGBUS data being checked.
<i>Bad Data =</i>	Equipage status for the rIRGBUS from rCKTDATA. (This is not bad data per se, but rather informational type data.)
<i>Good Data =</i>	Circuit name of the rIRGBUS being checked.
<i>Logical Index =</i>	rIRGBUS circuit name of the tuple being checked.
<i>Dump 1 =</i>	rIRGBUS tuple being checked.
<i>Dump 2 =</i>	rCKTDATA tuple for the rIRGBUS being checked.

RGOCUP

Software Release: 5E10 and later

1. RINGING OCCUPANCY TABLE (RGOCUP) AUDIT

The integrated services line unit (ISLU) Z analog interface (ISLU-Z) provides analog call processing on the ISLU, including plain old telephone service (POTS) calls, multiparty calls, and coin phone operations on the ISLU. Several audits check the data structures used by the ringing generators (RGs), metallic access network (MAN), and sparing of faulty line cards. The audits include the peripheral control (PC) RGOCUP, the PC ringing task data block (RGTDDB) audit, the ringing generator bus (RGTDB) audit, the channel control block user (CCBUSR) audit, the ISLU-Z access network bus (ISANBUS) audit, the spare line card (SPRCRD) audit, and the ISLU transaction (ISLUTRAN) audit. The RGOCUP audit checks the peripheral control ringing occupancy table (PCrgocup) to ensure that the subscribers are receiving single and multiple burst rings properly during each time slot.

The RGOCUP audit is part of the Peripheral Control audits subgroup. It is part of the ISLU application loadable package (ISLUAPP). It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The RGOCUP audit uses the peripheral control (PC) interface ringing occupancy table (PCrgocup) and the PC ringing task data block array (PCrgtdb). The PCrgtdb array contains a ring type bit (tasktype) that indicates single or multiple task ringing. The occupancy table contains fields for the number of task data blocks sent to the ISLU for ringing for the PCrgocup time slots and number of ringers during each time slot (rgringers). PCrgocup is the owner of PCrgtdb. Each PCrgocup tuple may own up to four PCrgtdb tuples. These data structures are shown in Figure 1.

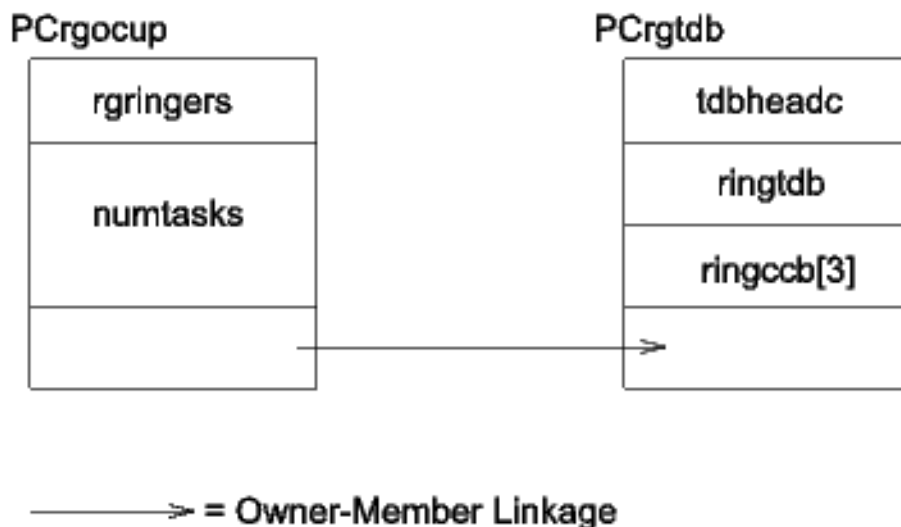


Figure 1 (RGOCUP-1) Data Structures of the RGOCUP Audit

3. AUDIT CHECKS

The audit invokes the linkage check function of the linkage driver (DRLINK) to perform point-to/point-back linkage checks. After the linkage is checked, the audit makes these semantic checks. The audit traverses the task data block linkages in PCrgtdb and counts the number of PCrgtdb tuples linked to this PCrgocup tuple (time slot). This number is the number of tasks sent to the ISLU for ringing. The audit compares it with the numtasks field in PCrgocup. The numbers should be equal. It also checks to see if the total number of tasks is greater than the maximum allowed.

COUNTS = Difference detected between numtasks and PCrgocup.

INV_STATE = Total tasks exceed maximum.

ZAP = The audit was entered after taking a reset.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of tasks (# of pcRGTDDB tuples) send down to the ISLU for this timeslot (pcRGOCUP tuple) is not = numtasks.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	The numtasks field has been corrected to agree with the number of pcRGTDDB tuples linked to this timeslot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the pcRGOCUP tuple.
<i>Bad Data</i> =	Number of tasks.
<i>Good Data</i> =	Total of pcRGTDDB tuples counted.
<i>Logical Index</i> =	Logical key of tuple which is same to physical.
<i>Dump</i> =	The pcRGOCUP tuple.

Table 2 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The number of tasks (# of pcRGTDDB tuples) send down to the ISLU for this timeslot (pcRGOCUP tuple) exceeds the maximum.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	Schedule the RGTDB audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the pcRGOCUP tuple.
<i>Bad Data</i> =	Number of tasks.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple which is same to physical.
<i>Dump</i> =	The pcRGOCUP tuple.

Table 3 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The PCrgocup[][] .rgringers attribute is out of range. The number of "ringers" (phones or other devices which ring) for a line must not exceed ISMAXRGS. The rgringers field was found corrupted to a value above this maximum.
Possible Error Effect:	Call processing will determine that the maximum number of "ringers" has been exceeded, and never send a ringing job to the customers phone(s).
Corrective Action Taken:	The rgringers field is initialized to 0, which is a known initialization state which call processing may again use and process.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the PCrgocup[][] array entry.
<i>Bad Data</i> =	The rgringers value found.
<i>Good Data</i> =	0. The value rgringers was initialized to.
<i>Logical Index</i> =	The address of the PCrgocup[][] array entry.
<i>Dump</i> =	The contents of the PCrgocup[][] array entry.

Table 4 Error Code: ZAP

Error Code:	ZAP
Error Description:	The RGOCUP has been entered in ZAP mode. The audit will be terminated and will start from the beginning the next time it is scheduled to run.
Possible Error Effect:	The RGOCUP audit is terminated before it completes all its checks.
Corrective Action Taken:	The audit will start from its beginning the next time it is scheduled to run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	0.
<i>Dump =</i>	NA.

RGTDDB

Software Release: 5E10 and later

1. RINGING TASK DATA BLOCK (RGTDDB) AUDIT

The integrated services line unit (ISLU) Z analog interface (ISLU-Z) provides analog call processing on the ISLU, including plain old telephone service (POTS) calls, multiparty calls, and coin phone operations on the ISLU. Several audits check the data structures used by the ringing generators (RGs), metallic access network (MAN), and sparing of faulty line cards. The audits include the peripheral control (PC) ringing occupancy table (RGOCUP) audit, the PC RGTDDB, the ringing generator bus (RGBUS) audit, the channel control block user (CCBUSR) audit, the ISLU-Z access network bus (ISANBUS) audit, the spare line card (SPRCRD) audit, and the ISLU transaction (ISLUTRAN) audit. The RGTDDB audit checks the ringing task data block information.

The RGTDDB audit is part of the peripheral control audits subgroup. It is part of the ISLU application loadable package (ISLUAPP). It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and non-segmented modes.

2. DATA STRUCTURES

The RGTDDB audit uses the peripheral control ringing task data block array (PCrgtdb) and the peripheral control ringing occupancy table array (PCrgocup). PCrgtdb is a member of the PCrgocup array. Four tuples of PCrgtdb may be linked to each PCrgocup time slot (tuple), one for each task to be sent to the ISLU for ringing. These data structures are shown in Figure 1.

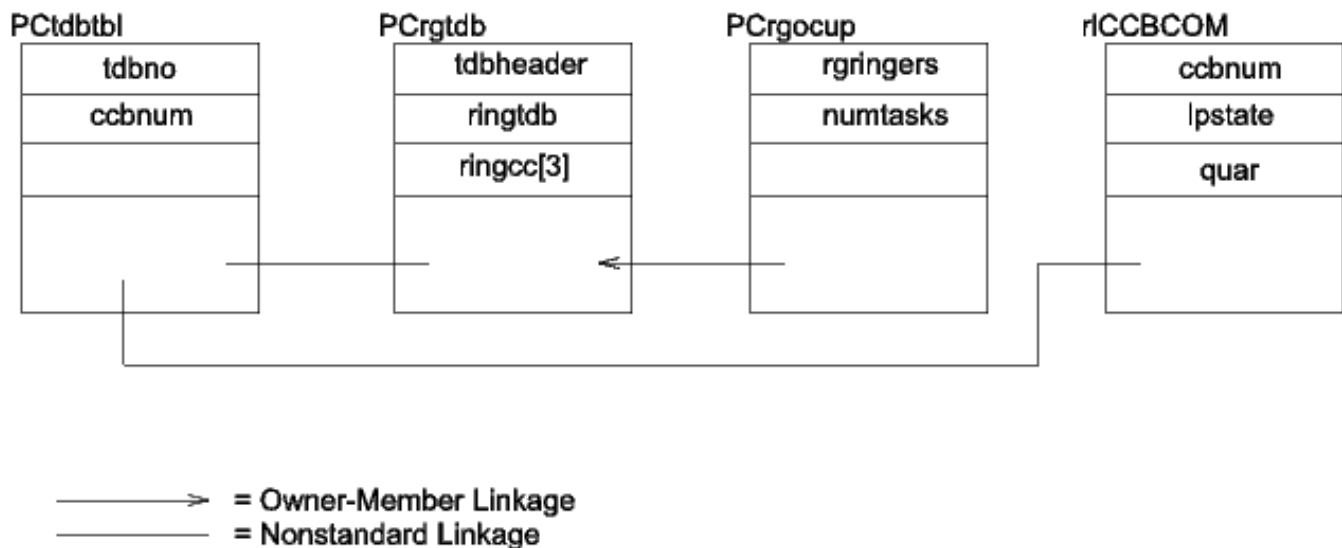


Figure 1 (RGTDDB-1) Data Structures of the RGTDDB Audit

3. AUDIT CHECKS

The RGTDDB audit invokes the linkage check function of the linkage driver (DRLINK) to perform point-to/point-back linkage checks between PCrgocup and PCrgtdb. When the linkage checks pass, the audit invokes the semantic check function to make the following checks.

- GRBG = Verify associated rCCBCOM exists).
- STATUS = Verify associated rCCBCOM key in rIRGTDDB).
- CSTATE = Ensure that for each busy ringccb[3] entry the local port state field (lpstate) in rCCBCOM with the

same ccbcom number is PACTIVE, PCRING or PCPXSTATE and is not idle).

INV_STATE = Ensure that for each busy tdbno field in PCrgrtdb the channel control block field in PCtdbtbl is PCRGCCB).

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CSTATE

Error Code:	CSTATE
Error Description:	A ringccb is marked busy in this pcRGTDDB tuple but the 1pstate of the corresponding rICCB COM tuple is not marked as PCRING, PCPXSTATE, PACTIVE, or the rICCB COM is idle.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	The tuple is recovered, and the rgringers field in the owning pcRGOCUP tuple is decremented.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the pcRGTDDB tuple.
<i>Bad Data =</i>	ccbnum.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of tuple which is also the physical.
<i>Dump 1 =</i>	The pcRGTDDB tuple.
<i>Dump 2 =</i>	The rICCB COM tuple.

Table 2 Error Code: GRBG

Error Code:	GRBG
Error Description:	The rICCB COM whose key was found in PCrgrtdb does not exist.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	The tuple is recovered, and the rgringers field in the owning pcRGOCUP tuple is decremented.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the pcRGTDDB tuple.
<i>Bad Data =</i>	ccbnum.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of tuple which is also the physical.
<i>Dump 1 =</i>	The pcRGTDDB tuple.
<i>Dump 2 =</i>	The rICCB COM tuple.

Table 3 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The ccbnum in the pcTDBTBL tuple that is indexed by the tdbno in tdbheader is not marked PCRGCCB.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	The appropriate pcRGOCUP tuple(s) are recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the pcRGTDDB tuple.
<i>Bad Data =</i>	ccbnum.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of tuple which is also the physical.
<i>Dump =</i>	The pcRGTDDB tuple.

Table 4 Error Code: STATUS

Error Code:	STATUS
Error Description:	The DAgetccb primitive returned a garbaged pointer.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	The PCgtadb tuple is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the pcRGTDDB tuple.
<i>Bad Data</i> =	Ringing ccb.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple which is also the physical.
<i>Dump 1</i> =	The pcRGTDDB tuple.
<i>Dump 2</i> =	The rICBCOM tuple.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The RGTDB has been entered in ZAP mode. The audit will be terminated and will start from the beginning the next time it is scheduled to run.
Possible Error Effect:	The RGTDB audit is terminated before it completes all its checks.
Corrective Action Taken:	The audit will start from its beginning the next time it is scheduled to run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	0.
<i>Dump</i> =	NA.

RLHPTPN

Software Release: 5E10 and later

1. REMOTE MEASUREMENT SYSTEM LIST HEAD OF PORT-TYPE-PORT-NUMBER (RLHPTPN) AUDIT

The remote measurement system list head of port-type-port-number audit audits the RLR_LHPTPN dynamic relation.

2. DATA STRUCTURES

For details on the fields contained in the RLR_LHPTPN relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

Relation RLR_LHPTN is the owner of relation RLR_SPTPN. Linkage checking is done by the linkage driver function which will restore the key attribute and null out the tuple if linkage problems are found or if the tuple is missing. The semantic check determines if the value of the number of tuples on each head list attribute agrees with the number of tuples in relation RLR_SPTPN and corrects the attribute if necessary.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NUMELEM

Error Code:	NUMELEM
Error Description:	The number of elements in the headlist does not agree with the "elems" attribute in relation rLR_LHPTPN.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The number of elements is updated in the relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rLR_LHPTPN tuple.
<i>Bad Data</i> =	Actual number of elements.
<i>Good Data</i> =	Expected number of elements.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rLR_LHPTPN tuple.

RMB

Software Release: 5E10 and later

1. RECEIVE MESSAGE BUFFER (RMB) AUDIT

The RMB audit performs some checks on the dynamic relation RLRMB. The tuples of RLRMB are temporary storage buffers for packets received from the applications processor (AP) destined for the applications processor system process (APSP).

The RMB audit uses the indirect access sequential key driver (DRINASEK) to traverse through the RLRMB relation tuple by tuple. The key and quarantine bit are checked for each tuple. The audit does not look at the other attributes in the tuple. Users of this other information do self auditing on the data.

2. DATA STRUCTURES

The RLRMB is the dynamic relation of temporary storage buffers for packets received from an AP destined for the APSP.

The RMB audit runs in the operating system for distributed switching-switching module (OSDS-SM) environment.

3. AUDIT CHECKS

The RMB audit insures that the key in each rLRMB tuple is correct and a check of the quarantine bit is made.

The indirect access sequential key driver (DRINASEK) is used to step through the RLRMB relation tuple by tuple. The audit is entered from audit control.

4. ERROR HANDLING

If an error is detected, the DRINASEK driver calls the function to zero out the tuple and put the key back in.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes.

RMCATPS

Software Release: 5E10 and later

1. REMOTE MEASUREMENT SYSTEM MASTER CONTROL AREA TEST POSITION STATUS (RMCATPS) AUDIT

The remote measurement system master control area test position status (RMCATPS) audit performs semantic checks for the RLR_MCATPS relation.

2. DATA STRUCTURES

The RMCATPS audit checks and verifies the RLR_MCATPS dynamic relation.

3. AUDIT CHECKS

The linkage driver checks linkage and restores the key attribute and nulls out the tuple if linkage problems are found. Semantic checks are run to see if the value of the test position attribute is consistent with the RLR_CATPS relation and correcting the attribute if necessary. Another semantic check ensures that the value of the number of control areas attribute agrees with the relation RLR_CATPS and correcting the attribute if it is incorrect.

4. ERROR HANDLING

This audit has a linkage driver function which will restore the key attribute and null out the tuple if linkage problems are found.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NUAP

Error Code:	NUAP
Error Description:	An rIR_MCATPS tuple was found to have an incorrect value in the "attended test positions" attribute.
Possible Error Effect:	The tuple contains the wrong information.
Corrective Action Taken:	The attribute is updated with correct information.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_MCATPS tuple.
<i>Bad Data</i> =	Actual value of "attended test positions" attribute.
<i>Good Data</i> =	Corrected value for same attribute.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_MCATPS tuple.

Table 2 Error Code: NUCA

Error Code:	NUCA
Error Description:	An rIR_MCATPS tuple was found to have an incorrect value in the "number of control areas" attribute.
Possible Error Effect:	The tuple contains the wrong information.
Corrective Action Taken:	The attribute is updated with correct information.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_MCATPS tuple.
<i>Bad Data</i> =	Actual value of "number of control areas" attribute.
<i>Good Data</i> =	Corrected value for same attribute.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_MCATPS tuple.

RQKEYPT

Software Release: 5E10 and later

1. REMOTE MEASUREMENT SYSTEM GLOBAL PORT NUMBER TO RANDOM MAPPING NUMBER (RQKEYPT) AUDIT

This audit uses a static driver to step through each tuple in relation RLR_QKEYPT.

2. DATA STRUCTURES

For details on the fields contained in the RLR_QKEYPT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

Each rIR_QKEYPT tuple should have a corresponding rIR_SPTPN tuple. If necessary, a new rIR_SPTPN tuple is created. If the GPORTs of the static tuple and dynamic tuple do not agree, the GPORT in the dynamic tuple must be fixed and the rest of the tuple must be nulled.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: GPT1

Error Code:	GPT1
Error Description:	The module and port (GPORT) do not agree between static relation RLR_QKEYPT and dynamic relation RLR_SPTPN.
Possible Error Effect:	The tuple contains the wrong information.
Corrective Action Taken:	The dynamic rIR_SPTPN tuple is altered so that the module and port are made to match those of the static tuple in relation RLR_QKEYPT. The type, idlestatus, and testposition attributes in dynamic relation RLR_SPTPN are also reset.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIR_SPTPN tuple.
<i>Bad Data =</i>	Key of rIR_SPTPN tuple.
<i>Good Data =</i>	Key of rIR_QKEYPT tuple.
<i>Logical Index =</i>	Logical key of rIR_SPTPN tuple.
<i>Dump =</i>	rIR_SPTPN tuple.

Table 2 Error Code: SPLOSTUP

Error Code:	SPLOSTUP
Error Description:	A dynamic rIR_SPTPN tuple does not exist that corresponds with a tuple in static relation RLR_QKEYPT.
Possible Error Effect:	The tuple contains the wrong information.
Corrective Action Taken:	A new rIR_SPTPN tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIR_SPTPN tuple.

<i>Bad Data</i> =	Key of the rIR_SPTPN tuple.
<i>Good Data</i> =	Key of the new rIR_QKEYPT tuple.
<i>Logical Index</i> =	Logical key of the rIR_SPTPN tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

RSPTPN

Software Release: 5E10 and later

1. REMOTE MEASUREMENT SYSTEM PORT TYPE TO PORT NUMBER (RSPTPN) AUDIT

The RSPTPN audit audits dynamic relation RLR_SPTPN.

2. DATA STRUCTURES

Relation RLR_SPTPN is a member of relation RLR_LHPTPN. Linkage checking is done by the linkage driver function which will restore the key attribute and null out the tuple if linkage problems are found.

3. AUDIT CHECKS

The semantic check, performed by function sr_sptpn, does a lot of verification as follows:

- (1) it makes sure that for each tuple in relation RLR_SPTPN there is a corresponding tuple in static relation RLR_QKEYPT,
- (2) it ensures that the GPORTs and RQKEYS between relation RLR_QKEYPT and RLR_SPTPN are identical,
- (3) it checks that if the rIR_SPTPN tuple is not of type RTPOS, then the test position attribute should be null,
- (4) it ensures that if the test position type is TOPAS test position and the test position is TNULL, then the idle status and the type are nulled,
- (5) it checks to see that if the tstpos attribute is not TNULL, then there is a corresponding tuple with the same GPORT in relation RLR_TPSTAT,
- (6) it checks to see that if the tuple is on the wrong head list as indicated in relation RLR_LHPTPN, it is removed from that list and placed on the correct list, or none at all, if appropriate, and
- (7) it checks to see that if the tuple is not on any list and should be, then it will be added to the correct head list.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ADDHLIST

Error Code:	ADDHLIST
Error Description:	An rIR_SPTPN tuple should have been found on any headlist of relation RLR_LHPTPN.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The tuple is placed on the correct headlist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIR_SPTPN tuple.
<i>Bad Data =</i>	Actual value of owner link.
<i>Good Data =</i>	Expected value of owner link.
<i>Logical Index =</i>	Logical key of tuple.
<i>Dump =</i>	rIR_SPTPN tuple.

Table 2 Error Code: BADTESTPOS

Error Code:	BADTESTPOS
Error Description:	A tuple in relation RLR_SPTPN has a non TOPAS type attribute and the test position attribute is not null and should be.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The test position attribute is made null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Value of test position attribute.
<i>Good Data</i> =	Expected value of test position attribute.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	A tuple in relation RLR_SPTPN is a TOPAS test position type and the test position is null, however, the idle status and type attributes are not null.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The idle status and type attributes are set to the null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Value of idle status.
<i>Good Data</i> =	Expected value of idle status.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

Table 4 Error Code: CHGHLIST

Error Code:	CHGHLIST
Error Description:	A rIR_SPTPN tuple was found on an incorrect headlist of relation RLR_LHPTPN.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The tuple is removed from the wrong headlist and placed on the correct one.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Actual value of owner link.
<i>Good Data</i> =	Expected value of owner link.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

Table 5 Error Code: GPT1

Error Code:	GPT1
Error Description:	The module and port (GPORT) do not agree between static relation RLR_QKEYPT and dynamic relation RLR_SPTPN.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The test position, idle status, type and module and port attributes in the rIR_SPTPN tuple are set to their null values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Module attribute of RLR_SPTPN relation.
<i>Good Data</i> =	Module attribute of RLR_QKEYPT relation.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

Table 6 Error Code: GPT2

Error Code:	GPT2
Error Description:	The corresponding tuple, based on test position, was found in relation RLR_TPSTAT but the module and port attributes (GPORT) do not match those of relation RLR_SPTPN.

Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The tuple in RLR_TPSTAT has its qbit set and if linked to owner tuples, it is unlinked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Module attribute of RLR_SPTPN relation.
<i>Good Data</i> =	Module attribute of RLR_QKEYPT relation.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

Table 7 Error Code: RMHLIST

Error Code:	RMHLIST
Error Description:	A rIR_SPTPN tuple was found on a headlist of relation RLR_LHPTPN and should not have been.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	The tuple is removed from the headlist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Actual value of owner link.
<i>Good Data</i> =	Expected value of owner link.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

Table 8 Error Code: SPLOSTUP

Error Code:	SPLOSTUP
Error Description:	A tuple in relation RLR_SPTPN has no corresponding tuple in static relation RLR_QKEYPT.
Possible Error Effect:	This is an extra tuple.
Corrective Action Taken:	The rIR_SPTPN tuple is removed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_SPTPN tuple.
<i>Bad Data</i> =	Key of rIR_SPTPN tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	rIR_SPTPN tuple.

RSTMR

Software Release: 5E11 and later

1. ROUTE SET TIMING LIST (RSTMR) AUDIT

RSTMR audit steps through the RLRSTMR (rout set timing list) relation. For each tuple, the audit checks the linkage integrity between the RLRSTMR and RLS7RDB (CCS7 routing data block) relations.

2. DATA STRUCTURES

For details on the fields contained in the RLRSTMR relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

29. RT - SC

RT303DS1

Software Release: 5E10 and later

1. INTEGRATED DIGITAL SERVICE UNIT 303 RT TO DS1 (RT303DS1) AUDIT

The RT303DS1 audit will perform the semantic checks of the PC303rt2ds1[][] array. The array tells the IDCU which of the 40 integrated digital carrier unit (IDCU) digital facilities (DS1) are connected to a particular TR303 type RT (from the IDCU perspective).

2. DATA STRUCTURES

The data structure being audited by RT303DS1 audit is the PC303rt2ds1[][] array. It is a doubly indexed array (PC303rt2ds1[idcu][rt]) where the indexes are the IDCU number and the TR303 RT number. The array components (fields and a brief description) is as follows:

PC303rt2ds1[][].lowhalf = From an IDCU perspective which one of the first group of 20 DS1s are connected to this RT. A "1" means it is connected, a "0" means it is not.

PC303rt2ds1[][].highhalf = From an IDCU perspective which one of the second group of 20 DS1s are connected to this RT. A "1" means it is connected, a "0" means it is not.

3. AUDIT CHECKS

The RT303DS1 audit will perform the semantic checks of the PC303rt2ds1[][] array fields. The audit sequence is as follows.

ZAP = The audit entry mode is verified.

LOWHALF = The PC303rt2ds1[][].lowhalf is audited against the rIRT2DS1 tuple.

HIGHHALF = The PC303rt2ds1[][].highhalf is audited against the rIRT2DS1 tuple.

4. ERROR HANDLING

The general audit error handling mechanism is to print the error code and make the PC303rt2ds1[][] array consistent with the data that it is being compared against (such as, rIRT2DS1 tuple). If a user intervention is required (such as in the case of a missing static data) then a user intervention error code will be printed.

The audit produces the following error code(s).

Table 1 Error Code: HIGHHALF

Error Code:	HIGHHALF
Error Description:	The RT303DS1 audit found the PC303rt2ds1[][].highhalf to be inconsistent with what the rIRT2DS1 tuple states it should be.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	The PC303rt2ds1[][].highhalf will be made consistent with the information derived from the static rIRT2DS1 tuple.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address</i> =	The address of the PC303rt2ds1[][] array.
<i>Bad Data</i> =	The contents of the PC303rt2ds1[][].lowhalf field.
<i>Good Data</i> =	The contents of what the PC303rt2ds1[][].lowhalf field should be.
<i>Logical Index</i> =	The IDCU number.
<i>Dump 1</i> =	The contents of the PC303rt2ds1[][] array.
<i>Dump 2</i> =	The contents of the rIRT2DS1 tuple.

Table 2 Error Code: LOWHALF

Error Code:	LOWHALF
Error Description:	The RT303DS1 audit found the PC303rt2ds1[][].lowhalf to be inconsistent with what the rIRT2DS1 tuple states it should be.
Possible Error Effect:	The call processing will not be able to use that particular DS1 facility.
Corrective Action Taken:	The PC303rt2ds1[][].lowhalf will be made consistent with the information derived from the static rIRT2DS1 tuple.
Manual Action Required:	NA.
Dump Description:	
<i>Error Address =</i>	The address of the PC303rt2ds1[][] array.
<i>Bad Data =</i>	The contents of the PC303rt2ds1[][].lowhalf field.
<i>Good Data =</i>	The contents of what the PC303rt2ds1[][].lowhalf field should be.
<i>Logical Index =</i>	The IDC number.
<i>Dump 1 =</i>	The contents of the PC303rt2ds1[][] array.
<i>Dump 2 =</i>	The contents of the rIRT2DS1 tuple.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to provide recovery after taking a reset. When a reset occurs (usually triggered by an illegal or out of range memory access), the audit which was running at the time is aborted, and is then rescheduled to run in AUZAP mode to provide recovery of related data left inconsistent as a result of the reset.
Possible Error Effect:	The audit may continue to cause memory access violations, increasing the probability of drastic recovery action and preventing the audit from checking for further errors.
Corrective Action Taken:	The data which caused the reset is initialized by the next entry of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

RTDB

Software Release: 5E10 and later

1. COMMON CHANNEL ROUTING DATA BLOCK (RTDB) AUDIT

The channel selection feature allows a user to access multiple channels on the basic rate interface (BRI) for packet mode calling with the same directory number (DN). The common channel routing data block array (PSrtdb[]) and the common channel routing data block array index (PSrtdbix[][]) are used by the channel selection algorithm to determine the channels assigned to a user.

2. DATA STRUCTURES

The RTDB audit references the PSrtdb[] and the PSrtdbix[][] arrays. These data structures are used by the channel selection algorithm to determine if a user is subscribed to the channel selection feature. Figure 1 shows an entry in the PSrtdb[] array. Figure 2 shows how the PSrtdbix[][] array is laid out. A PSrtdb[].chan[0] entry is used for the D-channel, while PSrtdb[].chan[1] and PSrtdb[].chan[2] are used for the B-channels. The PSrtdbix[][] array content is the index into the PSrtdb[] array. The D-channel uses parties 0 through 7, while the B-channels use party 0 only. To index into the PSrtdbix[][] array for the D channel, the chan[0].pcn and the party fields are used. For the B channel, the chan[1].pcn or chan[2].pcn and 0 are used.

PSRTDB[PSMXRTDB]

chan[0]	phcn	assigned	cctype	routable
chan[1]	phcn	assigned	cctype	routable
chan[2]	phcn	assigned	cctype	routable
fdrtdb_ind				
bkrtdb_ind				
tcld				
mlhg_flag				
party				

Figure 1 (RTDB-1)

PSrtdbix[PSMAXCHN][PSDMXNTEI]

party CHANNEL	0	1	2	3	4	5	6	7
0								
1								
2								
3								
⋮								
20								
⋮								
46								
⋮								
96								
⋮								
127								

Figure 2 (RTDB-1)

3. AUDIT CHECKS

For each tuple, the RTDB audit makes the following checks:

If (the tuple is on the free list)

OUTOFRANGE = Check the tuple's forward and backward linkages.

Otherwise:

NON_NULL = Check the tuple's forward and backward linkages.

BADUSE	= Check range of channels in RTDB.
AUD_CH	= Check number of channels in RTDB.
PIXOUTOFRNG	= Check range of the party field.
CLRCH	= Check RTDB B-channel number.
T1_TIME_SLOT	= Check RTDB B-channel time slot.
POINT2POINT	= Check linkage to RTDBIX.
TYPE	= Check MLHG flag.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: AUD_CH

Error Code:	AUD_CH
Error Description:	An RTDB entry has fewer than two channels assigned to it.
Possible Error Effect:	Channel selection calls could fail.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PSrtdb[] entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	RTDB index.
<i>Dump</i> =	The PSrtdb[] entry.

Table 2 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An RTDB entry has an assigned channel out of range (PSrtdb[].chan[].phcn ≥ PSXMAXCHAN).
Possible Error Effect:	Channel selection calls will route to an out of range channel and calls will fail.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PSrtdb[] entry.
<i>Bad Data</i> =	The out of range channel (PSrtdb[].chan[].phcn).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	RTDB index.
<i>Dump</i> =	The PSrtdb[] entry.

Table 3 Error Code: CLRCH

Error Code:	CLRCH
Error Description:	An RTDB entry has a B-channel that is not a multiple of four (PSrtdb[].chan[].phcn). B-channels assigned to the RTDB must be multiples of four.
Possible Error Effect:	Channel Selection calls will route incorrectly and may fail.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the PSrtdb[] entry.
<i>Bad Data</i> =	The invalid B-channel (PSrtdb[].chan[].phcn).
<i>Good Data</i> =	The index into the channel array (PSrtdb[].chan[]).
<i>Logical Index</i> =	RTDB index.
<i>Dump</i> =	The PSrtdb[] entry.

Table 4 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	An RTDB entry that is not on the free list has either its forward or backward index set to a non-null value (PSrtdb[].fdrtdb_ind != PSRTDBNULL or PSrtdb[].bkrtdb_ind != PSRTDBNULL).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PSrtdb[] entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	PSRTDBNULL.
<i>Logical Index</i> =	RTDB index.
<i>Dump</i> =	The PSrtdb[] entry.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An RTDB entry on the free list has either its forward or backward index out of range (PSrtdb[].fdrtdb_ind > PSRTDBNULL or PSrtdb[].bkrtdb_ind > PSRTDBNULL).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit calls the head cell recovery function to reconstruct the free linked list. The recovery function will step through the PSrtdb[] array and recover to the list any entry that has no channels assigned to it (PSrtdb[].chan[].assigned == DBNO).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PSrtdb[] entry.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	RTDB index.
<i>Dump 1</i> =	The current RTDB entry.
<i>Dump 2</i> =	The previous RTDB entry (if within range).
<i>Dump 3</i> =	The next RTDB entry (if within range).

Table 6 Error Code: PIXOUTOFRNG

Error Code:	PIXOUTOFRNG
Error Description:	An RTDB has the party field out of range (PSrtdb[].chan[].party ≥ FCP_NULL). The party field is only used for the D-channel (PSrtdb[].chan[0]).
Possible Error Effect:	Channel selection calls will route incorrectly and may fail.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The logical PH channel number for the D-channel (PSrtdb[].chan[0].phcn).
<i>Bad Data</i> =	PSrtdb[].chan[0].party.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	RTDB index.
<i>Dump</i> =	The RTDB entry.

Table 7 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The point-to-point-back linkage checks between the PSrtdb[] and the PSrtdbix[]

	[] arrays failed.
Possible Error Effect:	Channel selection calls will fail.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The index into the RTDB channel array (PSrtdb[].chan[]).
<i>Bad Data =</i>	The logical PH channel number assigned to the RTDB entry (PSrtdb[].chan[].phcn).
<i>Good Data =</i>	PSrtdb[].chan[].party.
<i>Logical Index =</i>	RTDB index.
<i>Dump =</i>	The RTDB entry.

Table 8 Error Code: T1_TIME_SLOT

Error Code:	T1_TIME_SLOT
Error Description:	An RTDB B-channel does not use an entire timeslot. If a B-channel is assigned to an RTDB entry, that channel should be a multiple of four and should use the entire timeslot. That is, all the remaining parties for the B-channel (1 through 7 with 0 used by the B-channel) and all the parties for the next three channels in the timeslot should not be used.
Possible Error Effect:	Channel selection calls will route incorrectly and may fail.
Corrective Action Taken:	The RTDB will be recovered to the free list and routine port conditioning will be requested to refresh the PH data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PSrtdb[] entry.
<i>Bad Data =</i>	The RTDB B-channel (PSrtdb[].chan[1].phcn or PSrtdb[].chan[2].phcn).
<i>Good Data =</i>	Party number from PSrtdbix[][] where the error was found. The party will have the following ranges: 1 - 7 = Parties for the first channel in the timeslot. 8 - 15 = Parties for the second channel in the timeslot. 16 - 23 = Parties for the third channel in the timeslot. 24 - 31 = Parties for the fourth channel in the timeslot.
<i>Logical Index =</i>	RTDB index.
<i>Dump =</i>	The RTDB entry.

Table 9 Error Code: TYPE

Error Code:	TYPE
Error Description:	An RTDB that has the MLHG flag set (PSrtdb[].mlhg_flag == DBYES) has no routable channel assigned to it (PSrtdb[].chan[].routable == DBNO). When the MLHG flag is set for an RTDB entry, at least one assigned channel should be routable.
Possible Error Effect:	This multi-line hunt group (MLHG) member will not be able to receive any MLHG calls.
Corrective Action Taken:	None.
Manual Action Required:	(1) Using ODBE read RLSPRT_GP using the following keys: module = SM number. ph_number = DSL group number. ph_cn = PH channel number. party = 0 (ZERO).

	(2) Get the grpnum and member number from RLPSPT_GP. (3) Using RCV go to view 23.2 using the following keys: MLHG = grpnum from RLPSPT_GP. TERM = member from RLPSPT_GP. (4) Check the following packet services to find the channel associated with this member: DPKT, ODB, PPB1, and PPB2. Go to each of these packet services pages. (5) Check if MLHG and TERM are equal to the keys used in view 23.2. • If yes, this channel is part of the MLHG member. Make sure there is at least one IN or 2W LCN assigned. If not, check the remaining packet services to see if other channels are part of the MLHG member. If no packet services has one IN or 2W LCN, choose one and assign an INC or 2W LCN. • Otherwise, go to the other packet services pages to find the channel that is part of the MLHG member. (6) If no problem was detected in the procedure above, check the ICB field for all packet services. If it is set to YES, set it to NO.
Dump Description:	
<i>Error Address</i> =	The preferred B-channel (PSrtdb[].chan[1].phcn).
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	RTDB index.
<i>Dump</i> =	The RTDB entry.

RTDBIX

Software Release: 5E10 and later

1. COMMON CHANNEL ROUTING DATA BLOCK INDEX (RTDBIX) AUDIT

The channel selection feature allows a user to access multiple channels on the basic rate interface (BRI) for packet mode calling with the same directory number (DN). The common channel routing data block index array (PSrtdbix[][]) and the common channel routing data block array (PSrtdb[]) are used by the channel selection algorithm to determine the channels assigned to a user.

2. DATA STRUCTURES

The RTDBIX audit references the PSrtdbix[][] and the PSrtdb[] arrays. These data structures are used by the channel selection algorithm to determine if a user is subscribed to the channel selection feature. Figure 1 shows an entry in the PSrtdb[] array. Figure 2 shows how the PSrtdbix[][] array is laid out. A PSrtdb[].chan[0] entry is used for the D-channel, while PSrtdb[].chan[1] and PSrtdb[].chan[2] are used for the B-channels. The PSrtdbix[][] array content is the index into the PSrtdb[] array. The D-channel uses parties 0 through 7, while the B-channels use party 0 only. To index into the PSrtdbix[][] array for the D-channel, the chan[0].phcn and the party fields are used. For the B-channel, the chan[1].phcn or chan[2].phcn and 0 are used.

3. AUDIT CHECKS

For each tuple, the RTDBIX audit makes the following checks:

LOGCHAN = Check RTDBIX assignment.

POINT2POINT = Check linkage to RTDB.

OUTOFRANGE = Check range of RTDB.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOGCHAN

Error Code:	LOGCHAN
Error Description:	An RTDBIX entry has been assigned more than once to an RTDB.
Possible Error Effect:	Channel selection calls will route incorrectly and may fail.
Corrective Action Taken:	The audit will null out the RTDBIX entry and request routine port conditioning for the PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index into the RTDB channel array (PSrtdb[].chan[]).
<i>Bad Data</i> =	RTDBIX party number.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical PH channel number.
<i>Dump</i> =	None.

Table 2 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The RTDB assigned to the RTDBIX entry is out of range.
Possible Error Effect:	Channel selection calls may fail.

Corrective Action Taken:	The audit will null out the RTDBIX entry and request routine port conditioning for the PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	RTDBIX party.
<i>Bad Data</i> =	The out of range RTDB assigned to the RTDBIX.
<i>Good Data</i> =	PSRTDBNULL
<i>Logical Index</i> =	Logical PH channel number.
<i>Dump</i> =	None.

Table 3 Error Code: POINT2POINT

Error Code:	POINT2POINT
Error Description:	The point-to-point-back linkage checks between the PSrtdbx[][] and the PSrtdb[] arrays failed.
Possible Error Effect:	Channel selection calls may fail.
Corrective Action Taken:	The audit will null out the RTDBIX entry and request routine port conditioning for the PH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	RTDBIX party.
<i>Bad Data</i> =	Invalid RTDB assigned to the RTDBIX.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical PH channel number.
<i>Dump</i> =	None.

RTEILA

Software Release: 5E10 and later

1. REMOTE TERMINAL ENDPOINT IDENTIFIER LINKAGE AREA (RTEILA) AUDIT

For a TR303 remote terminal (RT) on an integrated digital carrier unit (IDCU) two channels are used for the message link through the protocol handler (PH). They are the embedded operations channel (EOC) and timeslot management channel (TMC). These are dual channels with an active/standby status. Port names have been defined but no user messages will report the status which is only available from the MCC page. The outgoing and incoming messages are linked to the rRTEILA tuple which is always linked to the rIDPB for the EOC/TMC channels.

The RTEILA audit maintains the integrity of these message queues and the rIDPB and rRTEILA linkages are preserved at all times.

The dynamic data structures for EOC/TMC message processing on TR303 RT are shown in Figure 1 .

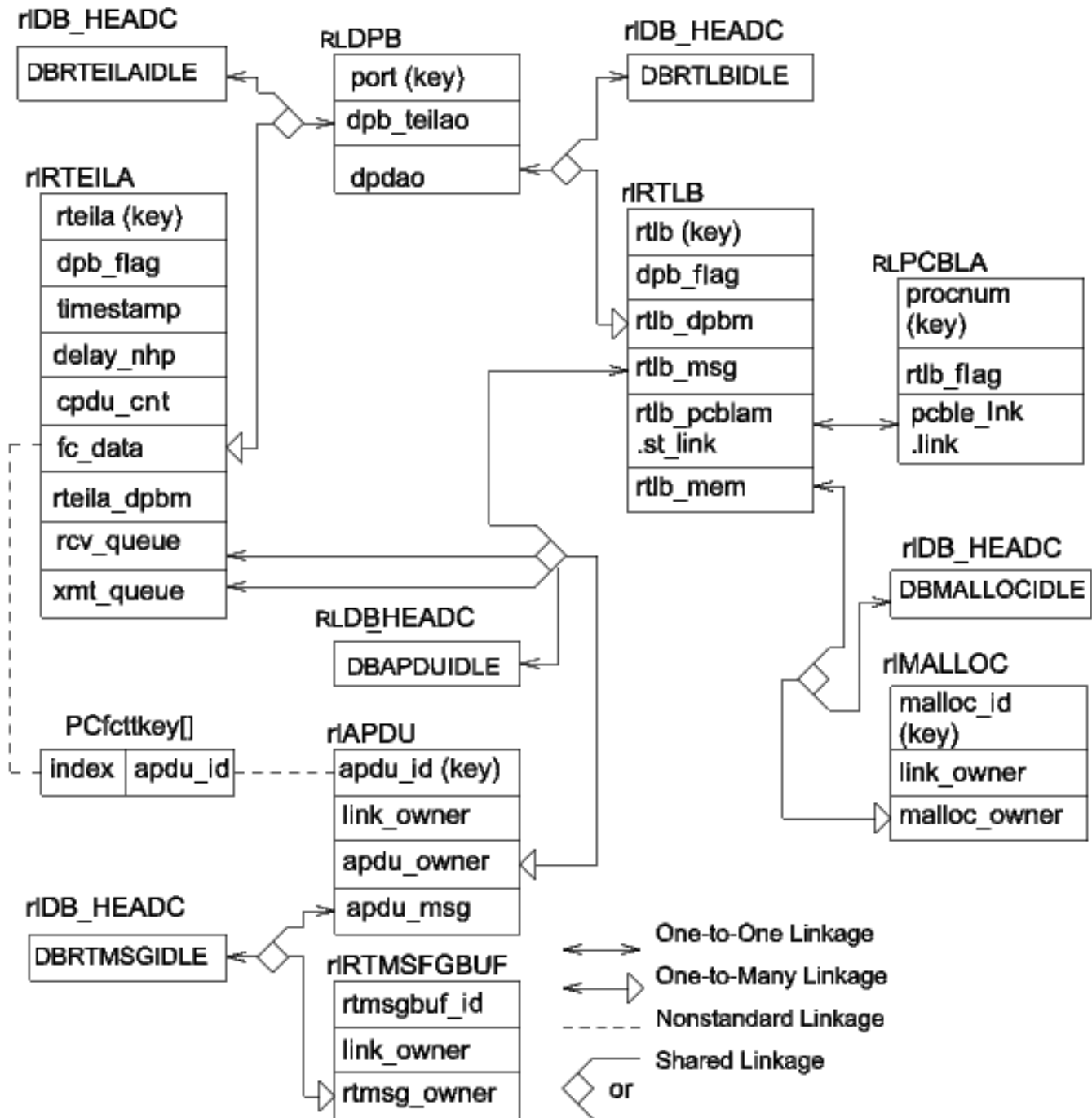


Figure 1 (RTEILA-1) Dynamic Data Structures for EOC/TMC Message Processing on TR30 RT

2. DATA STRUCTURES

Extra rRTEILA tuples are linked to the idle list (rDB_HEADC). Each active or standby EOC/TMC channel (rLDPB) has a linkage to a rRTEILA tuple. This rRTEILA tuple will be linked to all messages (rAPDU) received from the TR 303 RT or messages ready to transmit to the TR 303 RT for this channel. The dpb_flag is used to determine the correct linkage.

When messages are on the transmit queue, a flow control process uses other fields in the rRTEILA to maintain

control. If a timer is running for a message on the transmit queue, there is an entry in PCfcttkey[] with the key to the rIAPDU tuple containing the message information.

3. AUDIT CHECKS

The RTEILA audit invokes the DRINASEK driver to make the linkages and semantic checks for each rIRTEILA tuple. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given. For each tuple, the RTEILA audit makes the following checks:

MEMBAD	= Check member linkage to rIDPB or rIDB_HEADC.
COUNTS	= Check transmit queue and rIAPDU linkage.
LOST_PORTLA	= Check that rIORTLA exists for this EOC/TMC channel.
PROC_LOST	= Check that a process owns the rIORTLA tuple.
MISSTUP	= Check for head rIAPDU tuple for this message queue.
INV_TIMER	= Check that if messages are being transmitted then a timer is active.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	If the APDU count is equal to zero, no rIAPDU tuples should be linked to this rIRTEILA. If the APDU count is greater than zero, at least one rIAPDU tuple should be linked to this rIRTEILA. No check is made to verify the APDU count with the exact number of rIAPDU tuples linked to this rIRTEILA tuple.
Possible Error Effect:	Messages could be lost and the task process could be stuck.
Corrective Action Taken:	The APDU count is zeroed and the linkage to the rIAPDU cleared. This will also purge the process handling the messages.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRTEILA tuple.
<i>Bad Data</i> =	APDU count (rIRTEILA.apdu_cnt).
<i>Good Data</i> =	Linkage to rIAPDU tuple.
<i>Logical Index</i> =	Key of rIRTEILA (rIRTEILA.rteila)
<i>Dump</i> =	Contents of rIRTEILA tuple.

Table 2 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	There are messages on the transmit queue but no timer is active.
Possible Error Effect:	Messages may be stuck on the transmit queue.
Corrective Action Taken:	This EOC/TMC channel will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIAPDU tuple.
<i>Bad Data</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id).
<i>Good Data</i> =	Key to timer table (rIAPDU.fc_data).
<i>Logical Index</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id).
<i>Dump 1</i> =	Contents of rIRTEILA tuple.
<i>Dump 2</i> =	Contents of rIAPDU tuple.

Table 3 Error Code: LOST_PORTLA

Error Code:	LOST_PORTLA
Error Description:	The rIORTLA tuple for this EOC/TMC channel could not be found.
Possible Error Effect:	All message traffic for this channel is invalid.
Corrective Action Taken:	This EOC/TMC channel will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRTEILA tuple.
<i>Bad Data</i> =	rIDPB owner of rIRTEILA tuple (rIRTEILA.rteila_dpbm.st_ownk).
<i>Good Data</i> =	Key of Port used to read rIORTLA tuple.
<i>Logical Index</i> =	Key of rIRTEILA (rIRTEILA.rteila)
<i>Dump</i> =	Contents of rIRTEILA tuple.

Table 4 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	This rIRTEILA tuple is not linked to the owner indicated by the flag (rIRTEILA.dpb_flag).
Possible Error Effect:	The message for this channel could be dropped.
Corrective Action Taken:	This EOC/TMC channel will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRTEILA tuple.
<i>Bad Data</i> =	Key of owner tuple (rIDPB or rIDB_HEADC).
<i>Good Data</i> =	Flag for owner field (1 for rIDPB or 0 for rIDB_HEADC).
<i>Logical Index</i> =	Key of rIRTEILA (rIRTEILA.rteila)
<i>Dump 1</i> =	Contents of rIRTEILA tuple.
<i>Dump 2</i> =	Contents of either rIDPB or rIDB_HEADC.

Table 5 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	This rIRTEILA should have rIAPDU tuples linked on the transmit queue but the database cannot find the rIAPDU tuple.
Possible Error Effect:	Messages cannot be transmitted over this channel.
Corrective Action Taken:	This EOC/TMC channel will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRTEILA tuple.
<i>Bad Data</i> =	Key of rIAPDU tuple (rIAPDU.apdu_id).
<i>Good Data</i> =	Linkage to rIAPDU (rIRTEILA.xnt_queue.st_headk).
<i>Logical Index</i> =	Key of rIRTEILA (rIRTEILA.rteila)
<i>Dump</i> =	Contents of rIRTEILA tuple.

Table 6 Error Code: PROC_LOST

Error Code:	PROC_LOST
Error Description:	The message handler process is transmitting messages on this channel and should own the rIORTLA tuple.
Possible Error Effect:	Messages are not being transmitted over correct channel or messages are not being transmitted at all.
Corrective Action Taken:	This EOC/TMC channel will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIORTLA tuple.
<i>Bad Data</i> =	Process link field (rIORTLA.proc_cplr.st_headk = 0)
<i>Good Data</i> =	rIRTEILA for this port (rIRTEILA.rteila)
<i>Logical Index</i> =	Key of rIORTLA tuple (rIORTLA.port)
<i>Dump</i> =	Contents of rIORTLA.

RTLb

Software Release: 5E10 and later

1. REMOTE TERMINAL LINKAGE BLOCK (RTLb) AUDIT

For a TR303 remote terminal (RT) on integrated digital carrier Unit (IDCU) the embedded operations channel (EOC) is used by the Level 7 administration to process tasks using messages. The rRTLb tuple is linked to the EOC channels (rIDPB) and the task terminal process (rIPCBLA).

The RTLb audit maintains the integrity of these linkages.

The dynamic data structures for EOC/TMC message processing on TR303 RT are shown in Figure 1 .

2. DATA STRUCTURES

The rRTLb tuple is linked to either the idle list (rIDB_HEADC) or the message channel (RLDPB) when processing messages. The dpb_flag in the rRTLb tuple is used to determine which linkage is valid.

The rRTLb tuple has owner linkage to the message buffers (rIMALLOC or rIAPDU). When messages are being decoded the rRTLb tuple is owner of a rIMALLOC tuple. After the message is decoded the rRTLb is owner of a rIAPDU tuple. The process (rIPCBLA tuple) is always linked to the rRTLb when messages are processed.

The channel (rIDPB tuple) could be involved in other message traffic and have links to other rRTLb tuples and rIRTEILA tuples.

3. AUDIT CHECKS

The RTLb audit invokes the DRINASEK driver to make the linkages and semantic checks for each rRTLb tuple. The RTLb audit uses the following error codes. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given. For each tuple, the RTLb audit makes the following checks:

INVOWNER = Check valid owner linkage.

MEMBAD = Check member linkage to rIDPB or rIDB_HEADC.

DEADPID = Check rIPCBLA linkage.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	If the rRTLb.dpb_flag is DBYES (1) a rIPCBLA tuple should be linked to this rRTLb. No check is made to verify the rIPCBLA tuple represents a live process.
Possible Error Effect:	Messages could be lost and the task process could be stuck.
Corrective Action Taken:	The rRTLb tuple is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rRTLb tuple.

<i>Bad Data</i> =	Linkage field to rIPCBLA (rIRTLB.rtlb_pcbam.st_link).
<i>Good Data</i> =	Application method (rIRTLB.appl_int).
<i>Logical Index</i> =	Key of rIRTLB (rIRTLB.rtlb)
<i>Dump</i> =	Contents of rIRTLB tuple.

Table 2 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The link_owner field has been found in an invalid state.
Possible Error Effect:	The message queue is lost.
Corrective Action Taken:	The rIRTLB tuple is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRTLB tuple.
<i>Bad Data</i> =	The link_owner field in rIRTLB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIRTLB tuple.
<i>Dump</i> =	Contents of rIRTLB tuple.

Table 3 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	This rIRTLB tuple is not linked to the owner indicated by the flag (rIRTLB.dpb_flag).
Possible Error Effect:	The message for this channel could be dropped.
Corrective Action Taken:	This EOC/TMC channel will be recovered. Any process linked to this rIRTLB tuple is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIRTLB tuple.
<i>Bad Data</i> =	Key of owner tuple (rIDPB or rIDB_HEADC).
<i>Good Data</i> =	Flag for owner field (1 for rIDPB or 0 for rIDB_HEADC).
<i>Logical Index</i> =	Key of rIRTLB (rIRTLB.rteila)
<i>Dump 1</i> =	Contents of rIRTLB tuple.
<i>Dump 2</i> =	Contents of either rIDPB or rIDB_HEADC.

RTMHG

Software Release: 5E10 and later

1. MULTI-LINE HUNT GROUP INFORMATION (RTMHG) AUDIT

The routing process of the routing and terminal allocation (RTA) subsystem in the switching module (SM), optical remote switching module (ORM) or remote switching module (RSM) finds available lines for terminating calls in the SMs by hunting. Hunting is the process of selecting one of a set of available ports that satisfy a calling customer. It derives information about the status of lines from the multi-line hunt group (MLHG) information relation (RLRT_MHG). This relation is static (write-protected); it contains information about all available hunted lines. Each tuple of RLRT_MHG with the appropriate hunt type corresponds to either the line status relation (RLLNSTAT) or the large line status relation (RLLGLNSTAT) to expedite the hunting process.

The multi-line hunt group information audit (RTMHG) works with RLMEM, RLLGMEM, RLLNSTAT, RLLGLNSTAT and two automatic voice network (AUTOVON) relations:

- the precedence and preemption line status relation (RLPPLNST) and
- the precedence and preemption large line status relation (RLPPLLNST),

to ensure that missing tuples are added to these relations. The line status audit (LNSTAT) and the large line status audit (LGLNSTAT) find invalid keys and extra tuples in RLLNSTAT and RLLGLNSTAT; the LMHG member audit (MEM) and the large MLHG member status audit (LGMEM) find extra tuple, invalid keys and incorrect member status; the precedence and preemption line status (PPLNST) and the precedence and preemption large line status (PPLLNST) audits find extra tuples in RLPPLNST and RLPPLLNST.

This audit is part of the RTA audits subgroup. In an SM/RSM/ORM, the audit operates in the operating system for distributed switching-switching module (OSDS-M) environment in the segmented mode.

2. DATA STRUCTURES

The data structures checked by the RTMHG audit are the multi-line hunt group information relation (RLRT_MHG), the line status relation (RLLNSTAT), the large line status relation (RLLGLNSTAT), the precedence and preemption line status relation (RLPPLNST), and the precedence and preemption large line status relation (RLPPLLNST). AUTOVON groups in RLPPLNST and RLPPLLNST store extended information about lines with regular hunt types: RTH_PPSMHG in RLLNSTAT and RTH_PPLMHG in RLLGLNSTAT. These hunt types are stored in RLRT_MHG only. Figure 1 shows these data structures.

The existing RTMHG audit uses the static driver (DRSTATIC) to step through the static relation RLRTMHG, and performs the semantic check on each tuple by calling the semantic function AUstrtmhg(). The semantic function AUstrtmhg() performs a static to dynamic existence check, invalid hunt type check, and invalid key check on each tuple.

The semantic function AUstrtmhg() checks the hunt type and module number of the rRTMHG tuple in order to retrieve an appropriate dynamic tuple. The AUstrtmhg() retrieves the tuple from dynamic relation RLISATUCD if hunt type is equal to RTH_ISAT or RTH_LGISAT. If a dynamic tuple is missing, a tuple is inserted in RLISATUCD. A check is also included to verify the key value for each dynamic tuple.

The RTH_LGISAT is no longer a valid hunt type. It is now consolidated into the small hunt type RTH_ISAT.

The RLRT_MHG relation is made up of one tuple for each multi-line hunt group. Included in the information, within the tuples, is the key (group number), the hunt type, the module number, and the group type (global or local). The fields checked by the audit are the group number (grpnum), the hunt type (htype), and the module number (module).

SM & RSM*

AM

RTMHG AUDIT

↓ rIRT_MHG

1	T	AM	
2	LG	AM	
3	PT	AM	
4	PG	AM	
5	PT	RSM2	
86	T	RSM1	
87	LG	AM	

T = Trunk
 LG = Large Trunk
 PT = AUTOVON Trunk
 PG = AUTOVON Large Trunk

↓ rLNSTAT

1	
3	

↓ rLGLNSTAT

2	
4	
87	

↓ rPPLNST

3	

↓ rPPLLNST

4	

*In an RSM environment, the rIRT_MHG relation is fully redundant in both the AM and RSM.

Figure 1 (RTMHG-1) RTMHG Audit

3. AUDIT CHECKS

The RTMHG audit uses the static driver (DRSTATIC) to check the correspondence among the RLRT_MHG, line status, and large line status relations. The DRSTATIC driver invokes a semantic check function that performs these checks.

The audit checks the module number to ensure that the multi-line hunt group should exist in the processor. If it is correct, the audit proceeds as follows. If it is not correct, an error is reported and the audit continues with the next tuple.

- (a) The audit checks the hunt type and retrieves a tuple.

If the hunt type is RTH_CIRCULAR, RTH_PPSMHG, RTH_REGULAR, or RTH_UCD, the audit retrieves a rLNSTAT tuple.

If the hunt type is RTH_LGCIR, RTH_LGREG, RTH_LGUCD, or RTH_PPLMHG, the audit retrieves a rLGLNSTAT tuple.

If the hunt type is RTH_MUPH, the audit retrieves a rRTMUPHST tuple.

- (b) The audit checks the key of each tuple it retrieves and corrects the key if it is in error.
- (c) If no tuple is retrieved, the audit determines whether the tuple is vacant or lost. If the tuple is lost, the tuple inserted.
- (d) The semantic check function checks the hunt type after the checks for rLNSTAT and rLGLNSTAT pass.
- If the hunt type is RTH_PPSMHG, the audit retrieves rPPLNST and checks for lost tuples and key errors.
- If the hunt type is RTH_PPLMHG, the audit retrieves rPPLGLNST and checks for lost tuples and key errors.
- (e) This audit also checks for hunt type RTH_ISAT and RTH_LGISAT [for the integrated services digital network (ISDN) attendant hunt group].
- If the hunt type is RTH_ISAT or RTH_LGISAT, the RTMHG audit retrieves a rISATUCD tuple.
- (f) The key of each tuple retrieved from the rISATUCD relation is checked for correctness, and corrects it if in error. If no tuple is retrieved, the audit inserts a missing tuple in the rISATUCD relation.
- (g) This audit does not check for hunt type RTH_LGISAT anymore.

4. ERROR HANDLING

If rIRT_MHG tuples with the specified hunt and module types are not found in the line status relations, they are created. If a rIRT_MHG tuple has an improper hunt type, an error message requesting manual action is printed on the read-only printer (ROP). If a tuple of a line status relation has invalid data in its key, the key is corrected.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The multi-line hunt group information relation contains a tuple which has an invalid hunt type. This error will repeat until it is fixed.
Possible Error Effect:	Inability to hunt to lines in this group and possible assert failures.
Corrective Action Taken:	None.

Manual Action Required:	Determine the nature of the data base inconsistency and correct it. This message will repeat until corrective action is taken.
Dump Description:	
<i>Error Address</i> =	Address of the tuple buffer.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	An rLNSTAT tuple has bad data in the key (grpnum) field for a multi-line hunt group.
Possible Error Effect:	Inability to terminate to lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rLNSTAT tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rLNSTAT tuple.
<i>Good Data</i> =	Group number of the MLH group.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	rLNSTAT tuple.

Table 3 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	An rLGLNSTAT tuple has bad data in the key (grpnum) field for a multi-line hunt group.
Possible Error Effect:	Inability to terminate to lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rLGLNSTAT tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rLGLNSTAT tuple.
<i>Good Data</i> =	Group number of the MLH group.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

Table 4 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	There is a tuple missing in the RLLGLNSTAT relation for a large multi-line hunt group.
Possible Error Effect:	Inability to terminate to lines in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the RLLNSTAT relation for a multi-line hunt group.
Possible Error Effect:	Inability to terminate to lines in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.

<i>Dump</i> =	NA.

Table 6 Error Code: MUPHKEY

Error Code:	MUPHKEY
Error Description:	An rRTMUPHST tuple has bad data in the key (grpnum) field for a multi-line hunt group.
Possible Error Effect:	Inability to terminate to lines in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rRTMUPHST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rRTMUPHST tuple.
<i>Good Data</i> =	Group number of the MLH group.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	rRTMUPHST tuple.

Table 7 Error Code: MUPHLOST

Error Code:	MUPHLOST
Error Description:	There is a tuple missing in the RLRTMUPHST relation for a large multi-line hunt group.
Possible Error Effect:	Inability to terminate to lines in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

Table 8 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	An rIPPLNST tuple has bad data in the key (grpnum) field for a multi-line hunt group.
Possible Error Effect:	Unable to preempt the line in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rIPPLNST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rIPPLNST tuple.
<i>Good Data</i> =	Group number of the MLH group.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	rIPPLNST tuple.

Table 9 Error Code: PPLGLOST

Error Code:	PPLGLOST
Error Description:	There is a tuple missing in the RLPPLNST relation for a large multi-line hunt group.
Possible Error Effect:	Unable to preempt the lines in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

Table 10 Error Code: PPLKEY

Error Code:	PPLKEY
Error Description:	An rIPPLNST tuple has bad data in the key (grpnum) field for a multi-line hunt

	group.
Possible Error Effect:	Unable to preempt the line in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rIPPLNST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rIPPLNST tuple.
<i>Good Data</i> =	Group number of the MLH group.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

Table 11 Error Code: PPLOSTTUP

Error Code:	PPLOSTTUP
Error Description:	There is a tuple missing in the RLPPLNST relation for a multi-line hunt group.
Possible Error Effect:	Unable to preempt the lines in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the MLH group.
<i>Dump</i> =	NA.

RTMSGBUF

Software Release: 5E10 and later

1. REMOTE TERMINAL MESSAGE BUFFER (RTMSGBUF) AUDIT

For a TR303 remote terminal (RT) on integrated digital carrier unit (IDCU) the embedded operations channel (EOC) is used by the level 7 administration to process tasks using messages. The rRTMSGBUF tuple is linked to the rIAPDU tuple for the EOC channels when a message is ready to transmit or has been received from a TR303 RT.

The RTMSGBUF audit maintains the integrity of these linkages.

2. DATA STRUCTURES

The rRTMSGBUF tuple is linked to the idle list (rIDB_HEADC) or the application protocol data unit (rIAPDU) for IDCU EOC messages. The link_owner field is used to determine the correct linkage.

The dynamic data structures for EOC/TMC message processing on TR303 RT are shown in Figure 1 .

3. AUDIT CHECKS

The RTMSGBUF audit invokes the DRINASEK driver to make the linkages and semantic checks for each rRTMSGBUF tuple. This listing gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given. For each tuple, the RTMSGBUF audit makes the following checks:

MEMBAD = Check member linkage to rIAPDU or rIDB_HEADC.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	This rRTMSGBUF tuple is not linked to the owner indicated by the flag (rRTMSGBUF.link_owner).
Possible Error Effect:	The message for this channel could be dropped.
Corrective Action Taken:	All messages on this queue will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rRTMSGBUF tuple.
<i>Bad Data</i> =	Key of owner tuple (rIAPDU or rIDB_HEADC).
<i>Good Data</i> =	Flag for owner field (1 for rIAPDU or 0 for rIDB_HEADC).
<i>Logical Index</i> =	Key of rRTMSGBUF (rRTMSGBUF.rtmsgbuf_id)
<i>Dump</i> =	Contents of rRTMSGBUF tuple.

RTPSTAT

Software Release: 5E10 and later

1. REMOTE MEASUREMENT SYSTEM TEST POSITION STATUS (RTPSTAT) AUDIT

This audit audits dynamic relation RLR_TPSTAT.

2. DATA STRUCTURES

For details on the fields contained in the RLR_TPSTAT relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

Relation RLR_TPSTAT is a member of relation RLR_CATPS. Linkage checking is done by the linkage driver function `drlink` which will restore the key attribute and null out the tuple if linkage problems are found. The semantic checks, performed by function `sr_tpstat` are as follows:

- (1) Check the idle status of both ports against the status found in a corresponding tuple in relation RLR_SPTPN.
- (2) Make sure that for each tuple in the RLR_TPSTAT relation there is a tuple in the static relation RLR_QKEYPT.
- (3) Ensure that for each tuple in RLR_TPSTAT there is a corresponding tuple in RLR_SPTPN.
- (4) Remove the tuple in RLR_TPSTAT if the corresponding tuple in RLR_SPTPN is not of type RTPOS.
- (5) Alter the idle status of each TAT in RLR_TPSTAT to correspond with the corresponding tuple in RLR_SPTPN.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADTESTPOS

Error Code:	BADTESTPOS
Error Description:	For a dynamic tuple in RLR_TPSTAT the corresponding tuple was found in RLR_SPTPN but the test position attribute did not match or was not of type RTPOS.
Possible Error Effect:	The tuple has wrong information.
Corrective Action Taken:	If the dynamic tuple is linked to relation RLR_CATPS, it is unlinked and the number of attended test positions in relation <code>rlR_MCATPS</code> and relation <code>RLR_CATPS</code> are decremented. The <code>rlR_TPSTAT</code> tuple is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the <code>rlR_TPSTAT</code> tuple.
<i>Bad Data</i> =	Value of test position attribute.
<i>Good Data</i> =	Expected value of test position attribute.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	<code>rlR_TPSTAT</code> tuple.

Table 2 Error Code: CRSPTPN

Error Code:	CRSPTPN
Error Description:	A tuple in relation RLR_SPTPN corresponding to the module and port in relation RLR_TPSTAT could not be found.
Possible Error Effect:	Tuple has incorrect information.
Corrective Action Taken:	A new rIR_SPTPN tuple is created and initialized with information in relation RLR_TPSTAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_TPSTAT tuple.
<i>Bad Data</i> =	Key of the rIR_TPSTAT tuple.
<i>Good Data</i> =	Key of the new rIR_SPTPN tuple.
<i>Logical Index</i> =	Logical key of the rIR_TPSTAT tuple.
<i>Dump</i> =	rIR_TPSTAT tuple.

Table 3 Error Code: RMTPSTAT

Error Code:	RMTPSTAT
Error Description:	For a dynamic tuple in RLR_TPSTAT there could not be found a corresponding module and port in the static relation RLR_QKEYPT.
Possible Error Effect:	Tuple has incorrect information.
Corrective Action Taken:	If the dynamic tuple is linked to relation RLR_CATPS, it is unlinked and the number of attended test positions in relation RLR_MCATPS and relation RLR_CATPS are decremented. The rIR_TPSTAT tuple is nulled out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_TPSTAT tuple.
<i>Bad Data</i> =	Key of extra rIR_TPSTAT tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rIR_TPSTAT tuple.

Table 4 Error Code: TATO

Error Code:	TATO
Error Description:	The "idle status" attribute of tat0 port in the rIR_TPSTAT tuple is not the same as that of its corresponding tuple in rIR_SPTPN.
Possible Error Effect:	Tuple has incorrect information.
Corrective Action Taken:	The tuple is updated to reflect the status in rIR_SPTPN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_TPSTAT tuple.
<i>Bad Data</i> =	Actual value of the "idle status" attribute.
<i>Good Data</i> =	Expected value of the "idle status" attribute.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rIR_TPSTAT tuple.

Table 5 Error Code: TAT1

Error Code:	TAT1
Error Description:	The "idle status" attribute of tat1 port in the rIR_TPSTAT tuple is not the same as that of its corresponding tuple in rIR_SPTPN.
Possible Error Effect:	Tuple has incorrect information.
Corrective Action Taken:	The tuple is updated to reflect the status in rIR_SPTPN.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIR_TPSTAT tuple.
<i>Bad Data</i> =	Actual value of the "idle status" attribute.
<i>Good Data</i> =	Expected value of the "idle status" attribute.
<i>Logical Index</i> =	Logical key of the tuple.
<i>Dump</i> =	rIR_TPSTAT tuple.

RTTRKG

Software Release: 5E10 and later

1. TRUNK GROUP INFORMATION (RTTRKG) AUDIT

The routing process of the routing and terminal allocation (RTA) subsystem in both the Communications module processor (CMP) and switching module (SM) finds available trunks for terminating calls in the SMs by hunting. Hunting is the process of selecting one of a set of available ports that satisfy a calling customer. It derives information about the status of trunks from the trunk group information relation (RLRT_TRKG). This relation is static (write-protected) and is fully redundant in both the CMP and SM; it contains information about all available hunted trunks. Each tuple of RLRT_TRKG with the appropriate hunt type corresponds to either the trunk status relation (RLTKSTAT) or the large trunk status relation (RLLGTKSTAT) to expedite the hunting process.

The trunk group information audit (RTTRKG) works with both RLTKSTAT and RLLGTKSTAT to ensure that missing tuples are added to the relations. The trunk status audit (TKSTAT) and the large trunk status audit (LGTKSTAT) find invalid keys and extra tuples in RLTKSTAT and RLLGTKSTAT.

The trunk group information audit (RTTRKG) works with RLTKSTAT, RLLGTKSTAT, and two automatic voice network (AUTOVON) relations--the precedence and preemption trunk status relation (RLPPTKST) and the precedence and preemption large trunk status relation (RLPPLTKST)--to ensure that missing tuples are added to the relations. The trunk status audit (TKSTAT) and the large trunk status audit (LGTKSTAT) find invalid keys and extra tuples in RLTKSTAT and RLLGTKSTAT; the precedence and preemption trunk status (PPTKST) and the precedence and preemption large trunk status (PPLTKST) audits find lost tuples in RLPPTKST and RLPPLTKST.

This audit is part of the routing and terminal allocation audits subgroup.

2. DATA STRUCTURES

The data structures checked by the RTTRKG audit are the RLRT_TRKG, RLTKSTAT, RLLGTKSTAT, RLPPTKST, and RLPPLTKST relations. Figure 1 shows these data structures.

The trunk group information relation is made up of one tuple for each group of hunted trunks. Included in the information within the tuples is the key (group number), group type, hunt type, and the module number, which indicates whether the group is global or local. The fields checked by the audit are the group number (grpnum), the group type (type), module number (module), and the hunt type (htype).

SM & RSM*

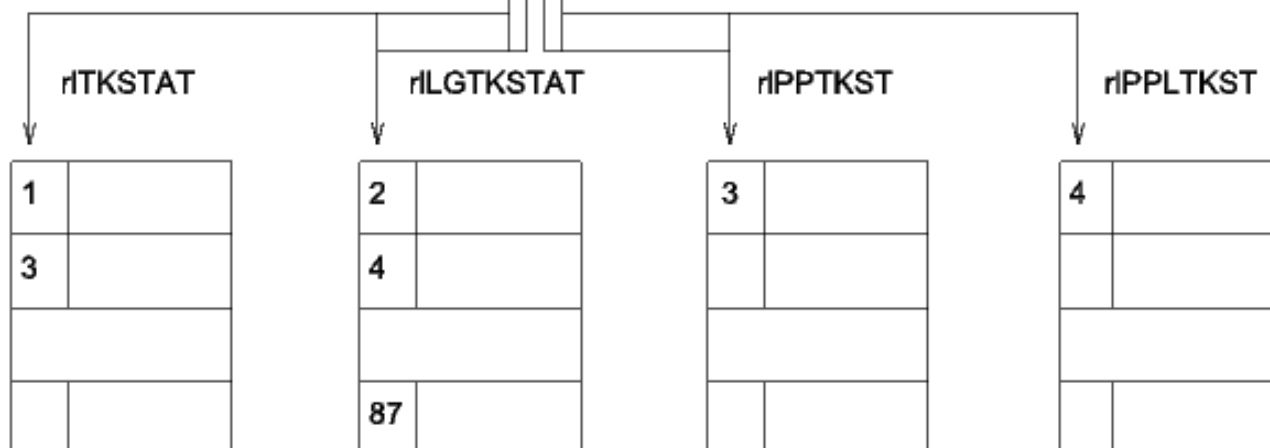
AM

RTTRKG AUDIT

rIRT_TRKG

1	T	AM	
2	LG	AM	
3	PT	AM	
4	PG	AM	
5	PT	RSM2	
86	T	RSM1	
87	LG	AM	

T = Trunk
 LG = Large Trunk
 PT = AUTOVON Trunk
 PG = AUTOVON Large Trunk



***IN AN RSM ENVIRONMENT, THE rRTTRKG RELATION IS FULLY REDUNDANT IN BOTH THE AM AND RSM.**

Figure 1 (RTTRKG-1) RTTRKG Audit

3. AUDIT CHECKS

The RTTRKG audit uses the static driver (DRSTATIC) to check the correspondence among the RLRT_TRKG, trunk status, and large trunk status relations. The DRSTATIC driver invokes a semantic check function that performs the

following check.

The audit checks the module number to ensure that the group type should exist in the processor. If it is correct, the audit proceeds as follows. If it is not correct, an error is reported and the audit continues with the next tuple.

- (1) The audit checks the hunt type and retrieves a tuple.

If the hunt type is RTH_BWDLIN, RTH_FWDLIN, RTH_UCD, RTH_GUCD, RTH_BGUCD, RTH_PRBLIN, or RTH_PRFLIN, the audit retrieves an rITKSTAT tuple.

If the hunt type is RTH_LGBWD, RTH_LGFWD, RTH_LGUCD, RTH_LGGUCD, RTH_LGBGUCD, RTH_PLBLIN, or RTH_PLFLIN, the audit retrieves an rLGTKSTAT tuple.

If the hunt type is first-in, first-out (RTH_FIFO), the audit checks RLPPTKST and RLPPLTKST. First-in, first-out hunt types are not stored in RLTKSTAT and RLLGTKSTAT because other relations (RLTKOWNER and RLTKQUE) store FIFO hunt types. The AUTOVON relations RLPPTKST and RLPPLGTKST relations do not have other relations to store FIFO hunt types.

- (2) The audit checks the key of each tuple it retrieves and corrects it if it is in error.
- (3) If no tuple is retrieved, the audit determines whether the tuple is vacant or lost. If it is lost, it is inserted.
- (4) The semantic check function checks the hunt type of rITKSTAT and rLGTKSTAT again.

If the hunt type is RTH_PRBLIN, the audit retrieves rIPPTKST and checks for lost tuples and key errors.

If the hunt type is RTH_PLBLIN, the audit retrieves rIPPLGTKST and checks for lost tuples and key errors.

4. ERROR HANDLING

If rIRT_TRKG tuples with the specified group, hunt, and module types are not found in the trunk status relations, they are created. If an rIRT_TRKG tuple has an improper hunt type, improper group type for trunks, or group type other than two-way, an error message requesting manual action is printed. If a tuple of a trunk status relation has invalid data in its key (group number), the key is corrected.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The trunk group information relation contains a tuple with an invalid hunt type. This error will repeat until it is fixed by maintenance personnel.
Possible Error Effect:	Inability to hunt to ports in this group and possible assert failures.
Corrective Action Taken:	None.
Manual Action Required:	Determine the nature of the database inconsistency and correct it. This message will repeat until corrective action is taken.
Dump Description:	
<i>Error Address</i> =	Address of the tuple buffer.
<i>Bad Data</i> =	Hunt type of the tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the trunk group.
<i>Dump</i> =	NA.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	An rTKSTAT tuple has bad data in the key (grpnum) field for a regular linear hunted port group.
Possible Error Effect:	Inability to terminate to ports in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rTKSTAT tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rTKSTAT tuple.
<i>Good Data</i> =	Group number of the port group.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	rTKSTAT tuple.

Table 3 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	A rLGTKSTAT tuple has bad data in the key (grpnum) field for a regular linear hunted port group.
Possible Error Effect:	Inability to terminate to ports in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rLGTKSTAT tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rLGTKSTAT tuple.
<i>Good Data</i> =	Group number of the port group.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	NA.

Table 4 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	There is a tuple missing in the rLGTKSTAT relation for a large linear hunted port group.
Possible Error Effect:	Inability to terminate to ports in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	NA.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the rTKSTAT relation for a regular linear hunted port group.
Possible Error Effect:	Inability to terminate to ports in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	NA.

Table 6 Error Code: PPKEY

Error Code:	PPKEY
Error Description:	An rIPPTKST tuple has bad data in the key (grpnum) field for a regular hunted port group.
Possible Error Effect:	Inability to preempt the ports in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rIPPTKST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rIPPTKST tuple.
<i>Good Data</i> =	Group number of the port group.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	rIPPTKST tuple.

Table 7 Error Code: PPLGLOST

Error Code:	PPLGLOST
Error Description:	There is a tuple missing in the RLPPLTKST relation for a large hunted port group.
Possible Error Effect:	Inability to preempt the ports in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	NA.

Table 8 Error Code: PPLKEY

Error Code:	PPLKEY
Error Description:	A rIPPLTKST tuple has bad data in the key (grpnum) field for a large hunted port group.
Possible Error Effect:	Inability to preempt the ports in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the grpnum field in the rIPPLTKST tuple.
<i>Bad Data</i> =	Contents of the grpnum field in the rIPPLTKST tuple.
<i>Good Data</i> =	Group number of the port group.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	NA.

Table 9 Error Code: PPLOSTTUP

Error Code:	PPLOSTTUP
Error Description:	There is a tuple missing in the RLPPTKST relation for a regular hunted port group.
Possible Error Effect:	Inability to preempt the ports in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the port group.
<i>Dump</i> =	NA.

Table 10 Error Code: TYPE

Error Code:	TYPE
Error Description:	The RLRT_TRKG relation contains a tuple with an invalid group type. This group type cannot be associated with a linear hunt type. This error will repeat until it is fixed by maintenance personnel.
Possible Error Effect:	Inability to terminate to ports in this group and possible assert failures.
Corrective Action Taken:	None.
Manual Action Required:	Determine the nature of the database inconsistency and correct it. This message will repeat until corrective action is taken.
Dump Description:	
<i>Error Address</i> =	Address of the tuple buffer.
<i>Bad Data</i> =	Group type of the tuple.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	Group number of the trunk group.
<i>Dump</i> =	NA.

S7MDB-A**Software Release:** 5E11 - 5E12**1. CCS7 MESSAGE DATA BLOCK (S7MDB) AUDIT**

S7MDB audit steps through the RLS7MDB (CCS7 message data block) relation. For each tuple, the audit makes sure that the linkage between the RLS7MDB and RLS7RDB (CCS7 routing data block) relations, and the RLLS_STAT (link set status) and RLDB_HEADC (database headcell) relations are accurate.

2. DATA STRUCTURES

For details on the fields contained in the RLS7MDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

S7MDB-B

Software Release: 5E13 and later

1. CCS7 MESSAGE DATA BLOCK (S7MDB) AUDIT

S7MDB audit steps through the RLS7MDB (CCS7 message data block) relation. For each tuple, the audit makes sure that the linkage between the RLS7MDB and RLS7RDB (CCS7 routing data block) relations, and the RLLS_STAT (link set status) and RLDB_HEADC (database headcell) relations are accurate.

2. DATA STRUCTURES

For details on the fields contained in the RLS7MDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	The error is due to one of the following: (1) The rLS7MDB is owned by its idle list (rLS7MDB.mhcm.st_ownk = DBS7IDLE + DBSFDB_HEADC) but is not owned by the null link list in rLS_STAT (rLS7MDB.rd_ls_s7.st_ownk != (DBLSNULL + DBSFLS_STAT) or rLS7MDB.linktype does not equal DBMDBLLINK. (2) The rLS7MDB is owned by its busy list (rLS7MDB.mhcm.st_ownk = DBS7BUSY + DBSFDB_HEADC) or its input list (rLS7MDB.mhcm.st_ownk = DBS7INPUT + DBSFDB_HEADC) but either it is not owned by rLS7RDB or rLS7MDB.linktype does not equal DBMDBRLINK. (3) The rLS7MDB is owned by its busy list (rLS7MDB.mhcm.st_ownk = DBS7BUSY + DBSFDB_HEADC) or its input list (rLS7MDB.mhcm.st_ownk = DBS7INPUT + DBSFDB_HEADC) but either it is not owned by rLS_STAT or rLS7MDB.linktype does not equal DBMDBLLINK. (4) The rLS7MDB is owned by its high list (rLS7MDB.mhcm.st_ownk = DBS7HIGH + DBSFDB_HEADC) but either it is not owned by rLS_STAT or rLS7MDB.linktype does not equal DBMDBLLINK.
Possible Error Effect:	A message data block may not be available for use.
Corrective Action Taken:	The audit sets all fields to 0 except the key, sets rLS7MDB.linktype to DBMDBLLINK, links the tuple to the null list of relation RLLS_STAT and links

	the tuple to its idle list in relation RLDB_HEADC (rIS7MDB.mhcm.st_ownk = DBS7IDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS7MDB tuple.
<i>Bad Data</i> =	rIS7MDB.linktype.
<i>Good Data</i> =	(1), (2), (3) or (4) - indicates error description.
<i>Logical Index</i> =	rIS7MDB.mdb_id.
<i>Dump</i> =	Contents of rIS7MDB tuple.

Table 2 Error Code: INV

Error Code:	INV
Error Description:	The error is due to one of the following: (1) The rIS7MDB.linktype indicates that the rIS7MDB tuple is not owned by any recognized relation but should be owned by either rILS_STAT or rIS7RDB. (2) The rIS7MDB.mhcm.st_ownk indicates that the rIS7MDB tuple is not owned by a recognized rIDB_HEADC list.
Possible Error Effect:	A message data block may not be available for use.
Corrective Action Taken:	The audit sets all fields to 0 except the key, sets rIS7MDB.linktype to DBMBLLINK, links the tuple to the null list of relation RLLS_STAT and links the tuple to its idle list in relation RLDB_HEADC (rIS7MDB.mhcm.st_ownk = DBS7IDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	(1) or (2) - indicates error description.
<i>Bad Data</i> =	rIS7MDB.linktype.
<i>Good Data</i> =	rIS7MDB.mhcm.st_ownk.
<i>Logical Index</i> =	rIS7MDB.mdb_id.
<i>Dump</i> =	Contents of rIS7MDB tuple.

Table 3 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The rIS7MDB is owned by its idle list (rIS7MDB.mhcm.st_ownk = DBS7IDLE + DBSFDB_HEADC) but is not owned by the null link list in rILS_STAT (rIS7MDB.rd_ls_s7.st_ownk != (DBLSNULL + DBSFLS_STAT)).
Possible Error Effect:	A message data block may not be available for use.
Corrective Action Taken:	The audit sets all fields to 0 except the key, sets rIS7MDB.linktype to DBMBLLINK, links the tuple to the null list of relation RLLS_STAT and links the tuple to its idle list in relation RLDB_HEADC (rIS7MDB.mhcm.st_ownk = DBS7IDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIS7MDB tuple.
<i>Bad Data</i> =	rIS7MDB.rd_ls_s7.st_ownk.
<i>Good Data</i> =	rIS7MDB.linktype.
<i>Logical Index</i> =	rIS7MDB.mdb_id.
<i>Dump</i> =	Contents of rIS7MDB tuple.

S7RDB

Software Release: 5E11 and later

1. CCS7 ROUTING DATA BLOCK (S7RDB) AUDIT

S7RDB audit steps through the RLS7RDB (CCS7 routing data block) relation. For each tuple in the relation, it checks the corresponding static relation RLDPC_INFO (destination point code information) and delete the extra rIS7RDB tuple. The audit also checks the linkage between the RLS7RDB and RLRSTMR (rout set timing list) relations, and the RLXPTMR (transfer prohibit timing list) and RLS7MDB (CCS7 message data block) relations. Data in the dynamic rIS7RDB tuple and its alternate structure alCCRDB will be checked against data in the dynamic relation RLDPC_STAT (destination point code status) and static relations RLLS_GRP (link set group), RLAPCI_APC (adjacent point code information) and RLDPC_DPCI (destination point code index).

2. DATA STRUCTURES

For details on the fields contained in the RLS7RDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: APC_DPCI

Error Code:	APC_DPCI
Error Description:	The error is due to one of the following: 1 = The apc_dpci in dynamic rIS7RDB does not equal dpcindx in static rIDPC_DPCI for a valid link set. 2 = An invalid link set is linked to idle timer lists, but apc_dpci is not equal to 0xffff.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Corresponds to the Error description as follows: 1 = Audit correct apc_dpci. 2 = Audit link tuple to all timer idle lists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	apc_dpci from rIS7RDB.
<i>Good Data</i> =	Corresponds to the Error description as follows:

	1 = dpcindx from rIDPC_DPCI.
	2 = 1.
<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 2 Error Code: APCI_CON_ERR

Error Code:	APCI_CON_ERR
Error Description:	The apci from static rILS_GRP cannot be found in static rIAPCI_APC.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the static relations.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	apci from rILS_GRP.
<i>Good Data</i> =	apci from rILS_GRP.
<i>Logical Index</i> =	Key of rILS_GRP tuple.
<i>Dump</i> =	NA.

Table 3 Error Code: ARRAY_ENTRY_BAD

Error Code:	ARRAY_ENTRY_BAD
Error Description:	Either the CCcrdata[].cluster does not equal the cluster value returned from a common channel signaling (CCS) function or CCcrdata[].apci does not equal the adjacent point code index (APCI) returned from the same function for the representative routing data block (RDB) for the cluster-route.
Possible Error Effect:	Some destination point codes (DPCs) of this cluster-route may have an incorrect accessibility status and signaling traffic may be routed to one of these DPCs over incorrect link sets.
Corrective Action Taken:	The audit sets CCcrdata[].cluster and CCcrdata[].apci equal to the cluster and APCI values, respectively, returned from the function.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster value returned from the function.
<i>Bad Data</i> =	Adjacent point code index (APCI) returned from the function.
<i>Good Data</i> =	Index into CCcrdata[] and CCcr_rdb[].
<i>Logical Index</i> =	rIS7RDB.rdb_id from the current tuple.
<i>Dump</i> =	Contents of CCcrdata[index].

Table 4 Error Code: BAD_RTMR_INV_LS

Error Code:	BAD_RTMR_INV_LS
Error Description:	The link set for an rIS7RDB tuple as determined by the offset is invalid. The tuple is invalidly linked to dynamic rIRSTMR.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit links rIS7RDB tuple to idle list of rIRSTMR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	Key of invalid owner in rIRSTMR.
<i>Good Data</i> =	-1.
<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 5 Error Code: BAD_TTMR_INV_LS

Error Code:	BAD_TTMR_INV_LS
Error Description:	The link set for an rIS7RDB tuple as determined by the offset is invalid. The tuple is invalidly linked to dynamic rITCTMR.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit links rIS7RDB tuple to idle list of rITPTMR.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	rlS7RDB tuple.
<i>Bad Data</i> =	Key of invalid owner in rlTCTMR.
<i>Good Data</i> =	-1.
<i>Logical Index</i> =	Key of rlS7RDB tuple.
<i>Dump</i> =	NA.

Table 6 Error Code: BAD_XTMR_INV_LS

Error Code:	BAD_XTMR_INV_LS
Error Description:	The link set for an rlS7RDB tuple as determined by the offset is invalid. The tuple is invalidly linked to dynamic rlXPTMR.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit links rlS7RDB tuple to idle list of rlXPTMR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rlS7RDB tuple.
<i>Bad Data</i> =	Key of invalid owner in rlXPTMR.
<i>Good Data</i> =	-1.
<i>Logical Index</i> =	Key of rlS7RDB tuple.
<i>Dump</i> =	NA.

Table 7 Error Code: CON_ERR_APC_DPC

Error Code:	CON_ERR_APC_DPC
Error Description:	The apc as determined from static rlAPC_APCI is used to read static rlDPC_DPCI and finds no match.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit AUDPCSTAT runs.
Manual Action Required:	Investigate the static relations.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	apc as determined from rlAPC_APCI.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rlAPC_APCI tuple.
<i>Dump</i> =	NA.

Table 8 Error Code: DPCI_BAD

Error Code:	DPCI_BAD
Error Description:	The audit tries to find a matching tuple in relation RLDPC_STAT whose key (rlDPC_STAT.dpci) equals rlS7RDB.rdb_id/CCRDB_DPC but does not find the tuple. The rlS7RDB.rdb_id has been previously verified so relation RLDPC_STAT is in error.
Possible Error Effect:	A DPC may have an incorrect accessibility status and signaling traffic may be routed to the DPC over incorrect link sets.
Corrective Action Taken:	Another audit inserts the missing tuple in relation RLDPC_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rlS7RDB.rdb_id.
<i>Dump</i> =	NA.

Table 9 Error Code: DPCI_CON_ERR

Error Code:	DPCI_CON_ERR
Error Description:	A destination point code index (DPCI) is computed from rlS7RDB.rdb_id (rlS7RDB.rdb_id/CCRDB_DPC) and is used as the key to read a tuple in rlDPC_STAT. The rlDPC_STAT tuple does not exist.
Possible Error Effect:	CCS cannot route any message signaling units (MSUs) to that destination.
Corrective Action Taken:	The related audit DPCDPCI runs and inserts the missing rlDPC_STAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rlS7RDB tuple.

<i>Bad Data</i> =	The computed DPCI.
<i>Good Data</i> =	Offset (which one of CCRDB_DPC possible routing blocks) for this DPCI.
<i>Logical Index</i> =	Key (rIS7RDB.rdb_id) of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: DPC_CON_ERR

Error Code:	DPC_CON_ERR
Error Description:	The dpc field as read from the dynamic rIDPC_STAT tuple was not found in the static rIDPC_DPCI tuple.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit AUDPCSTAT runs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	dpc from rIDPC_STAT.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 11 Error Code: INV

Error Code:	INV
Error Description:	The audit calls a CCS function to retrieve the corresponding CCcr_rdb[] entry for a CCcrdata[] entry. The function returns a value which the audit does not recognize.
Possible Error Effect:	DPCs may have incorrect accessibility statuses and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	The audit requests the retrieval during its next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster for the current RDB.
<i>Bad Data</i> =	APCI for the current RDB.
<i>Good Data</i> =	Invalid return value from function call.
<i>Logical Index</i> =	rIS7RDB.rdb_id from the current tuple.
<i>Dump</i> =	NA.

Table 12 Error Code: LSNUM_BAD

Error Code:	LSNUM_BAD
Error Description:	Link set number in dynamic rIS7RDB does not equal link set number as read from static rIDPC_DPCI/rIDPC_INFO.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit corrects link set number in rIS7RDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	Link set number from rIS7RDB.
<i>Good Data</i> =	Link set number from static relation.
<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 13 Error Code: LS_CON_ERR

Error Code:	LS_CON_ERR
Error Description:	The offset computed from dynamic rIS7RDB is used to read the link set number in statics rIDPC_DPCI or rIDPC_INFO. This link set number is then used to read static rLS_GRP, and cannot be found.
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit AUDPCSTAT runs.
Manual Action Required:	Investigate the static relations.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	Link set number.
<i>Good Data</i> =	Offset of key of rIS7RDB.

<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 14 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The audit cannot find the corresponding entry in CCcr_rdb[] which has the representative routing data block (RDB) for the current cluster-route CCcrdata[] entry.
Possible Error Effect:	DPCs may have incorrect accessibility statuses and signaling traffic may be routed to DPCs over incorrect link sets.
Corrective Action Taken:	The audit sets CCcr_rdb[] equal to rIS7RDB.rdb_id, CCcrdata[].rdb_count equal to one, CCcrdata[].cluster and CCcrdata[].apci equal to the cluster value and the APCI, respectively, for the representative RDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster for the current RDB.
<i>Bad Data</i> =	APCI for the current RDB.
<i>Good Data</i> =	Representative RDB from CCcr_rdb[].
<i>Logical Index</i> =	rIS7RDB.rdb_id from the current tuple.
<i>Dump</i> =	NA.

Table 15 Error Code: RDB_BAD

Error Code:	RDB_BAD
Error Description:	The current RDB (alCCRDB.rdb_id) is not the representative RDB for a cluster-route in CCcrdata[] (alCCRDB.rdb_id != CCcr_rdb[]), but either alCCRDB.clust_rrs or alCCRDB.outsync, or both, set to DBYES instead of DBNO for the RDB.
Possible Error Effect:	This particular DPC may have an incorrect accessibility status, and signaling traffic may be routed to it over incorrect link sets.
Corrective Action Taken:	The audits sets both alCCRDB.clust_rrs and alCCRDB.outsync to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster for the current RDB.
<i>Bad Data</i> =	APCI for the current RDB.
<i>Good Data</i> =	Representative RDB from CCcr_rdb[].
<i>Logical Index</i> =	rIS7RDB.rdb_id from the current tuple.
<i>Dump</i> =	NA.

Table 16 Error Code: RDB_BAD_TIMR

Error Code:	RDB_BAD_TIMR
Error Description:	An rIS7RDB tuple with an invalid link set is linked to a timer tuple whose physical key is 0. Good data in the dump description indicates which timer, as follows: 1 = rIRSTMR 2 = rIXPTMR 3 = rITCTMR
Possible Error Effect:	Unknown.
Corrective Action Taken:	Audit links rIS7RDB tuple to idle list of all timers.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	Logical key of invalid owner tuple.
<i>Good Data</i> =	1, 2, or 3, as stated in error description.
<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 17 Error Code: READ

Error Code:	READ
Error Description:	The audit attempts to find a tuple in relation RLDPC_STAT whose key (rIDPC_STAT.dpci) equals aICCRDB.apc_dpci for an RDB but does not find a tuple.
Possible Error Effect:	A DPC may have an incorrect accessibility status and signaling traffic may be routed to the DPC over incorrect link sets.
Corrective Action Taken:	Another audit inserts the missing tuple into relation RLDPC_STAT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster for the current rIS7RDB tuple.
<i>Bad Data</i> =	aICCRDB.apc_dpci for the current rIS7RDB tuple.
<i>Good Data</i> =	APCI for the current rIS7RDB tuple.
<i>Logical Index</i> =	rIS7RDB.rdb_id for the current rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 18 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The error is due to one of the following: 1 = The link set in static rIDPC_DPCI is not found in static rCLS_LS. 2 = The alternate link set in static rIDPC_INFO is not found in static rCLS_LS.
Possible Error Effect:	Unknown.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the link set numbers in the relations listed in error description.
Dump Description:	
<i>Error Address</i> =	rIS7RDB tuple.
<i>Bad Data</i> =	Corresponds to the Error Description as follows: 1 = Value of lnk_set in rIDPC_DPCI. 2 = Value of alternate link set in rIDPC_INFO.
<i>Good Data</i> =	Corresponds to the Error Description as follows: 1 = Indicates error listed in error description and bad data. 2 = Indicates error listed in error description and bad data.
<i>Logical Index</i> =	Key of rIS7RDB tuple.
<i>Dump</i> =	NA.

Table 19 Error Code: TABLE_FULL

Error Code:	TABLE_FULL
Error Description:	There is an entry missing for a cluster-route in CCcrdata[] The audit attempts to insert the missing entry but there is no more room in the table.
Possible Error Effect:	DPCs for this cluster-route may have incorrect accessibility statuses and signaling traffic may be routed these DPCs over incorrect link sets.
Corrective Action Taken:	The audit requests a rebuild of CCcrdata[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Cluster for the current RDB.
<i>Bad Data</i> =	APCI for the current RDB.
<i>Good Data</i> =	Representative RDB from CCcr_rdb[].
<i>Logical Index</i> =	rIS7RDB.rdb_id from the current tuple.
<i>Dump</i> =	NA.

SCB

Software Release: 5E10 and later

1. STACK CONTROL BLOCK (SCB) AUDIT

A stack implements function calls and returns, passes parameters, and allocates storage for local variables. Processes are associated with stacks using a one-to-one linkage between the process control block (PCB) and the stack control block (SCB). Each SCB is mapped to an area of memory that represents the stack. In the CMP, there are two sizes of stacks: normal and large. In the SM, there are two types of stacks: those with an associated PMDB and those without. operating system for distributed switching (OSDS) provides a fixed number of each per processor.

The stack control block audit runs in the operational kernel process (OKP) and switch maintenance kernel process (SMKP) environments in the administrative module, the switching module (SM) and the communications module processor (CMP).

2. DATA STRUCTURES

The fields checked by this audit are the llink, the rlink and the sowner.

In addition to the SCBs themselves (OSscb[]), the audit verifies and uses the global data:

- The array of process control blocks (OSbpcb[]) any of which may own a single SCB.
- The global indexes OSffscb and OSlfscb which indicate the first and last members of the doubly linked SCB idle list.
- The indexes of the SCBs to determine if they have process message data blocks (PMDBs) associated with them.

It should also be noted that normal stacks are added and removed from the top of the idle list (OSffscb), while large stacks are added and removed from the bottom of the idle list (OSlfscb).

Figure 1 illustrates how a process is associated with a stack.

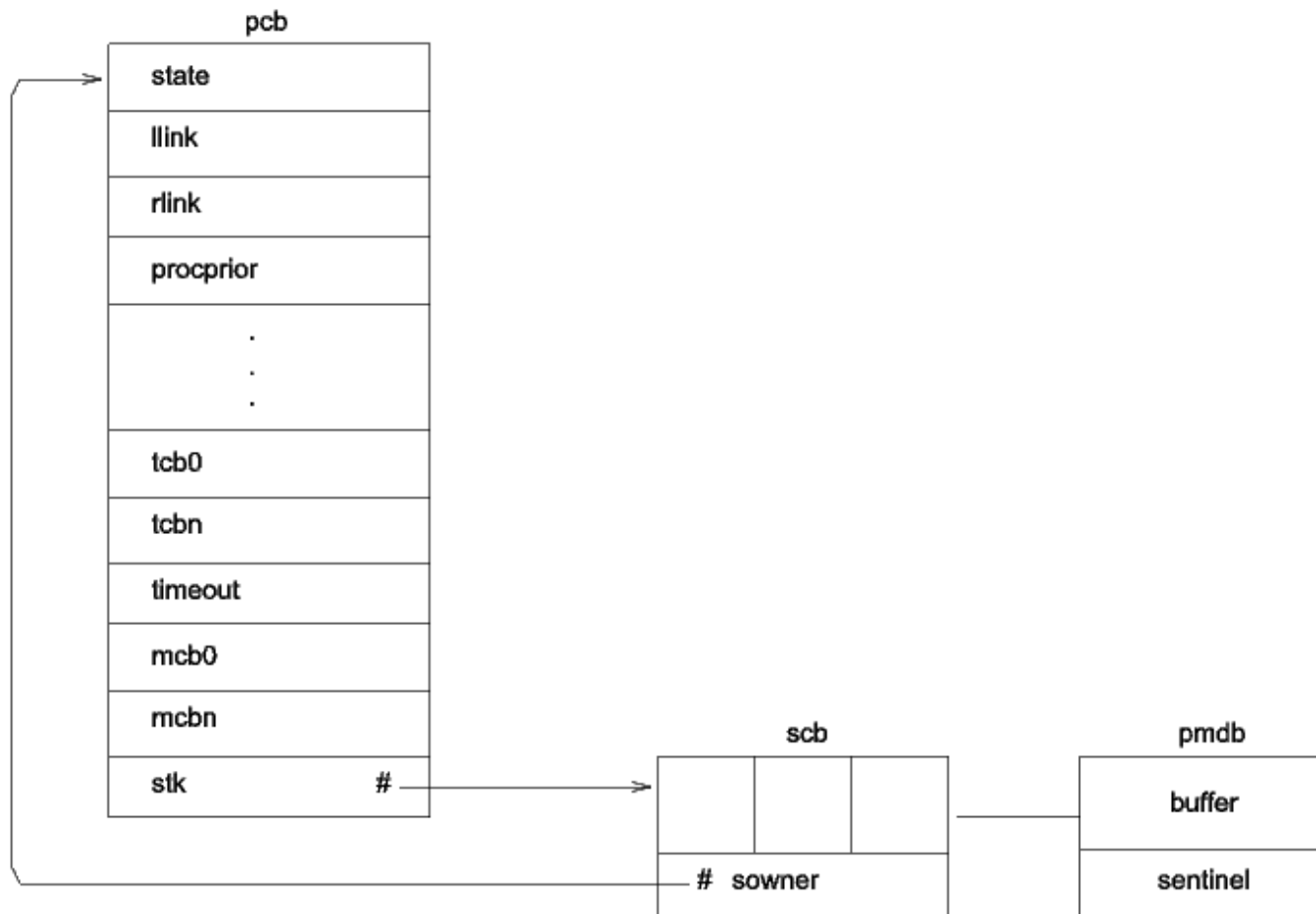


Figure 1 (SCB-1) Stack and Process Association

3. AUDIT CHECKS

The SCB audit does not use any audit driver. All error codes are described. The list provides a high-level view of the order in which these checks are made.

If the audit has just been entered:

ZAP = Check for system interrupt handling.

If the idle list is being audited:

LNK2ALST = Range check OSIfscb.

ALNK2AQUE = Semantic check of OSIfscb.

BLNKINQUE = Semantic check of OSffscb.

LNKIALST = Range check SCB.llink.

PT2PT = Point-to/point-back check between SCBs.

BADOWNER = Check SCB.sowner.

WRONGLIST = Check for non-PMDB SCBs.

If a PCB's SCB is being checked:

LNK2ALST = Range check PCB.stk.

ALNK2QUE = Point-to/point-back check between PCB and SCB.

LNKIALST = Range check SCB.link.

TYPE = Check stack type.

PMDBUSE = Check for a system process using a PMDB.

If the audit is looking for lost SCBs:

INVOWNER = Range check potentially lost SCB's sowner field.

INV = Check for lost SCBs.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ALNK2QUE

Error Code:	ALNK2QUE
Error Description:	The index to the last element in the free list of SCBs (OSIfscb), or the "stk" field of a PCB, indicates an SCB which either has an incorrect owner (not OSNULL for the free list, different from the PCB number in all other cases), or has a non-null (not OSNULL) link field.
Possible Error Effect:	For the free list, this could mean lost SCBs (leading to overload situations), or non-idle SCBs being taken for new processes (affecting the process to which they rightfully belong). For SCBs attached to PCBs, this problem could cause the owning process to use the wrong stack. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the SCB audit in directed mode. It is also likely that SCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of SCBs, and the audit is running in segmented mode, the audit will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list immediately. The list will be rebuilt by verifying OSffscb (index to the first free SCB) and, if it is ok, verifying the list from the top down as far as possible. If the SCB is attached to a PCB the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the SCB.
<i>Bad Data</i> =	The owner of the list we're trying to access (either OSNULL or a PCB index).
<i>Good Data</i> =	Contents of the SCB's sowner field.
<i>Logical Index</i> =	The SCB's index (= OSIfscb in the free list case, = PCB.stk otherwise).
<i>Dump 1</i> =	The SCB.
<i>Dump 2</i> =	The owning PCB or nothing if the free list was being checked.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	An SCB was found linked in the free list whose sowner field was not OSNULL. This error should only fire on the free list since at most one SCB is allowed per PCB, and checks such as ALNK2QUE would catch this problem before the BADOWNER check is made. If, somehow, this error is taken elsewhere (indicated by the KEY, dumped, being something other than OSNULL), then a linked list of SCBs has been attached to a PCB. (This error would also be caught and corrected using a BLNKINQUE error).
Possible Error Effect:	If the stack is actually in use by some process, then that process may be purged using checks made later in this audit. INV errors are also likely to show up.
Corrective Action Taken:	Relink the list around the bad SCB, and let an INV error fire to put it back on the idle list, if that's where it really belongs. In this is not the free list, the owning process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SCB.
<i>Bad Data =</i>	Contents of the SCB's sowner field.
<i>Good Data =</i>	The owner of the list the SCB was found in (should be OSNULL, otherwise a PCB index).
<i>Logical Index =</i>	The SCB's index (= OSffscb on the free list, PCB.stk otherwise).
<i>Dump =</i>	The SCB.

Table 3 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	While traversing a list (bottom up), an SCB was found with its llink field = OSNULL, but was not the first SCB in the list according to the list owner. Specifically, if the SCB was found on the free list, it was not indicated by OSffscb (index to the first free SCB). If the SCB was found on a list belonging to a PCB, then more than one SCB was attached to this PCB. This is not allowed.
Possible Error Effect:	The free list is broken, OSffscb is corrupt, or a PCB has too many stacks. The last case could corrupt the process and is a waste of resources. This could have any of the effects of an ALNK2QUE, LNKIALST or PT2PT error.
Corrective Action Taken:	If this is the free list of SCBs and the audit is running in segmented mode, escalate to directed mode and rebuild the list. If the audit is already running in directed mode then rebuild the list right away. The list will be rebuilt by verifying OSffscb (index to the first free SCB) and, if it is ok, verifying the list from the top down as far as possible. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SCB.
<i>Bad Data =</i>	Index of the SCB the list's owner says is first on the list (OSffscb or PCB.stk).
<i>Good Data =</i>	The owner of the list the SCB was found in (either OSNULL or a PCB index).
<i>Logical Index =</i>	The SCB's index.
<i>Dump 1 =</i>	The SCB.
<i>Dump 2 =</i>	The owning PCB or nothing if the free list was being checked.

Table 4 Error Code: INV

Error Code:	INV
Error Description:	An SCB has been found, which is not linked to any PCB or the free list.

Possible Error Effect:	The SCB is lost and inaccessible to the system. If enough SCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	The SCB is cleared and linked to the idle list. In particular, its sowner field is set to OSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SCB.
<i>Bad Data =</i>	0 (Indicates that the audit did not find the SCB on any list).
<i>Good Data =</i>	1 (Indicates that the audit should have found the SCB on some list).
<i>Logical Index =</i>	The SCB's index.
<i>Dump =</i>	The SCB.

Table 5 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	An SCB has been found, which the audit did not see on any list. Before taking an INV error, the audit will verify that the SCB is truly lost. For example, if the audit were running in segmented mode, the SCB may have changed lists while the audit took a timebreak. In this case, the audit would not have seen the SCB before now. The audit uses the SCB's sowner field to determine which list to check. This error says that the sowner field is out of range (not OSNULL and not a PCB index).
Possible Error Effect:	The SCB is effectively lost and inaccessible to the system. If enough SCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	If possible, the SCB is unlinked from the list its in (the linkages of the SCBs it points to are reset). In any case, the SCB is cleared and linked to the idle list. In particular, its sowner field is set to OSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SCB.
<i>Bad Data =</i>	Contents of the SCB's sowner field.
<i>Good Data =</i>	OSNULL.
<i>Logical Index =</i>	The SCB's index.
<i>Dump =</i>	The SCB.

Table 6 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Either the tail of the SCB free list (OSlfsb) was null when the head (OSffscb) was not or the tail of the SCB free list was out of range or a PCB's "stk" field was out of range.
Possible Error Effect:	If the bad index was found in a PCB, that process will be unable to access its stack. The results of this depend on the process involved. If the free list is involved, this could result in lost SCBs which could, eventually, lead to OSDS SCB overloads.
Corrective Action Taken:	If the bad value was found in a PCB, that process will be purged. Otherwise, if the audit is running segmented, it will escalate to directed mode. If the audit is already running in directed mode it will attempt to rebuild the idle list from the top (OSffscb) down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The out of range SCB index.
<i>Good Data =</i>	The source of the bad value (a PCB index or OSNULL for the idle list).
<i>Logical Index =</i>	OSNULL.
<i>Dump =</i>	The PCB (if any) which contained the bad index.

Table 7 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	While traversing a list, an SCB was found with an out of range llink field (not OSNULL and not an SCB index). This should only occur when checking the free list. Otherwise a PCB has a linked list of SCBs attached to it (using PCB.stk) instead of just one or none.
Possible Error Effect:	Either the free list is broken, which could result in lost SCBs (leading to overload situations), or the PCB (which has too many stacks attached to it) could be using the wrong one. In the free list case because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the SCB audit in directed mode. It is also likely that SCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of SCBs and the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying OSffscb (index to the first free SCB) and, if possible, verifying the list from the top down and then linking it with the segment (bottom up) the audit has already checked. If the list is attached to a PCB, the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SCB.
<i>Bad Data =</i>	The SCB's llink field.
<i>Good Data =</i>	OSSCBNUM (llink should have been OSNULL or less than this value).
<i>Logical Index =</i>	The SCB index.
<i>Dump =</i>	The SCB.

Table 8 Error Code: PMDBUSE

Error Code:	PMDBUSE
Error Description:	A system process using a PMDB was found when there are non-PMDB SCBs available.
Possible Error Effect:	A terminal process which needs a PMDB might not be able to get one.
Corrective Action Taken:	The system process with a PMDB will be purged. OSDs will restart it using the stack from the idle list which does not have a PMDB associated with it.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SCB.
<i>Bad Data =</i>	Contents of the SCB's sowner field.
<i>Good Data =</i>	The PMDB's index (same as the SCB's index).
<i>Logical Index =</i>	The SCB's Index.
<i>Dump =</i>	The SCB and the PCB.

Table 9 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	While traversing a list, an SCB was found whose llink field indicated an SCB whose rlink field was not equal to the first SCB's index. This should only occur when checking the free list. Otherwise a PCB has a linked list of SCBs attached to it (using PCB.stk) instead of just one or none.
Possible Error Effect:	Either the free list is broken which could result in lost SCBs (leading to overload situations), or the PCB (which has too many stacks attached to it) could be using the wrong one. In the free list case because of defensive checks in OS, this condition is much more likely to result in asserts from OS, which schedule the SCB audit

	in directed mode.
	It is also likely that SCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of SCBs and the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by verifying OSffscb (index to the first free SCB) and, if it is ok, verifying the list from the top down as far as possible. The audit will then link this segment (top down) with the segment (bottom up) it had already checked.
	If the list is attached to a PCB the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the SCB.
<i>Bad Data</i> =	The SCB's llink field.
<i>Good Data</i> =	The rlink field of the SCB indexed by "BAD DATA".
<i>Logical Index</i> =	The SCB's index.
<i>Dump</i> =	The SCB.

Table 10 Error Code: TYPE

Error Code:	TYPE
Error Description:	In the CMP, a large stack is being used by a process other than recent change.
Possible Error Effect:	A recent change terminal process might not be able to get the larger stack size it needs. This could cause failures when attempting to perform recent changes.
Corrective Action Taken:	The process which is using the large stack will be purged, thus freeing the SCB and associated stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the SCB.
<i>Bad Data</i> =	Program identifier of the process which owns the SCB (OSprogtab[PCB.progi]).
<i>Good Data</i> =	The program identifier for recent change processes (GLPGDBRC).
<i>Logical Index</i> =	Index of the SCB.
<i>Dump 1</i> =	The SCB.
<i>Dump 2</i> =	The PCB which owns the SCB.

Table 11 Error Code: WRONGLIST

Error Code:	WRONGLIST
Error Description:	While traversing the SCB idle list, a non-PMDB SCB was found. This implies that there is a system process unnecessarily using a PMDB.
Possible Error Effect:	A terminal process which needs a PMDB might not be able to get one.
Corrective Action Taken:	The SCB is unlinked from the list (the linkages of the SCB it points to are reset) and relinked back to the end of the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the SCB.
<i>Bad Data</i> =	OSNULL.
<i>Good Data</i> =	Number of non-PMDB SCBs seen on the SCB idle list so far.
<i>Logical Index</i> =	The SCB's Index.
<i>Dump</i> =	The SCB.

Table 12 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as a result of a write protect, out of range read, or invalid operation, by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will idle the last SCB it was looking at (if any), abort and be restarted in directed mode. Otherwise, it will just be restarted in directed mode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the SCB (if any).
<i>Bad Data</i> =	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index of the SCB (if any).
<i>Dump</i> =	The SCB (if any).

SCCPPC

Software Release: 5E14 and later

1. SCCP POINT CODES (SCCPPC) AUDIT

This audit steps through the RLSCCP_PC (SCCP point codes relation) and checks the data in it against the RLSCCPSTAT (SCCP point codes status relation).

2. DATA STRUCTURES

For details on the fields contained in the RLSCCP_PC and the RLSCCPSTAT relations refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	The audit attempts to read a rISCCPSTAT tuple but fails.
Possible Error Effect:	Application code cannot inform signaling connection control point (SCCP) management about congestion status for the destination point code (DPC) associated with the rISCCPSTAT tuple.
Corrective Action Taken:	The related audit SCCPSTAT is scheduled and attempts to insert the missing rISCCPSTAT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rISCCP_PC tuple.
<i>Bad Data =</i>	Dynamic point code index (rISCCPSTAT.sccpci); key of dynamic rISCCPSTAT tuple.
<i>Good Data =</i>	Static point code index (rISCCP_PC.sccpci).
<i>Logical Index =</i>	NA.
<i>Dump =</i>	Contents of rISCCP_PC tuple.

Table 2 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	The audit detects rISCCPSTAT.sccp_dpc does not match rISCCP_PC.sccp_pc.
Possible Error Effect:	The rISCCPSTAT tuple represents the incorrect DPC which means that SCCP cannot get the correct information about the DPC.
Corrective Action Taken:	The audit sets rISCCPSTAT.sccp_dpc equal to rISCCP_PC.sccp_pc and notifies application code about the correction.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Dynamic point code index (rISCCPSTAT.sccp_dpc).
<i>Good Data =</i>	Static point code (rISCCP_PC.sccp_pc).
<i>Logical Index =</i>	Dynamic point code index (rISCCPSTAT.sccpci); key of dynamic rISCCPSTAT

	tuple.
<i>Dump 1</i> =	Contents of rISCCP_PC tuple.
<i>Dump 2</i> =	Contents of rISCCPSTAT tuple.

SCCPSTAT

Software Release: 5E14 and later

1. ROUTING DATA STATUS FOR SCCP DESTINATIONS (SCCPSTAT) AUDIT

This audit steps through the signaling connection control point routing status relation (RLSCCPSTAT) and checks the data in it against the SCCP point code relation (RLSCCP_PC), the destination point code status relation (RLDPC_STAT), cluster status relation (RLCLU_STAT) and the network status relation (RLNET_STAT).

2. DATA STRUCTURES

For details on the fields contained in the RLSCCPSTAT, RLSCCP_PC, RLDPC_STAT, RLCLU_STAT and RLNET_STAT relations refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_DPC

Error Code:	BAD_DPC
Error Description:	If Error Address is: 1 = The audit detects that rISCCPSTAT.sccp_dpc does not match the rIDPC_STAT.dpc of the owning rIDPC_STAT tuple. 2 = The rISCCPSTAT.sccp_dpc does not belong to the rICLU_STAT.cluster of the rICLU_STAT tuple which owns the rISCCPSTAT tuple. 3 = The rISCCPSTAT.sccp_dpc does not belong to the rINET_STAT.network of the rINET_STAT tuple which owns the rISCCPSTAT tuple.
Possible Error Effect:	In all cases, signaling connection control point (SCCP) may fail to correctly handle congestion/accessibility for the destination point code (DPC) associated with the rISCCPSTAT tuple.
Corrective Action Taken:	In all cases, the audit unlinks the rISCCPSTAT tuple from any still valid linkage, sets non-0 fields to the correct values and notifies application code about the recovered tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1 = Corresponds to Error Description. 2 = Corresponds to Error Description. 3 = Corresponds to Error Description.

<i>Bad Data =</i>	rISCCPSTAT.sccp_dpc.
<i>Good Data =</i>	If the Error Address is: 1 = rIDPC_STAT.dpc. 2 = Cluster to which the rISCCPSTAT.sccp_dpc should belong. 3 = Network to which the rISCCPSTAT.sccp_dpc should belong.
<i>Logical Index =</i>	rISCCPSTAT.sccpci.
<i>Dump 1 =</i>	Contents of rISCCP_STAT tuple.
<i>Dump 2 =</i>	1 = Contents of rIDPC_STAT tuple. 2 = Contents of rICLU_STAT tuple. 3 = Contents of rINET_STAT tuple.

Table 2 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	If Error Address is: 1 = The audit attempts to read the owning rIDPC_STAT tuple of the current rISCCPSTAT tuple but fails. 2 = The audit attempts to read the owning rICLU_STAT tuple of the current rISCCPSTAT tuple but fails. 3 = The audit attempts to read the owning rINET_STAT tuple of the current rISCCPSTAT tuple but fails.
Possible Error Effect:	In all cases, message transfer part (MTP) may fail to inform SCCP management about the congestion/accessibility of the DPC associated with the rISCCPSTAT tuple.
Corrective Action Taken:	In all cases, the audit unlinks the rISCCPSTAT tuple from any still valid linkage, sets non-0 fields to the correct values and notifies application code about the recovered tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1 = Corresponds to Error Description. 2 = Corresponds to Error Description. 3 = Corresponds to Error Description.
<i>Bad Data =</i>	If the Error Address is: 1 = rIDPC_STAT.dpci. 2 = rICLU_STAT.clix. 3 = rINET_STAT.netid.
<i>Good Data =</i>	If the Error Address is: 1 = rISCCPSTAT.routeset.st_ownk - DBSFDPC_STAT.

	2 = rISCCPSTAT.routeset.st_ownk - DBSFCLU_STAT.
	3 = rISCCPSTAT.routeset.st_ownk - DBSFNET_STAT.
<i>Logical Index</i> =	rISCCPSTAT.sccpci.
<i>Dump</i> =	Contents of rISCCPSTAT tuple.

Table 3 Error Code: INV

Error Code:	INV
Error Description:	The audit detects an invalid value in the rISCCPSTAT.routeset_type.
Possible Error Effect:	The MTP may fail to inform SCCP management about the congestion/accessibility of the DPC associated with the rISCCPSTAT tuple.
Corrective Action Taken:	The audit reads the next rISCCPSTAT tuple. If this error rolls, notify Customer Technical Support. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISCCPSTAT tuple.
<i>Bad Data</i> =	rISCCPSTAT.routeset_type.
<i>Good Data</i> =	rISCCPSTAT.sccp_dpc.
<i>Logical Index</i> =	rISCCPSTAT.sccpci.
<i>Dump</i> =	Contents of rISCCPSTAT tuple.

Table 4 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	If Error Address is: 1 = The value of rISCCPSTAT.routeset_type (CCDPC_RTSET) indicates that the rISCCPSTAT tuple should be owned by a rIDPC_STAT tuple but it is not. 2 = The value of rISCCPSTAT.routeset_type (CCCLU_RTSET) indicates that the rISCCPSTAT tuple should be owned by a rICLU_STAT tuple but it is not. 3 = The value of rISCCPSTAT.routeset_type (CCNET_RTSET) indicates that the rISCCPSTAT tuple should be owned by a rINET_STAT tuple but it is not.
Possible Error Effect:	In all cases, MTP may fail to inform SCCP management about the congestion/accessibility of the DPC associated with the rISCCPSTAT tuple.
Corrective Action Taken:	In all cases, the audit unlinks the rISCCPSTAT tuple from any still valid linkage, sets non-0 fields to the correct values and notifies application code about the recovered tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1 = Corresponds to Error Description. 2 = Corresponds to Error Description. 3 = Corresponds to Error Description.
<i>Bad Data</i> =	rISCCPSTAT.routeset_type.
<i>Good Data</i> =	rISCCPSTAT.sccp_dpc.
<i>Logical Index</i> =	rISCCPSTAT.sccpci.
<i>Dump</i> =	Contents of rISCCPSTAT tuple.

Table 5 Error Code: PCI_BAD

Error Code:	PCI_BAD
Error Description:	The audit uses rISCCPSTAT.sccp_dpc to find a matching tuple in relation RLSCCP_PC. The audit succeeds but detects the rISCCPSTAT.sccpci does not

	match rISCCP_PC.sccpci.
Possible Error Effect:	The MTP may fail to inform SCCP management about the congestion/accessibility of the DPC associated with the rISCCPSTAT tuple.
Corrective Action Taken:	The audit has the incorrect rISCCPSTAT deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISCCP_PC tuple.
<i>Bad Data</i> =	rISCCPSTAT.sccpci.
<i>Good Data</i> =	rISCCP_PC.sccpci.
<i>Logical Index</i> =	rISCCPSTAT.sccpci.
<i>Dump 1</i> =	Contents of rISCCPSTAT tuple.
<i>Dump 2</i> =	Contents of rISCCP_PC tuple.

SCMDB

Software Release: 5E10 and later

1. SHARED CALL MODEL DATA BLOCK (SCMDB) AUDIT

The SCMDB audit insures the internal consistency of the RLSCMDB relation and performs linkage checks between the RLSCMDB relation and the RLDB_HEADC and RLCLDB relations.

2. DATA STRUCTURES

The RLSCMDB relation is a dynamic indexed sequential relation that is used to save per call information for all the shared associated terminals in the key system group. When the rISCMDB tuple is busy it is linked to an rICLDB tuple. All idle rISCMDB tuples are linked to an idle list that indicates a specified tuple in the RLDB_HEADC relation.

3. AUDIT CHECKS

The SCMDB audit is invoked through audit control which calls the SCMDB audit entry function. The function invokes the indexed sequential driver (DRINASEK) to step through all tuples in the RLSCMDB relation in a sequential manner.

If all the linkage and semantic checks pass then the driver steps to the next tuple and continues another cycle. If any checks fail, recovery is started.

The following is a list of semantic checks that are performed:

- (a) Since RLSCMDB is only used by small shared key system groups, the rISCMDB.lrg_grp field must indicate a small group (zero).
- (b) If the rISCMDB tuple is on the idle link list, then the linkage to the rICLDB tuple should be null.

NOTE: The following checks are only made if the rISCMDB tuple is on the busy list:

- (c) The terminal process one field in the rISCMDB should be associated with a live process.
- (d) If the bridge bit in the rISCMDB tuple is set to DBNO then the terminal process two array in the rISCMDB tuple should be null.
- (e) If the bridge bit in the rISCMDB tuple is set to DBYES then the terminal process two array (rISCMDB.tp2[]) in the rISCMDB tuple should contain at least one live process.

4. ERROR HANDLING

When an error is detected either in the semantic check or in the linkage checking routine, a failure return is returned back to the indexed sequential driver. If the rISCMDB tuple is linked onto the idle list, the rIADB is unlinked from that list. If the rISCMDB tuple is mutilated in such a way that it cannot be unlinked from the list the indexed sequential driver will patch the list. Recovery is performed on the rISCMDB tuple that was found in error or any rISCMDB tuples that were left hanging as a result of a list being patched.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	For an inactive bridge/conference there is a PID of the TP of a bridged party which is non-null.
Possible Error Effect:	The call will have an incorrect path.
Corrective Action Taken:	The tuple is reinitialized and returned to the idle list, purging any associated calls.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSCMDB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rSCMDB tuple.
<i>Dump =</i>	The contents of the rSCMDB tuple.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The OSPID of the TP of the terminal actively on the call is not an active process for this module.
Possible Error Effect:	The rSCMDB tuple is connected to a dead process and cannot be used for other calls.
Corrective Action Taken:	The process owning the rSCMDB will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSCMDB tuple.
<i>Bad Data =</i>	Process number of tp1 for rSCMDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rSCMDB tuple.
<i>Dump =</i>	The contents of the rSCMDB tuple.

Table 3 Error Code: FPID_DEAD

Error Code:	FPID_DEAD
Error Description:	The bridge bit is set to DBYES and the process (tp2) is not null, but tp2 is not an active process.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rSCMDB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSCMDB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key to the rSCMDB tuple.
<i>Dump =</i>	The contents of the rSCMDB tuple.

Table 4 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	All tuples in RLSCMDB are used for small key system groups; therefore, all tuples in this relation should reflect this in the "lrg_grp" field. This rSCMDB tuple contains garbage in its lrg_grp field.
Possible Error Effect:	The rSCMDB tuple may contain other erroneous data.
Corrective Action Taken:	The audit will idle the rSCMDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rSCMDB tuple.
<i>Bad Data =</i>	Owner link of rSCMDB tuple.
<i>Good Data =</i>	0
<i>Logical Index =</i>	Logical key of rSCMDB tuple.
<i>Dump =</i>	The contents of the rSCMDB tuple.

Table 5 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rSCMDB tuple is on an invalid list.

Possible Error Effect:	The rSCMDB tuple is in a limbo condition which is unusable by call processing.
Corrective Action Taken:	The audit will idle the rSCMDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSCMDB tuple.
<i>Bad Data</i> =	Owner link of rSCMDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSCMDB tuple.
<i>Dump</i> =	The contents of the rSCMDB tuple.

Table 6 Error Code: LIVEPID

Error Code:	LIVEPID
Error Description:	The TP (tp2) is null when the bridge bit is set to DBYES. The TP should not be null.
Possible Error Effect:	Lost resources.
Corrective Action Taken:	The rSCMDB tuple will be put on an idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSCMDB tuple.
<i>Bad Data</i> =	The process number for tp2.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key to the rSCMDB tuple.
<i>Dump</i> =	The contents of the rSCMDB tuple.

Table 7 Error Code: PID

Error Code:	PID
Error Description:	When the rSCMDB tuple is busy, the PID should not be null. However, in this case the PID of tp1 is null.
Possible Error Effect:	Incorrect call path.
Corrective Action Taken:	The rSCMDB tuple will be placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSCMDB tuple.
<i>Bad Data</i> =	Contents of tp1 field in rSCMDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSCMDB tuple.
<i>Dump</i> =	The contents of the rSCMDB tuple.

Table 8 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rSCMDB tuple has a nonzero linkage in one of its link blocks.
Possible Error Effect:	Resource is linked to a rSCMDB tuple on the idle list and could be unusable by call processing.
Corrective Action Taken:	The rSCMDB tuple will be placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSCMDB tuple.
<i>Bad Data</i> =	Nonzero value of linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rSCMDB tuple.
<i>Dump</i> =	NA.

SCNMSK-A

Software Release: 5E10 - 5E11

1. SCAN MASK (SCNMSK) AUDIT

Peripheral control (PC) scanning detects input signals from the switching periphery in the switching module (SM) and reports them to appropriate processes. These input signals include line and trunk originations, digit collection, service request signals, and other supervision signals. Two types of scanning are audited: control interface (CI) scanning and summary scanning.

Control interface scanning, shown in Figure 1, detects input signals from SM peripherals such as the digital service unit (DSU), metallic service unit (MSU), line unit (LU), digital carrier line unit (DCLU), digital line trunk unit (DLTU), and trunk unit (TU). The scan mask audit (SCNMSK) checks CI scanning. Summary scanning detects input signals from line concentrators (grids) connected to the line unit. The line concentrator scan mask audit (LCSMSK) and the line concentrator report inhibit mask audit (LCGREP) check summary scanning.

2. DATA STRUCTURES

The data structures verified by this audit are the scan mask array (Clscnmsk), the switch maintenance equipment status table relation (rSMEST), and the scan mask copy array (Slscnmsk).

3. AUDIT CHECKS

The scan mask audit calls the bit mask driver (DRBMSK) to step through the Clscnmsk array. The DRBMSK driver calculates a controller circuit key, retrieves the rSMEST tuple with that key, and calls a semantic check function to perform the following checks for each bit in Clscnmsk:

- (a) The audit uses the information in the rSMEST tuple to predict the state of the Clscnmsk bit and the Slscnmsk bit.
- (b) The audit verifies the predicted state of the Clscnmsk bit and the Slscnmsk bit.
- (c) If the Clscnmsk bit or the Slscnmsk bit differs from the predicted value, the difference is noted. If the difference still exists, the next time the audit runs, the error is reported and the bit is changed to agree with the status in rSMEST.

If the audit is called in the directed mode, the errors noted by the caller (by means of assert 22024) are corrected in one pass of the audit.

4. ERROR HANDLING

If an inconsistency exists between Clscnmsk or Slscnmsk and rSMEST for two passes of the audit (except when the audit runs in the directed mode), both Clscnmsk and Slscnmsk are changed to agree with rSMEST.

The audit produces these error codes and those generated by the DRBMSK driver.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: GRBG

Error Code:	GRBG
Error Description:	The upper eight bits of the Slscnmsk[] are not used at this time. These bits

	should all be set to zeros initially and never change. The bits are audited because the integrity monitor checks them and for wild writes.
Possible Error Effect:	Causes rolling ASSERTs from the integrity monitor.
Corrective Action Taken:	The upper eight bits in Slscnmsk[] are reset to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Good value of bit.
<i>Logical Index</i> =	Number of element in Slscnmsk[].
<i>Dump</i> =	NA.

Table 2 Error Code: INHBOTH

Error Code:	INHBOTH
Error Description:	The audit found Clscnmsk[] and Slscnmsk[] scan bits set to inhibit when rSMEST says that scanning should be allowed.
Possible Error Effect:	Equipment which should be scanned will not be scanned.
Corrective Action Taken:	Scanning will be turned on for that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 3 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	Scanning was found incorrectly allowed for a PCDCLCC circuit in Slscnmsk[].
Possible Error Effect:	Scanning will be done on this circuit which is in a manually removed state.
Corrective Action Taken:	Scanning is turned off for that circuit in Slscnmsk[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Slscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 4 Error Code: INV

Error Code:	INV
Error Description:	The audit was called in DEMAND (DIRECTED) mode to immediately correct an inconsistency between the Clscnmsk[] and Slscnmsk[] that has now existed for at least 5 seconds. Slscnmsk[] was found to be wrong.
Possible Error Effect:	Scanning may be incorrectly handled by others who use Slscnmsk[].
Corrective Action Taken:	The audit reads a structure in the SI subsystem to see which bit is inconsistent and corrects it immediately according relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Slscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 5 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	A database no-match was returned when this circuit was read from the database containing the RLCKTDATA relations. A bit was found to be set when it should be cleared, or vice versa.
Possible Error Effect:	Unnecessary scanning might be done on this circuit which could possibly cause

	PC foreground to reset the SM.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple for the circuit not found.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Number of the circuit in the tuple.
<i>Dump</i> =	NA.

Table 6 Error Code: M_NOTINH

Error Code:	M_NOTINH
Error Description:	Scanning was found incorrectly allowed for this circuit in Clscnmsk[].
Possible Error Effect:	This will cause PC foreground to scan circuits which shouldn't be scanned now. This could cause the SM to go into a reset.
Corrective Action Taken:	Scanning is turned off for this circuit in Clscnmsk[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 7 Error Code: PICB

Error Code:	PICB
Error Description:	The rPICB tuple could not be read from the database.
Possible Error Effect:	Many and unknown.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPICB tuple we tried to read.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	PICB number we tried to read from rPICB.
<i>Dump</i> =	Value of the rPICB we tried to read.

Table 8 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The rSMEST tuple could not be read from the database.
Possible Error Effect:	Many and unknown.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple we tried to read.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	RLSMEST relation used for the read.
<i>Logical Index</i> =	Value of the rSMEST we tried to read.
<i>Dump</i> =	NA.

Table 9 Error Code: SALW

Error Code:	SALW
Error Description:	A bit in Clscnmsk[] says that scanning is allowed for a common board when the maintenance information in the corresponding rSMEST tuple says scanning should not be allowed.
Possible Error Effect:	PC foreground is scanning a rPICB that it shouldn't. This causes extra work for the system and may result in CI interrupts. These in turn could cause system initializations.
Corrective Action Taken:	The incorrect bit in Clscnmsk[] is reset to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.

<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 10 Error Code: SCNMSK

Error Code:	SCNMSK
Error Description:	The upper eight bits of the Clscnmsk[] are not used at this time. These bits should all be set to zeros initially and never change. The bits are audited because the integrity monitor checks them and for wild writes.
Possible Error Effect:	Causes rolling ASSERTs from the integrity monitor.
Corrective Action Taken:	The upper eight bits in Clscnmsk[] are reset to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Good value of bit.
<i>Logical Index</i> =	Number of element in Clscnmsk[].
<i>Dump</i> =	NA.

Table 11 Error Code: SICPY

Error Code:	SICPY
Error Description:	The corresponding bits in Slscnmsk[] and rSMEST are unequal. It is assumed that the values in rSMEST are correct.
Possible Error Effect:	The integrity monitor will ASSERT periodically (and frequently).
Corrective Action Taken:	The SI copy is corrected to correspond with rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Slscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Name of the circuit being audited.

Table 12 Error Code: SINH

Error Code:	SINH
Error Description:	A bit in Clscnmsk[] says that scanning is inhibited for a common board when the maintenance information in the corresponding rSMEST tuple says scanning should be allowed.
Possible Error Effect:	PC foreground isn't scanning a rPICB that it should. This causes call processing, terminal maintenance, peripheral fault recovery, and peripheral maintenance failures.
Corrective Action Taken:	The incorrect bit in Clscnmsk[] is reset to allow scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 13 Error Code: SMEST

Error Code:	SMEST
Error Description:	A database no-match was returned when this circuit was read from the database containing the rSMEST relations.
Possible Error Effect:	Unnecessary scanning might be done on this circuit which could possibly cause PC foreground to reset the SM.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple that we tried to read.

<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Unit number of the circuit in the tuple.
<i>Dump</i> =	NA.

Table 14 Error Code: STATE

Error Code:	STATE
Error Description:	Scanning was incorrectly found turned off for two routine audit cycles for for circuit types PCDCLCC or PCDCLDC.
Possible Error Effect:	These circuits will not be scanned for originations.
Corrective Action Taken:	Scanning is turned on for the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rISMEST tuple.

Table 15 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUscntim, used for transient timing may have occurred.
Possible Error Effect:	This will cause the SCNMSK audit not to run on this invocation of the audit. The SCNMSK audit should not run faster than every 10 seconds.
Corrective Action Taken:	AUscntim will be updated with the current OSDS ITIME (OSnow) and the audit will not run until the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the transient timing variable (AUscntim).
<i>Bad Data</i> =	Time difference between the current ITIME (OSnow) and the transient timing variable (AUscntim).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	Content of the current OSDS ITIME (OSnow).
<i>Dump 2</i> =	Content of the transient timing variable (AUscntim).

Table 16 Error Code: UNEQUIP

Error Code:	UNEQUIP
Error Description:	The bits in Clscnmsk[] and Slscnmsk[] were found to be not zero when either the audit was unable to read rUNITDATA for a given circuit or the rUNITDATA read for a given circuit showed it was unequipped. Whenever we are unable to read rUNITDATA or when the given circuit is unequipped the bits in Clscnmsk[] and Slscnmsk[] should always be zero.
Possible Error Effect:	Scanning may be incorrectly handled by others who use Clscnmsk[] or Slscnmsk[].
Corrective Action Taken:	The audit will set the scan bits to zero and clear the touch bits as well.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of eqstat in rUNITDATA if tuple is retrieved.
<i>Logical Index</i> =	Bit position in element.
<i>Dump 1</i> =	Circuit index used as key to read RLUNITDATA.
<i>Dump 2</i> =	Copy of corresponding rPICB tuple.

Table 17 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The audit was called in DEMAND (DIRECTED) mode to immediately correct an

	inconsistency between the Clscnmsk[] and Slscnmsk[] that has now existed for at least 5 seconds. Clscnmsk[] was found to be wrong.
Possible Error Effect:	Scanning may be incorrectly handled by others who use Clscnmsk[].
Corrective Action Taken:	The audit reads a structure in the SI subsystem to see which bit is inconsistent and corrects it immediately according to derived scanning information from the corresponding RLSMEST relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rLSMEST tuple.

SCNMSK-B

Software Release: 5E12 and later

1. SCAN MASK (SCNMSK) AUDIT

Peripheral control (PC) scanning detects input signals from the switching periphery in the switching module (SM) and reports them to appropriate processes. These input signals include line and trunk originations, digit collection, service request signals, and other supervision signals. Two types of scanning are audited: control interface (CI) scanning and summary scanning.

Control interface scanning, shown in Figure 1, detects input signals from SM peripherals such as the digital service unit (DSU), metallic service unit (MSU), line unit (LU), digital carrier line unit (DCLU), digital line trunk unit (DLTU), and trunk unit (TU). The scan mask audit (SCNMSK) checks CI scanning. Summary scanning detects input signals from line concentrators (grids) connected to the line unit. The line concentrator scan mask audit (LCSMSK) and the line concentrator report inhibit mask audit (LCGREP) check summary scanning.

2. DATA STRUCTURES

The data structures verified by this audit are the scan mask array (CIsnmsk), the switch maintenance equipment status table relation (rSMEST), and the scan mask copy array (SIsnmsk).

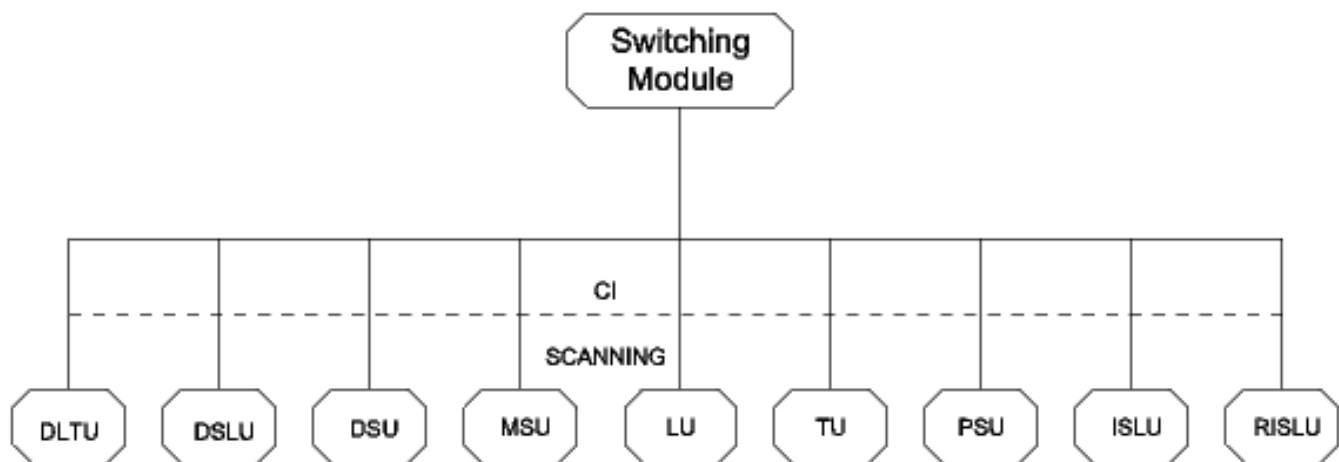


Figure 1 (SCNMSK-1) Control Interface (CI) Scanning

3. AUDIT CHECKS

The scan mask audit calls the bit mask driver (DRBMSK) to step through the CIsnmsk array. The DRBMSK driver calculates a controller circuit key, retrieves the rSMEST tuple with that key, and calls a semantic check function to perform the following checks for each bit in CIsnmsk:

- The audit uses the information in the rSMEST tuple to predict the state of the CIsnmsk bit and the SIsnmsk bit.
- The audit verifies the predicted state of the CIsnmsk bit and the SIsnmsk bit.
- If the CIsnmsk bit or the SIsnmsk bit differs from the predicted value, the difference is noted. If the difference still exists, the next time the audit runs, the error is reported and the bit is changed to agree with the status in rSMEST.

If the audit is called in the directed mode, the errors noted by the caller (by means of assert 22024) are corrected in one pass of the audit.

4. ERROR HANDLING

If an inconsistency exists between Clscnmsk or Slscnmsk and rSMEST for two passes of the audit (except when the audit runs in the directed mode), both Clscnmsk and Slscnmsk are changed to agree with rSMEST.

The audit produces these error codes and those generated by the DRBMSK driver.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: GRBG

Error Code:	GRBG
Error Description:	The upper eight bits of the Slscnmsk[] are not used at this time. These bits should all be set to zeros initially and never change. The bits are audited because the integrity monitor checks them and for wild writes.
Possible Error Effect:	Causes rolling asserts from the integrity monitor.
Corrective Action Taken:	The upper eight bits in Slscnmsk[] are reset to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Good value of bit.
<i>Logical Index</i> =	Number of element in Slscnmsk[].
<i>Dump</i> =	NA.

Table 2 Error Code: INHBOTH

Error Code:	INHBOTH
Error Description:	The audit found Clscnmsk[] and Slscnmsk[] scan bits set to inhibit when rSMEST says that scanning should be allowed.
Possible Error Effect:	Equipment which should be scanned will not be scanned.
Corrective Action Taken:	Scanning will be turned on for that circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 3 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	Scanning was found incorrectly allowed for a PCDCLCC circuit in Slscnmsk[].
Possible Error Effect:	Scanning will be done on this circuit which is in a manually removed state.
Corrective Action Taken:	Scanning is turned off for that circuit in Slscnmsk[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Slscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 4 Error Code: INV

Error Code:	INV

Error Description:	The audit was called in DEMAND (DIRECTED) mode to immediately correct an inconsistency between the Clscnmsk[] and Slscnmsk[] that has now existed for at least 5 seconds. Slscnmsk[] was found to be wrong.
Possible Error Effect:	Scanning may be incorrectly handled by others who use Slscnmsk[].
Corrective Action Taken:	The audit reads a structure in the SI subsystem to see which bit is inconsistent and corrects it immediately according relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Slscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rISMEST tuple.

Table 5 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	A database no-match was returned when this circuit was read from the database containing the RLCKTDATA relations. A bit was found to be set when it should be cleared, or vice versa.
Possible Error Effect:	Unnecessary scanning might be done on this circuit which could possibly cause PC foreground to reset the SM.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple for the circuit not found.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Number of the circuit in the tuple.
<i>Dump</i> =	NA.

Table 6 Error Code: M_NOTINH

Error Code:	M_NOTINH
Error Description:	Scanning was found incorrectly allowed for this circuit in Clscnmsk[].
Possible Error Effect:	This will cause PC foreground to scan circuits which shouldn't be scanned now. This could cause the SM to go into a reset.
Corrective Action Taken:	Scanning is turned off for this circuit in Clscnmsk[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rISMEST tuple.

Table 7 Error Code: PICB

Error Code:	PICB
Error Description:	The rPICB tuple could not be read from the database.
Possible Error Effect:	Many and unknown.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rPICB tuple that could not be read.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	PICB number that could not be read from rPICB.
<i>Dump</i> =	Value of the rPICB that could not be read.

Table 8 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The rISMEST tuple could not be read from the database.
Possible Error Effect:	Many and unknown.

Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple that could not be read.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	rSMEST relation used for the read.
<i>Logical Index</i> =	Value of the rSMEST that could not be read.
<i>Dump</i> =	NA.

Table 9 Error Code: SALW

Error Code:	SALW
Error Description:	A bit in Clscnmsk[] says that scanning is allowed for a common board when the maintenance information in the corresponding rSMEST tuple says scanning should not be allowed.
Possible Error Effect:	PC foreground is scanning a rPICB that it shouldn't. This causes extra work for the system and may result in CI interrupts. These in turn could cause system initializations.
Corrective Action Taken:	The incorrect bit in Clscnmsk[] is reset to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rSMEST tuple.

Table 10 Error Code: SCNMSK

Error Code:	SCNMSK
Error Description:	The upper eight bits of the Clscnmsk[] are not used at this time. These bits should all be set to zeros initially and never change. The bits are audited because the integrity monitor checks them and for wild writes.
Possible Error Effect:	Causes rolling asserts from the integrity monitor.
Corrective Action Taken:	The upper eight bits in Clscnmsk[] are reset to zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Good value of bit.
<i>Logical Index</i> =	Number of element in Clscnmsk[].
<i>Dump</i> =	NA.

Table 11 Error Code: SICPY

Error Code:	SICPY
Error Description:	The corresponding bits in Slscnmsk[] and rSMEST are unequal. It is assumed that the values in rSMEST are correct.
Possible Error Effect:	The integrity monitor will assert periodically (and frequently).
Corrective Action Taken:	The SI copy is corrected to correspond with rSMEST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Slscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Slscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Name of the circuit being audited.

Table 12 Error Code: SINH

Error Code:	SINH
Error Description:	A bit in Clscnmsk[] says that scanning is inhibited for a common board when the maintenance information in the corresponding rSMEST tuple says scanning should be allowed.
Possible Error Effect:	PC foreground isn't scanning a rPICB that it should. This causes call

	processing, terminal maintenance, peripheral fault recovery, and peripheral maintenance failures.
Corrective Action Taken:	The incorrect bit in Clscnmsk[] is reset to allow scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of element of Clscnmsk[] containing error.
<i>Bad Data =</i>	Value of element of Clscnmsk[] containing error.
<i>Good Data =</i>	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index =</i>	Name of the circuit in error.
<i>Dump =</i>	Copy of corresponding rLSMEST tuple.

Table 13 Error Code: SMEST

Error Code:	SMEST
Error Description:	A database no-match was returned when this circuit was read from the database containing the RLSMEST relations.
Possible Error Effect:	Unnecessary scanning might be done on this circuit which could possibly cause PC foreground to reset the SM.
Corrective Action Taken:	The scanning bits for that circuit in the scan masks are set to inhibit scanning.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the tuple that could not be read.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Unit number of the circuit in the tuple.
<i>Dump =</i>	NA.

Table 14 Error Code: STATE

Error Code:	STATE
Error Description:	Scanning was incorrectly found turned off for two routine audit cycles for for circuit types PCDCLCC or PCDCLDC.
Possible Error Effect:	These circuits will not be scanned for originations.
Corrective Action Taken:	Scanning is turned on for the circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of element of Clscnmsk[] containing error.
<i>Bad Data =</i>	Value of element of Clscnmsk[] containing error.
<i>Good Data =</i>	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index =</i>	Name of the circuit in error.
<i>Dump =</i>	Copy of corresponding rLSMEST tuple.

Table 15 Error Code: TIMESTAMP

Error Code:	TIMESTAMP
Error Description:	The operating systems (OSDS) ITIME (OSnow) seems to have moved backwards. Possible corruption of the global variable, AUscntim, used for transient timing may have occurred.
Possible Error Effect:	This will cause the SCNMSK audit not to run on this invocation of the audit. The SCNMSK audit should not run faster than every 10 seconds.
Corrective Action Taken:	AUscntim will be updated with the current OSDS ITIME (OSnow) and the audit will not run until the next invocation of the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the transient timing variable (AUscntim).
<i>Bad Data =</i>	Time difference between the current ITIME (OSnow) and the transient timing variable (AUscntim).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Content of the current OSDS ITIME (OSnow).
<i>Dump 2 =</i>	Content of the transient timing variable (AUscntim).

Table 16 Error Code: UNEQUIP

Error Code:	UNEQUIP

Error Description:	The bits in Clscnmsk[] and Slscnmsk[] were found to be not zero when either the audit was unable to read rUNITDATA for a given circuit or the rUNITDATA read for a given circuit showed it was unequipped. Whenever rUNITDATA can not be read or when the given circuit is unequipped the bits in Clscnmsk[] and Slscnmsk[] should always be zero.
Possible Error Effect:	Scanning may be incorrectly handled by others who use Clscnmsk[] or Slscnmsk[].
Corrective Action Taken:	The audit will set the scan bits to zero and clear the touch bits as well.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Slscnmsk[] containing error.
<i>Good Data</i> =	Value of eqstat in rUNITDATA if tuple is retrieved.
<i>Logical Index</i> =	Bit position in element.
<i>Dump 1</i> =	Circuit index used as key to read RLUNITDATA.
<i>Dump 2</i> =	Copy of corresponding rPICB tuple.

Table 17 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The audit was called in DEMAND (DIRECTED) mode to immediately correct an inconsistency between the Clscnmsk[] and Slscnmsk[] that has now existed for at least 5 seconds. Clscnmsk[] was found to be wrong.
Possible Error Effect:	Scanning may be incorrectly handled by others who use Clscnmsk[].
Corrective Action Taken:	The audit reads a structure in the SI subsystem to see which bit is inconsistent and corrects it immediately according to derived scanning information from the corresponding RLSMEST relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of element of Clscnmsk[] containing error.
<i>Bad Data</i> =	Value of element of Clscnmsk[] containing error.
<i>Good Data</i> =	Value of element of Clscnmsk[] with the single bit in error corrected.
<i>Logical Index</i> =	Name of the circuit in error.
<i>Dump</i> =	Copy of corresponding rLSMEST tuple.

SCPSM

Software Release: 5E11 and later

1. SCCP GLOBAL SM INFORMATION (SCPSM) AUDIT

SCPSM audit verifies values in the structure CCscp_sm_info against the static relation RLGSM_NET (global SM and network).

2. DATA STRUCTURES

For details on the fields contained in the RLGSM_NET relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: INFO_ERR

Error Code:	INFO_ERR
Error Description:	CCscp_sm_info has the signaling connection control point (SCCP) information about networks known by this switch. One or more of the fields in CCscp_sm_info does not equal its corresponding field in an rLGSM_NET tuple whose key (rLGSM_NET.gsm) equals the current global switching module (GSM) number (Gl_pcid).
Possible Error Effect:	SCCP may have network routing errors and/or accounting errors.
Corrective Action Taken:	The audit sets the following: CCscp_sm_info.net_num = gsm_tup.net_num. CCscp_sm_info.sccp_init = gsm_tup.sccp_init. CCscp_sm_info.sccp_type = gsm_tup.sccp_type. CCscp_sm_info.sig_type = gsm_tup.sig_type. CCscp_sm_info.sccp_prc = gsm_tup.sccp_prc. CCscp_sm_info.sccp_acc = gsm_tup.sccp_acc. CCscp_sm_info.sccp_scr = gsm_tup.actv_scr. CCscp_sm_info.allssn_s = Glallssn_s. CCscp_sm_info.scp_version = gsm_tup.scp_version.

	CCscp_sm_info.hop_cnt = gsm_tup.hop_cnt. CCscp_sm_info.reasmbtmr = gsm_tup.reasmbtmr. CCscp_sm_info.monitor = gsm_tup.monitor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rlGSM_NET.gsm.
<i>Dump</i> =	Contents of CCscp_sm_info.
<i>Dump</i> =	Contents of rlGSM_NET tuple.

Table 2 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	There is no tuple in relation RL GSM_NET whose key (rlGSM_NET.gsm) equals the current switching module (SM) number (GI_pcrd), but the fields in CCscp_sm_info are not their null values.
Possible Error Effect:	SCCP may have network routing errors and/or accounting errors.
Corrective Action Taken:	The audit sets all the fields in CCscp_sm_info to their default values by calling the macro CCDEFSCP_INFO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	Contents of CCscp_sm_info.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	The structure CCscp_sm_info is not verified for this iteration of the audit.
Corrective Action Taken:	The audit tries again during its next iteration.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

30. SD - SZ

SDAPSDL

Software Release: 5E10 and later

1. SPEED DIAL AUTOPROVISIONING SPEED DIAL LIST (SDAPSDL) AUDIT

The speed dialing autoprovisioning feature allows holders of calling cards to use their own personal speed dial codes to make card billed calls more conveniently by using an abbreviated speed dial code instead of a full forward number to specify the destination of a call. This feature provides an interactive, menu-driven system which allows card holders to add, delete, change, or listen to the speed codes associated with their calling cards.

The SDAPSDL audit is responsible for ensuring the integrity of tuples in the RLSDAP_SDL relation.

2. DATA STRUCTURES

The RLSDAP_SDL relation is a dynamic relation that contains a *sd/* field to store the speed dial list. The relation is keyed by the *sd/* field.

The RLSDAP_SDL relation contains a *busy* field which is used to denote the status of the tuple. If idle, the rISDAP_SDL tuple will be linked to the rIDB_HEADC tuple. If busy, the rISDAP_SDL tuple will be linked to the rIPCBLA tuple.

The *sdlmem* field contains the set member linkage.

The *quarstate* field represents the quarantine status for auditing purposes.

3. AUDIT CHECKS

The SDAPSDL audit will perform several different checks in the SM environment. The SDAPSDL audit uses the DRINASEK driver to traverse through all tuples in the RLSDAP_SDL relation.

If the SDAPSDL audit is entered, the following checks are performed for each tuple in the RLSDAP_SDL relation:

NOTIDLE	= Check to make sure that an rISDAPSDL tuple that is idle is linked to the rIDB_HEADC tuple.
IDLE_LINK	= Check to make sure the forward and backward tuple linkages are correct in an idle rISDAPSDL tuple.
IDLE_BUSY	= Check to make sure that in an idle rISDAPSDL tuple, the next rISDAPSDL tuple pointed to by the forward linked field is idle.
BDLINK	= Check to make sure the forward and backward tuple linkages are correct in a busy rISDAPSDL tuple having the same value in the owner field as an idle tuple.
BUSYIDLE	= Check to make sure that in a busy rISDAPSDL tuple, having the same value in the owner field as an idle tuple, that the next rISDAPSDL tuple pointed to by the forward linked field is busy.
BADLINK	= Check to make sure that in a busy rISDAPSDL tuple, having the same value in the owner field as an idle tuple, can have at a maximum 3 tuples linked to the rIPCBLA tuple.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The SDAPSDL audit has detected that a busy rSDAPSDL tuple is in an erroneous state where more than 3 tuples are linked to the rIPCBLA tuple.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rSDAPSDL tuples.
Corrective Action Taken:	The tuple under audit will be cleared and the key will be re-calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the next forward rSDAPSDL tuple.
<i>Bad Data =</i>	The <i>st_fork</i> field of the next forward rSDAPSDL tuple.
<i>Good Data =</i>	NA
<i>Logical Index =</i>	The logical key of the next forward rSDAPSDL tuple.
<i>Dump =</i>	The next forward rSDAPSDL tuple.

Table 2 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	A busy rSDAPSDL tuple is not linked correctly to a rIPCBLA tuple.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rSDAPSDL tuples.
Corrective Action Taken:	The tuple under audit will be cleared and the key will be re-calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rSDAPSDL tuple.
<i>Bad Data =</i>	The logical key of the rSDAPSDL tuple (<i>sdl_key</i>) indicates a linkage problem.
<i>Good Data =</i>	The <i>busy</i> field of the rSDAPSDL tuple.
<i>Logical Index =</i>	The logical key of the rSDAPSDL tuple (<i>sdl_key</i>).
<i>Dump 1 =</i>	The rSDAPSDL tuple.
<i>Dump 2 =</i>	The rIPCBLA tuple.

Table 3 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	A busy rSDAPSDL tuple is linked to an idle rSDAPSDL tuple.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rSDAPSDL tuples.
Corrective Action Taken:	The tuple under audit will be cleared and the key will be re-calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the next forward rSDAPSDL tuple.
<i>Bad Data =</i>	The <i>busy</i> field of the next forward rSDAPSDL tuple.
<i>Good Data =</i>	The <i>busy</i> field of the current rSDAPSDL tuple is DBYES.
<i>Logical Index =</i>	The logical key of the rSDAPSDL tuple.
<i>Dump 1 =</i>	The rSDAPSDL tuple.
<i>Dump 2 =</i>	The next forward rSDAPSDL tuple.

Table 4 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The SDAPSDL audit has detected that an idle rSDAPSDL tuple whose forward linkage field <i>st_fork</i> points to an rSDAPSDL tuple whose <i>busy</i> field indicates it is busy.
Possible Error Effect:	This resource is incorrectly marked and it can not be determined if the tuple should be marked idle or busy. This may cause possible call failures if the system tries to select this as an available rSDAPSDL tuple.
Corrective Action Taken:	The tuple under audit will be cleared and the key will be re-calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the next forward rSDAPSDL tuple.
<i>Bad Data =</i>	The <i>busy</i> field of the next forward rSDAPSDL tuple.
<i>Good Data =</i>	DBNO
<i>Logical Index =</i>	The logical key of the rSDAPSDL tuple (<i>sdl_key</i>).

<i>Dump 1</i> =	The rSDAPSDL tuple.
<i>Dump 2</i> =	The next forward rSDAPSDL tuple.

Table 5 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The SDAPSDL audit has detected that an idle rSDAPSDL tuple has a linkage problem in the <i>st_fork</i> or <i>st_back</i> linkage fields.
Possible Error Effect:	This resource is inaccessible for any calls. This may cause possible call failures if the system exhausts the number of available rSDAPSDL tuples.
Corrective Action Taken:	The tuple under audit will be cleared and the key will be re-calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rSDAPSDL tuple.
<i>Bad Data</i> =	The logical key of this rSDAPSDL tuple (<i>sdl_key</i>) indicates a linkage problem.
<i>Good Data</i> =	The <i>busy</i> field of the rSDAPSDL tuple.
<i>Logical Index</i> =	The logical key of the rSDAPSDL tuple (<i>sdl_key</i>).
<i>Dump 1</i> =	The rSDAPSDL tuple.
<i>Dump 2</i> =	The rIDB_HEADC tuple.

Table 6 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	An rSDAPSDL tuple whose <i>busy</i> field indicates it is an idle tuple but the <i>st_ownk</i> field indicates it is not linked to the rIDB_HEADC tuple.
Possible Error Effect:	This may indicate an rSDAPSDL tuple that was not released after the SDAP call was processed. This may lead to the possible loss of the rSDAPSDL tuple resource.
Corrective Action Taken:	The tuple under audit will be cleared and the key will be re-calculated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the corresponding rSDAPSDL tuple.
<i>Bad Data</i> =	The key of the owner that is linked to this rSDAPSDL tuple.
<i>Good Data</i> =	The <i>busy</i> field of the rSDAPSDL tuple.
<i>Logical Index</i> =	The logical key of the rSDAPSDL tuple. (<i>sdl_key</i>)
<i>Dump</i> =	The rSDAPSDL tuple.

SFG

Software Release: 5E10 and later

1. SIMULATED FACILITY GROUP (SFG) AUDIT

In private networks (PBX- and centrex-type systems), customers lease dedicated trunks from local offices to connect their facilities to the local office and among local offices. An alternative to leasing dedicated trunks is available with the simulated facility group feature of business and residence custom services-II (BRCS-II). Simulated facility groups give customers the equivalent of dedicated trunks. They also impose the burden of limiting traffic on the trunks to the local telephone office; this feature is primarily for centrex lines, which traditionally have no control over the number of calls being placed because the phones are connected directly to the local office. The simulated facility group simulates PBX control so that a group of customers can not access all the switch's resources. Another aspect of this control enables customers to keep trunks open for six-way and other conference calls. The simulated facility group audit (SFG) ensures that the data needed to maintain the simulated facility groups is consistent.

The SFG audit is part of the routing and terminal allocation audits subgroup. It runs in the communications module processor (CMP) in the initialization and segmented modes.

2. DATA STRUCTURES

The SFG audit checks the three relations depicted in Figure 1. The simulated facilities status relation (RLSFSTAT), a dynamic relation in the CMP, contains status information on the current simulated facility groups. The large simulated facilities status relation (RLLGSFTSTAT), a dynamic relation in the CMP, contains status information on the large (more than 96 trunks) simulated facility groups. The simulated facility group relation (RLRT_SFG), a static relation in the CMP, contains a group number field (the key), a group size field, group type field (large or small), and hunt type information for all the simulated facility groups. Note that the simulated facility groups do not correspond to any actual hardware (as does the trunk group relation, RLRT_TRKG); a routing table shows which actual hardware the simulated facility (trunk) uses.

The audit accesses these relations by their key (trunk group number) fields; only nonstandard linkages exist among them.

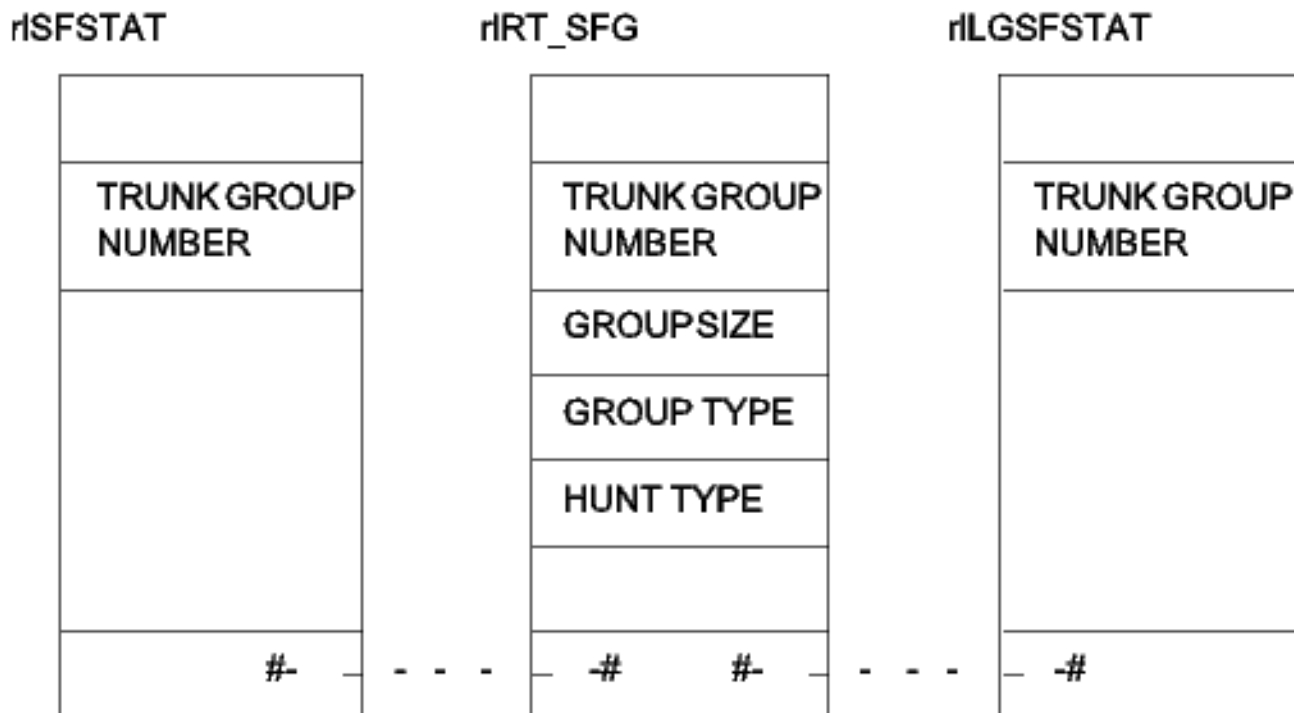


Figure 1 (SFG-1) Data Structures of the SFG Audit

3. AUDIT CHECKS

The SFG audit calls the static driver (DRSTATIC) to make the following checks:

- The audit makes a static-to-dynamic existence check between the static RLRT_SFG and the dynamic RLSFSTAT and RLLGSFSTAT relations. Large groups that exist in rLRT_SFG should exist in RLLGSFSTAT; small groups that exist in rLRT_SFG should exist in RLSFSTAT.
- The audit checks the key of the dynamic tuple to ensure that it matches the group number in RLRT_SFG.
- The audit checks the hunt type in RLRT_SFG. If the hunt type is other than large or small, an error is reported.

4. ERROR HANDLING

If the static-to-dynamic existence check fails, the audit adds the missing tuples to RLSFSTAT and RLLGSFSTAT. If either of the semantic checks fails, the audit prints a manual action error to ensure that the local office corrects the static RLRT_SFG.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The simulated-facility hunt group information relation contains a tuple which has an invalid hunt type. This error will repeat until fixed by office personnel.
Possible Error Effect:	Inability to hunt to facilities in this group and possible assert failures.
Corrective Action Taken:	None.
Manual Action Required:	Determine the nature of the data base inconsistency and correct it. This message will repeat until corrective action is taken by the end user. Use RC/V or ODBE to review and correct the data for this simulated facility group.
Dump Description:	
<i>Error Address =</i>	Address of the rIRT_SF group tuple buffer.
<i>Bad Data =</i>	Hunt type of the group.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Group number of the SF group.
<i>Dump =</i>	rLGSFSTAT tuple.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	A rSFSTAT tuple has bad data in the key (grpnum) field for a SF group.
Possible Error Effect:	Inability to terminate to facilities in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSFSTAT tuple.
<i>Bad Data =</i>	Contents of the grpnum field in the rSFSTAT tuple.
<i>Good Data =</i>	Group number of the SF group.
<i>Logical Index =</i>	Group number of the SF group.
<i>Dump =</i>	rSFSTAT tuple.

Table 3 Error Code: LGKEY

Error Code:	LGKEY
Error Description:	A rLGSFSTAT tuple has bad data in the key (grpnum) field for a SF group.
Possible Error Effect:	Inability to terminate to facilities in this group and possible assert failures.
Corrective Action Taken:	The key is restored to its correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLGSFSTAT tuple.
<i>Bad Data =</i>	Contents of the grpnum field in the rLGSFSTAT tuple.
<i>Good Data =</i>	Group number of the SF group.
<i>Logical Index =</i>	Group number of the SF group.
<i>Dump =</i>	NA.

Table 4 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	There is a tuple missing in the RLLGSFSTAT relation for a large simulated facility group.
Possible Error Effect:	Inability to terminate to facilities in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Group number of the SF group.
<i>Dump =</i>	NA.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the RLSFSTAT relation for a simulated facility group.
Possible Error Effect:	Inability to terminate to facilities in this group and possible assert failures.
Corrective Action Taken:	The tuple is created and initialized.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number of the SF group
<i>Dump</i> =	NA.

SFGCOR-A

Software Release: 5E10 - 5E11

1. SIMULATED FACILITY GROUP CALL INFORMATION STATUS (SFG CORRAL) RELATION (SFGCOR) AUDIT

Simulated facility groups (SFGs) restrict the number of incoming or outgoing calls. In this way, end customers may throttle telephone usage, and operating companies can obtain various usage measurements on a per group basis. The size of a SFG is customer-specified. Unlimited SFGs provide no throttling, but do provide all of the measurements data that conventional SFGs do.

All SFG call data in the switching modules (SMs) is consolidated into one dynamic data area called the SFG CORRAL. The SFGCOR audit verifies the live simulated facility (SF) call data contained in this SFG CORRAL relation (RLSFGCOR). This relation exists in all SMs.

The SFGCOR audit runs in routine and elevated segmented modes.

2. DATA STRUCTURES

The SFGCOR audit provides checks on rLSFGCOR tuples to determine if the tuples are associated with live (active) SFG calls. The audit provides these checks based on the following fields:

<i>crnum</i>	= The key of the rLSFGCOR tuple. For non 6-port SFG tuple, this value should correspond to the key of an associated rICRECORD (call record) tuple which is validly busy. For SFGs associated with 6-port conference circuits, there is no associated call record. For these SFGs, the rLSFGCOR tuple key will be set to a value at or above the maximum call record (DBQNCRECORD) but less than the maximum key value allowed in the RLSFGCOR relation (SFGCORMAX).
<i>sfg[]</i>	= Contains the group and member data for up to three SFG members.
<i>sfgtype[]</i>	= Contains the SFG type for up to three SFG members.
<i>pid</i>	= For 6-port SFGs, this field will contain the 6-port conference circuit Keep Alive terminal process ID. This process ID is used by call processing applications to periodically check that the 6-port terminal process on the originating SM is active. For ICR SFGs, this field is used to store the ICR tracking SM number. For other types of SFG calls, this field is not used.
<i>wsf_head</i>	= For wide band SFG calls, this field provides a semantic linkage to the head of a list of wide band SFG corral (RTwsfcorral[]) array entries.
<i>wsf_tail</i>	= For wide band SFG calls, this field provides a semantic linkage to the tail of a list of wide band SFG corral (RTwsfcorral[]) array entries. For wide band lists containing one entry, this field is set to WSFSAMEIDX.

The rLSFGCOR tuple and related data structures are shown in Figure 1. Figure 2 shows the wide band SFG scenario. Figure 3 shows an example of list with single entry.

3. AUDIT CHECKS

For each tuple in the RLSFGCOR relation, the SFGCOR audit makes the following checks: If the SFGCOR audit gets entered:

INV_IDLE	= Check the status of the SFG member data contained in the tuple to verify that it is marked non-zero.

GRPTYP_NUM	= Range check the values stored in the group and member data.
OUTOFRANGE	= Verify that the SFG type is valid for the particular Call Record associated with the SFG member.
NOPCBLA	= Check that there is a rIPCBLA tuple associated with a 6-port SFG call.
CALL_ID	= Check the <i>call_id.crindx</i> in RLPCBLA against the <i>crnum</i> (key) in RLSFGCOR.
PID_6PORT	= Verify that a 6-port SFG call has a corresponding live 6-port process.
CCINDEX	= Verify that a 6-port SFG call has the appropriate bit marked in the RTccsfg[] array.
TRACK_SM	= Verify that the tracking SM number is valid for an ICR SFG call.
NO_ICR	= Verify that there is a corresponding rLICR static tuple for an ICR SFG call.
SFG_CRECORD	= Verify that there is a call record for a SFG call.
STATUS	= Verify that the <i>sfg_call</i> bit in a corresponding call record (rICRECORD) tuple indicates SFG.
INV_STATE	= Verify that the SFG type for a call is validly marked.
UNPACKED	= Verify that all of the SFG group and member data is packed in the most forward locations of the <i>sfg[]</i> array.
If this is a wide band SFG call:	
INDEX	= Verify that the values marked in the <i>wsf_head</i> and <i>wsf_tail</i> linkages are within a valid range.
HTYPE	= Verify that the SFG type is valid for wide band calls.
GRPINFO	= Verify that the group and member data is validly populated for wide band SFG calls.
LIST	= Verify the integrity of the list of wide band (RTwsfcorral[]) array entries.
STATE_SET	= Verify that every marked bit in the RTccsfg[] 6-port SFG call array maps to a corresponding tuple in the RLSFGCOR relation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: CALL_ID

Error Code:	CALL_ID
Error Description:	While auditing a 6-port SFG call, the SFGCOR audit was able to access a rIPCBLA tuple, however, the call record ID in the rIPCBLA tuple did not match the key of the rISFGCOR tuple containing the 6-port SFG call. For 6-port calls, a rIPCBLA tuple should exist such that rIPCBLA.procnum == rISFGCOR.pid.procno, AND rIPCBLA.call_id.crindx == (key) rISFGCOR.crnum,

	AND rISFGCOR.sfgtype[0] == SFG6PORT.
Possible Error Effect:	The 6-port SFG call(s) associated with this entry do not have a valid call ID, and may become stuck busy, since call processing applications will not be able to access them.
Corrective Action Taken:	The rISFGCOR tuple, associated with the 6-port SFG call, which does not match the rIPCBLA call ID, is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the rISFGCOR tuple.
<i>Bad Data =</i>	The call ID found in the rIPCBLA tuple (rIPCBLA.call_id.crindx).
<i>Good Data =</i>	The key of the rISFGCOR tuple (rISFGCOR.crnnum).
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.crnnum).
<i>Dump 1 =</i>	The rISFGCOR tuple where the error was found.
<i>Dump 2 =</i>	The rIPCBLA tuple where the call ID was found.

Table 2 Error Code: CCINDX

Error Code:	CCINDX
Error Description:	A tuple in the RLSFGCOR relation was marked as being used for 6-port conference circuit SFG calls (rISFGCOR.sfgtype[] == SFG6PORT), however the associated bit marked in the RTccsfg[] 6-port conference circuit SFG global array indicated that a 6-port SFG call was inactive. For every tuple marked in RLSFGCOR as being used for 6-port, an associated bit must be marked (set to 0) in the RTccsfg[] array indicating that the 6-port SFG conference circuit feature is active, such that rISFGCOR.crnnum(key) == (DBQNCRECOR + bit position in RTccsfg[]).
Possible Error Effect:	The 6-port SFG call(s) associated with this entry do not have a valid call ID, or tuple in RLSFGCOR and may become stuck busy, since call processing applications will not be able to access them.
Corrective Action Taken:	The appropriate bit in the RTccsfg[] array will be marked to an idle condition (set to 1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple marked for 6-port.
<i>Bad Data =</i>	The incorrect state of the bit found in RTccsfg[] (DBYES).
<i>Good Data =</i>	The correct state for the bit (DBNO).
<i>Logical Index =</i>	The corresponding key of the rISFGCOR tuple which was marked for 6-port (bit position in RTccsfg[] + DBQNCRECOR).
<i>Dump 1 =</i>	The contents of the rISFGCOR tuple.
<i>Dump 2 =</i>	The entire contents of the RTccsfg[] conference circuit SFG array.

Table 3 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	For a wide band SFG call, the number of SFG group/member combinations marked in the rISFGCOR tuple is invalid. For rISFGCOR tuple associated with wide band calls (<i>wsf_tail</i> marked to a non-zero value), a maximum of (MAXSF-1) group/member combinations are allowed.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s), and hung (stuck busy) SFG members in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and any associated wide band list of RTwsfcorral[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The contents of the SFG hunt type data <i>sfgtype[]</i> from the rISFGCOR tuple under audit ((sfgtype[0] <<8) sfgtype[1]).
<i>Good Data =</i>	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit ((wsf_head << 16) wsf_tail).
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.crnnum).
<i>Dump =</i>	The contents of the rISFGCOR tuple.

Table 4 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The group number for this SFG call is not in the legal range of allowable SFG group numbers.
Possible Error Effect:	Asserts from call processing applications, and eventual illegal segment table entry errors and hung SFG members in the CMP.
Corrective Action Taken:	The SFG corral entry position for this SFG call (rISFGCOR.sfg[position]) will be zeroed out. If this is the last marked position in the tuple, the tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFGCOR tuple.
<i>Bad Data</i> =	The SFG group number found.
<i>Good Data</i> =	The SFG group member number found.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple.

Table 5 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SFG group type <i>sfgtype[]</i> marked in the rISFGCOR tuple is invalid for wide band. For wide band SFG calls, the SFG group type must be either SFGEDSL, SFG2WESSX, SFGOUTESSX, or SFGINESSX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and any associated wide band list of RTwscorral[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The position in the list of the RTwscorral[] array entry that was found invalid.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the rISFGCOR tuple under audit (<i>sfgtype[0]</i> <<8) <i>sfgtype[1]</i>).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple.

Table 6 Error Code: INDEX

Error Code:	INDEX
Error Description:	One or both of the wide band array linkages in the rISFGCOR tuple were found to be out of range. The <i>wsf_head</i> and <i>wsf_tail</i> linkages should contain valid values between zero and (WSFMAXCOR - 1) which is the maximum allowable entry in RTwscorral[] or, for the case where the RTwscorral[] list contains only one entry, <i>wsf_tail</i> should be marked WSFSAMEIDX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and the associated wide band list of RTwscorral[] entries will be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The position in the list of the RTwscorral[] array entry that was found invalid.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the rISFGCOR tuple under audit (<i>sfgtype[0]</i> <<8) <i>sfgtype[1]</i>).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple under audit.

Table 7 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	A rISFGCOR tuple was found with no active SFG group and member data contained in it. All rISFGCOR tuples must have at least one SFG group and member marked in them (rISFGCOR.sfg[].sfg_grp != 0).
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	A flag indicating no SFG data could be found by the audit (AUFALSE).
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump =</i>	The contents of the rISFGCOR tuple under audit.

Table 8 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An SFG call in the RLSFGCOR relation was found to have an unknown or invalid SFG type. The SFG type indicates the particular feature or service which is using the SFG throttling mechanism. There are two major reasons why this error may occur: (1) A new feature has very recently been added which uses the SFG throttling mechanism, and thus, a new SFG type has also been added to identify that this feature is using SFGs. The audit, however, could not identify the new type associated with this SFG call among those it has knowledge of. (2) The SFG type field in the RLSFGCOR relation which is associated with this SFG call has become corrupted.
Possible Error Effect:	Call processing applications will not be able to release this call since it cannot determine the SFG type. This will result in hung SFG data in the CMP.
Corrective Action Taken:	The call with the invalid SFG type will be zeroed out in the appropriate rISFGCOR tuple and any other VALID SF calls in the entry will be silently repacked to their most forward locations in this entry. If, after the invalid call information has been deleted, there is no remaining SFG call data in the tuple, the tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple which contained which contained the bad SFG type.
<i>Bad Data =</i>	The SFG type that was found associated with this SF call (rISFGCOR.sfgtype[]).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The address of the rISFGCOR tuple.
<i>Dump =</i>	The contents of the rISFGCOR tuple.

Table 9 Error Code: LIST

Error Code:	LIST
Error Description:	The list of RTwsfcorral[] wide band array entries linked to the rISFGCOR tuple under audit was invalid. For every rISFGCOR tuple marked for wide band (rISFGCOR.wsf_tail != 0) a list of one to AUWBLISTSZ RTwsfcorral[] wide band array entries should be linked off of the rISFGCOR tuple.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG member data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and any associated wide band list of RTwsfcorral[] entries will be quarantined. These entries will then be recovered

	by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	For cases where the head entry of the list cannot be accessed (error type is marked AUWHEAD) or cases where the list size was out of range (error type is AUWLSIZE), the value dumped will be AUWBLISTSZ, indicating the list itself could not be accessed and audited from the rISFGCOR tuple. For all other cases, the entry position in the RTwsfcorral[] wide band array list between zero and AUWBLISTSZ where the error was found will be dumped.
<i>Bad Data =</i>	The contents of the <i>prev_idx</i> and <i>next_idx</i> linkages in the RTwsfcorral[] wide band array entry which was found to be corrupt (<i>prev_idx</i> << 16) <i>next_idx</i>).
<i>Good Data =</i>	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index =</i>	The key of the rISFGCOR tuple under audit (rISFGCOR.cnum) and the error type ((key of tuple under audit << 16) error type). The error type indicates the specific problem that was found with the RTwsfcorral[] wide band array list. The error types are: AUWHEAD = The head entry of the RTwsfcorral[] wide band array list could not be accessed or was not found linked to the rISFGCOR tuple. For RTwsfcorral[rISFGCOR.wsf_head].cid, rISFGCOR.cnum must be equal to <i>cid</i> in RTwsfcorral[]. AUWTAIL = The tail entry of the RTwsfcorral[] wide band array was not found linked to the rISFGCOR tuple. For RTwsfcorral[rISFGCOR.wsf_tail].cid, rISFGCOR.cnum must equal <i>cid</i> in RTwsfcorral[] for cases where the wide band list contains more than one entry. AUWNRANGE = A RTwsfcorral[] list entry was found which is not the tail of the list, but with a <i>next_idx</i> field (index to the next entry on the list) out of range. This field must be zero or greater, but less than WSFMAXCOR. AUWPLINK = A RTwsfcorral[] list entry did not point back to its predecessor (<i>prev_idx</i> was invalid). AUWLSIZE = The RTwsfcorral[] wide band array list was found to contain too many array entries for this wide band call. The list must contain one to AUWBLISTSZ entries.
<i>Dump =</i>	The contents of the rISFGCOR tuple under audit.

Table 10 Error Code: NO_ICR

Error Code:	NO_ICR
Error Description:	The corresponding rICR static tuple for an ICR SFG call could not be found. For ICR SFG calls, a static rICR tuple should exist, such that rICR.sfg_num == rISFGCOR.sfg[].sfg_grp, AND rISFGCOR.sfgtype[] == SFGICR.
Possible Error Effect:	An ICR SFG call (member) is associated with a simulated facility group (SFG) which does not exist. This is an illegal condition, and the call is stuck busy, since its corresponding data cannot be properly accessed. This will result in asserts out of call processing data, and hung SFG data in the ICR tracking SM.
Corrective Action Taken:	The ICR SFG data for the call is zeroed out from the rISFGCOR tuple. If this is the last data populated in the tuple, then the tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the rISFGCOR tuple.
<i>Bad Data =</i>	The SFG group number in RLSFGCOR associated with the ICR SFG (rISFGCOR.sfg[].sfg_grp.
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	The key of the rISFGCOR tuple where the error was found (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 11 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	The associated rIPCBLA tuple could not be found for a 6-port call located in a rISFGCOR tuple. For all 6-port related processes, there should be a corresponding rIPCBLA tuple with its key equal to the 6-port process number contained in the PID field of the rISFGCOR tuple, (where rISFGCOR.cnum ≥ DBQNCRECO, AND rISFGCOR.sfgtype[] == SFG6PORT AND rISFGCOR.pid.procno == rIPCBLA.procno).
Possible Error Effect:	Possible customer reorder treatment because the 6-port SF calls located in this entry are stuck busy.
Corrective Action Taken:	The rISSFGCOR tuple associated with the 6-port conference call is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the rISFGCOR tuple.
<i>Bad Data</i> =	The process number found in the rISFGCOR tuple for which there was no corresponding rIPCBLA tuple (rISFGCOR.pid.procno).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 12 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The SFG call information found in this entry was found in illegal location in the RLSFGCOR relation. Non-6-port type SFG calls belong in the range indexed below the maximum call record (rISFGCOR.cnum < CRMAX and rISFGCOR.sfgtype[] != SFG6PORT). 6-port conference circuit SFG calls belong in the range indexed at or above the maximum configurable call record (rISFGCOR.cnum ≥ DBQNCRECOR < SFGCORMAX AND rISFGCOR.sfgtype[] == SFG6PORT).
Possible Error Effect:	Lost resources since the SFG call data cannot be correctly accessed. Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the rISFGCOR tuple.
<i>Bad Data</i> =	The SFG type found at the incorrect location (rISFGCOR.sfgtype[]).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 13 Error Code: PID_6PORT

Error Code:	PID_6PORT
Error Description:	The 6-port SF call(s) in a rISFGCOR tuple were found to be associated with a process which is not alive. For 6-port calls, a live process ID must be marked in rISFGCOR.pid.
Possible Error Effect:	The 6-port SF call(s) in this rISFGCOR tuple are stuck busy, and the member data in the CMP is in a hung state.
Corrective Action Taken:	The tuple found in error is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the rISFGCOR tuple.
<i>Bad Data</i> =	The process number of the dead 6-port process (rISFGCOR.pid.procno).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 14 Error Code: SFG_CRECORD

Error Code:	SFG_CRECORD
Error Description:	SFG call(s) in a rISFGCOR tuple in the non-6-port range (rISFGCOR.cnum < CRMAX) of the RLSFGCOR relation did not have a corresponding call record tuple in use. For all rISFGCOR tuples not associated with 6-port calls, there should be a valid call record for the tuple (where rICRECORD.cnum == rISFGCOR.cnum).
Possible Error Effect:	All of the SFG calls in this entry are stuck busy, and the associated data in the CMP is hung. Possible customer reorder treatment may result.
Corrective Action Taken:	The rISFGCOR tuple where the error was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the rISFGCOR tuple.
<i>Bad Data =</i>	The key of the associated rICRECORD tuple which could not be accessed (rICRECORD.cnum).
<i>Good Data =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump =</i>	The contents of the rISFGCOR tuple where the error was found.

Table 15 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	A bit was marked in the RTccsfg[] 6-port conference circuit SFG global array indicating that a 6-port SFG call is active, however no corresponding tuple in the RLSFGCOR relation could be found. For every bit marked in the RTccsfg[] array (set to 0), there should exist a corresponding tuple in the RLSFGCOR relation, such that rISFGCOR.cnum(key) == (DBQNCRECOR + bit position in RTccsfg[]).
Possible Error Effect:	The 6-port SFG call(s) associated with this entry do not have a valid call ID, or tuple in RLSFGCOR and may become stuck busy, since call processing applications will not be able to access them.
Corrective Action Taken:	The appropriate bit in the RTccsfg[] array will be marked to an idle condition (set to 1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the RTccsfg[] global array.
<i>Bad Data =</i>	The bit position in the array which was found invalidly marked.
<i>Good Data =</i>	The correct state for the bit (DBYES).
<i>Logical Index =</i>	The corresponding key of the unpopulated rISFGCOR tuple which the bit mapped to (DBQNCRECOR + bit).
<i>Dump =</i>	The entire contents of the RTccsfg[] array.

Table 16 Error Code: STATUS

Error Code:	STATUS
Error Description:	SFG call(s) in a rISFGCOR tuple in the non-6-port range (rISFGCOR.cnum < CRMAX) of the RLSFGCOR relation did not have a corresponding call record tuple which was marked busy and active for SFG calls. For all rISFGCOR tuples not associated with 6-port calls, there should be a valid call record which is marked busy for SFG calls associated with the rISFGCOR tuple (where rICRECORD.cnum == rISFGCOR.cnum, rICRECORD.crlink.st_ownk - DBSFDB_HEA == DBCRECBUSY, AND rICRECORD.sfg_call == DBYES).
Possible Error Effect:	All of the SFG calls in this entry are stuck busy, and the associated data in the CMP is hung. Possible customer reorder treatment may result.
Corrective Action Taken:	The rISFGCOR tuple where the error was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the rISFGCOR tuple.
<i>Bad Data =</i>	The owner head cell list that the rICRECORD tuple is linked to (rICRECORD.crlink.st_ownk - DBSFDB_HEA).

<i>Good Data</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump 1</i> =	The contents of the rISFGCOR tuple where the error was found.
<i>Dump 2</i> =	The contents of the associated call record tuple where rICRECORD.cnum == rISFGCOR.cnum.

Table 17 Error Code: TRACK_SM

Error Code:	TRACK_SM
Error Description:	The tracking SM number associated with an ICR SFG call did not match the tracking SM number stored in the static ICR data relation (RLICR). For all rISFGCOR tuples used for ICR calls (rISFGCOR.sfgtype[] == SFGICR), the tracking SM marked in the rISFGCOR tuple should match the tracking SM marked in the rLICR static tuple (where rISFGCOR.pid.pcrd == rLICR.track_sm for tuples where rISFGCOR.sfg[].sfg_grp == rLICR.sfg_num).
Possible Error Effect:	Lost resources, and asserts from call processing applications. Hung SFG members on the ICR tracking SM.
Corrective Action Taken:	The tracking SM number associated with the ICR SFG call is corrected to match the one found in static data (rLICR.track_sm).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the rISFGCOR tuple.
<i>Bad Data</i> =	The tracking SM number found in the rISFGCOR tuple (rISFGCOR.pid.pcrd).
<i>Good Data</i> =	The tracking SM number found in static data (rLICR.track_sm).
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 18 Error Code: UNPACKED

Error Code:	UNPACKED
Error Description:	SFG call information was not loaded properly into a tuple in the RLSFGCOR relation. The SFG call data containing SFG group, member number, and type must be "packed" into the most forward (lowest indexed) locations of the <i>sfg[]</i> and <i>sfgtype[]</i> arrays in each rISFGCOR tuple.
Possible Error Effect:	Invalid operation out of call processing applications. Stuck SFG member data, and hung SFG data in the CMP.
Corrective Action Taken:	The SFG data is repacked to its most forward locations. Note: If the inconsistency exists because of earlier audit recovery, the audit recovers (repacks) the data without firing an error. In any other instance, this error will be fired.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the rISFGCOR tuple.
<i>Bad Data</i> =	The empty position in the entry (rISFGCOR.sfg[position]) which preceded a non-empty position.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISFGCOR tuple where the error was found.
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

SFGCOR-B

Software Release: 5E12 and later

1. SIMULATED FACILITY GROUP CALL INFORMATION STATUS (SFG CORRAL) RELATION (SFGCOR) AUDIT

Simulated facility groups (SFGs) restrict the number of incoming or outgoing calls. In this way, end customers may throttle telephone usage, and operating companies can obtain various usage measurements on a per group basis. The size of a SFG is customer-specified. Unlimited SFGs provide no throttling, but do provide all of the measurements data that conventional SFGs do.

All SFG call data in the switching modules (SMs) is consolidated into one dynamic data area called the SFG CORRAL. The SFGCOR audit verifies the live simulated facility (SF) call data contained in this SFG CORRAL relation (RLSFGCOR). This relation exists in all SMs.

The SFGCOR audit runs in routine and elevated segmented modes.

2. DATA STRUCTURES

The SFGCOR audit provides checks on rLSFGCOR tuples to determine if the tuples are associated with live (active) SFG calls. The audit provides these checks based on the following fields:

<i>crnum</i>	= The key of the rLSFGCOR tuple. For non 6-port SFG tuple, this value should correspond to the key of an associated rICRECORD (call record) tuple which is validly busy. For SFGs associated with 6-port conference circuits, there is no associated call record. For these SFGs, the rLSFGCOR tuple key will be set to a value at or above the maximum call record (DBQNCRECORD) but less than the maximum key value allowed in the RLSFGCOR relation (SFGCORMAX).
<i>sfg[]</i>	= Contains the group and member data for up to three SFG members.
<i>sfgtype[]</i>	= Contains the SFG type for up to three SFG members.
<i>pid</i>	= For 6-port SFGs, this field will contain the 6-port conference circuit keep alive terminal process ID. This process ID is used by call processing applications to periodically check that the 6-port terminal process on the originating SM is active. For ICR SFGs, this field is used to store the ICR tracking SM number. For other types of SFG calls, this field is not used.
<i>wsf_head</i>	= For wide band SFG calls, this field provides a semantic linkage to the head of a list of wide band SFG corral (RTwsfcorral[]) array entries.
<i>wsf_tail</i>	= For wide band SFG calls, this field provides a semantic linkage to the tail of a list of wide band SFG corral (RTwsfcorral[]) array entries. For wide band lists containing one entry, this field is set to WSFSAMEIDX.

The rLSFGCOR tuple and related data structures are shown in Figure 1. Figure 2 shows the wide band SFG scenario. Figure 3 shows an example of list with single entry.

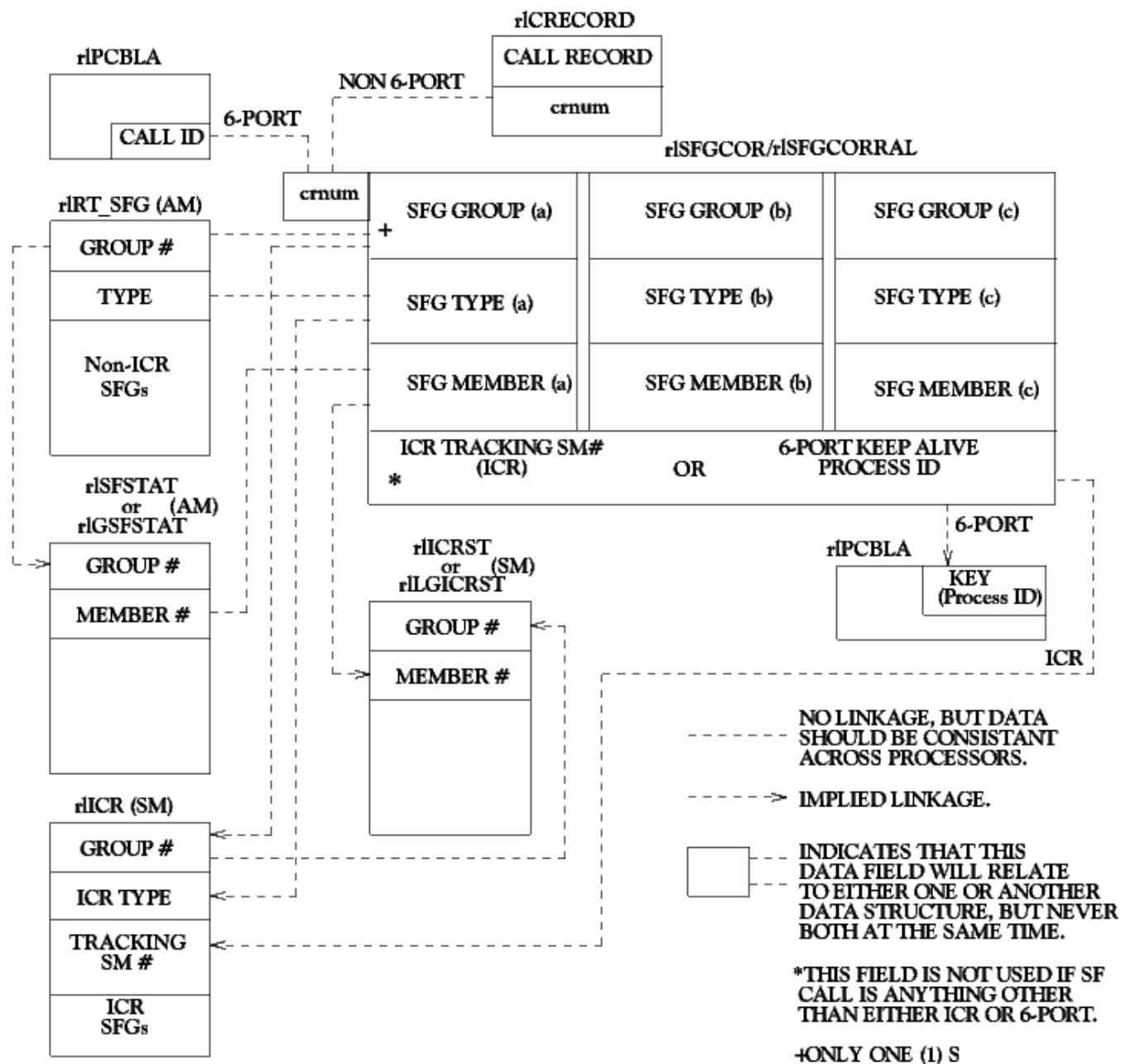


Figure 1 (SFGCOR-1) SFG CORRAL and Related Data Structures

rISFGCOR/rISFGCORRAL

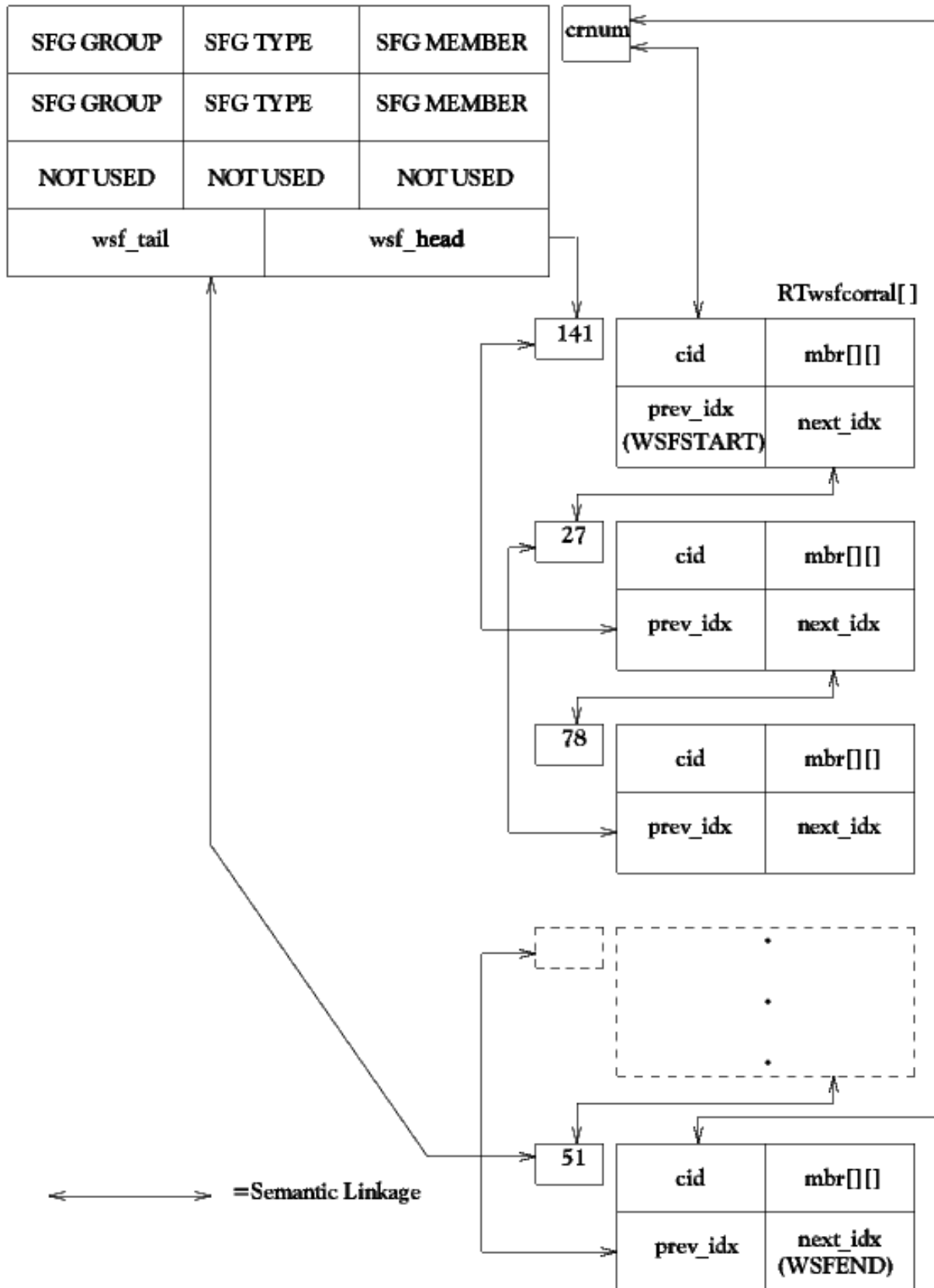
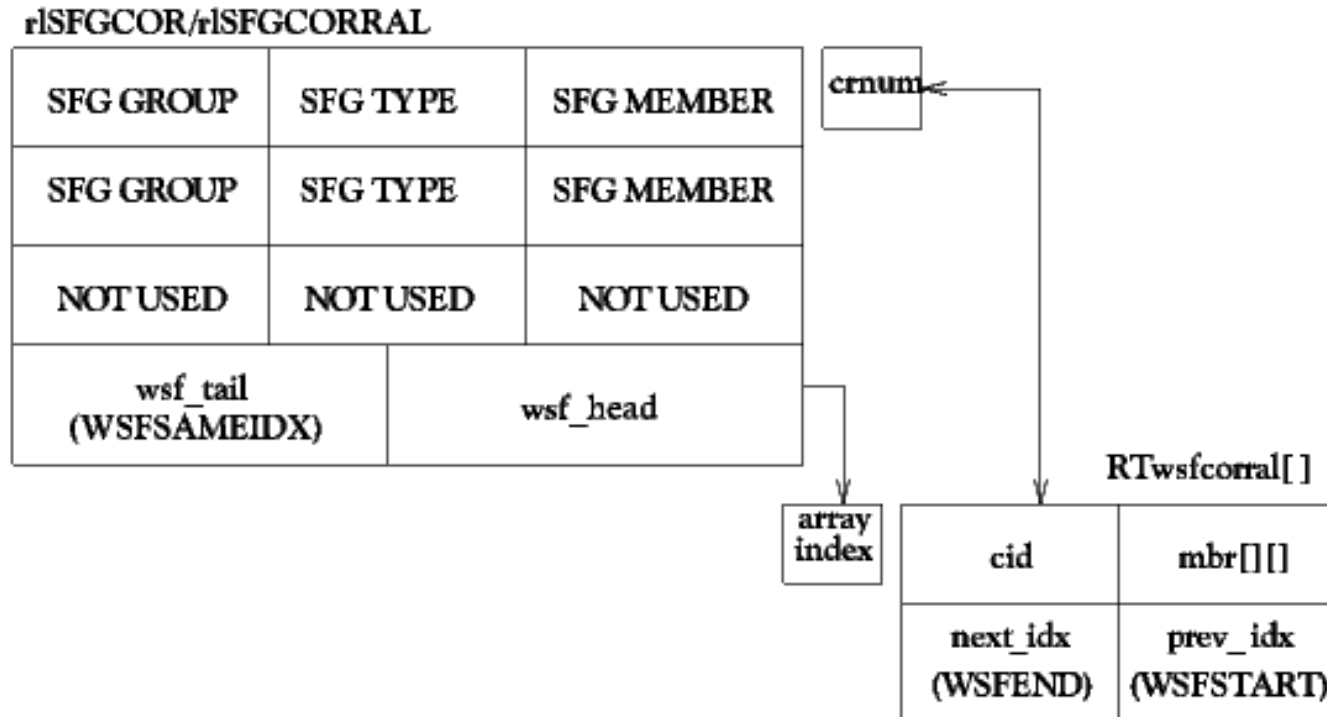


Figure 2 (SFGCOR-2) Wide Band SFG Scenario



↔ = Semantic Linkage

Figure 3 (SFGCOR-3) Example of list with single entry

3. AUDIT CHECKS

For each tuple in the RLSFGCOR relation, the SFGCOR audit makes the following checks:

If the SFGCOR audit gets entered:

- INV_IDLE = Check the status of the SFG member data contained in the tuple to verify that it is marked non-zero.
- GRPTYP_NUM = Range check the values stored in the group and member data.
- OUTOFRANGE = Verify that the SFG type is valid for the particular Call Record associated with the SFG member.
- NOPCBLA = Check that there is a rIPCBLA tuple associated with a 6-port SFG call.
- CALL_ID = Check the *call_id.crindx* in RLPCBLA against the *crnum* (key) in RLSFGCOR.
- PID_6PORT = Verify that a 6-port SFG call has a corresponding live 6-port process.
- CCINDEX = Verify that a 6-port SFG call has the appropriate bit marked in the RTccsfg[] array.
- TRACK_SM = Verify that the tracking SM number is valid for an ICR SFG call.

NO_ICR	= Verify that there is a corresponding rLICR static tuple for an ICR SFG call.
SFG_CRECORD	= Verify that there is a call record for a SFG call.
STATUS	= Verify that the <i>sfg_call</i> bit in a corresponding call record (rICRECORD) tuple indicates SFG.
INV_STATE	= Verify that the SFG type for a call is validly marked.
UNPACKED	= Verify that all of the SFG group and member data is packed in the most forward locations of the <i>sfg[]</i> array.
If this is a wide band SFG call:	
INDEX	= Verify that the values marked in the <i>wsf_head</i> and <i>wsf_tail</i> linkages are within a valid range.
HTYPE	= Verify that the SFG type is valid for wide band calls.
GRPINFO	= Verify that the group and member data is validly populated for wide band SFG calls.
LIST	= Verify the integrity of the list of wide band (RTwsfcorral[]) array entries.
STATE_SET	= Verify that every marked bit in the RTccsfg[] 6-port SFG call array maps to a corresponding tuple in the RLSFGCOR relation.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: CALL_ID

Error Code:	CALL_ID
Error Description:	While auditing a 6-port SFG call, the SFGCOR audit was able to access a rPCBLA tuple, however, the call record ID in the rPCBLA tuple did not match the key of the rSFGCOR tuple containing the 6-port SFG call. For 6-port calls, a rPCBLA tuple should exist such that rPCBLA.procnum == rSFGCOR.pid.procno, AND rPCBLA.call_id.cridx == (key) rSFGCOR.cnum, AND rSFGCOR.sfgtype[0] == SFG6PORT.
Possible Error Effect:	The 6-port SFG call(s) associated with this entry do not have a valid call ID, and may become stuck busy, since call processing applications will not be able to access them.
Corrective Action Taken:	The rSFGCOR tuple, associated with the 6-port SFG call, which does not match the rPCBLA call ID, is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rSFGCOR tuple.
<i>Bad Data</i> =	The call ID found in the rPCBLA tuple (rPCBLA.call_id.cridx).
<i>Good Data</i> =	The key of the rSFGCOR tuple (rSFGCOR.cnum).
<i>Logical Index</i> =	The key of the rSFGCOR tuple (rSFGCOR.cnum).
<i>Dump 1</i> =	The rSFGCOR tuple where the error was found.
<i>Dump 2</i> =	The rPCBLA tuple where the call ID was found.

Table 2 Error Code: CCINDX

Error Code:	CCINDEX
Error Description:	A tuple in the RLSFGCOR relation was marked as being used for 6-port conference circuit SFG calls (rlSFGCOR.sfgtype[] == SFG6PORT), however the associated bit marked in the RTccsfg[] 6-port conference circuit SFG global array indicated that a 6-port SFG call was inactive. For every tuple marked in RLSFGCOR as being used for 6-port, an associated bit must be marked (set to 0) in the RTccsfg[] array indicating that the 6-port SFG conference circuit feature is active, such that rlSFGCOR.cnum(key) == (DBQNCRECOR + bit position in RTccsfg[]).
Possible Error Effect:	The 6-port SFG call(s) associated with this entry do not have a valid call ID, or tuple in RLSFGCOR and may become stuck busy, since call processing applications will not be able to access them.
Corrective Action Taken:	The appropriate bit in the RTccsfg[] array will be marked to an idle condition (set to 1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rlSFGCOR tuple marked for 6-port.
<i>Bad Data =</i>	The incorrect state of the bit found in RTccsfg[] (DBYES).
<i>Good Data =</i>	The correct state for the bit (DBNO).
<i>Logical Index =</i>	The corresponding key of the rlSFGCOR tuple which was marked for 6-port (bit position in RTccsfg[] + DBQNCRECOR).
<i>Dump 1 =</i>	The contents of the rlSFGCOR tuple.
<i>Dump 2 =</i>	The entire contents of the RTccsfg[] conference circuit SFG array.

Table 3 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	For a wide band SFG call, the number of SFG group/member combinations marked in the rlSFGCOR tuple is invalid. For rlSFGCOR tuple associated with wide band calls (<i>wsf_tail</i> marked to a non-zero value), a maximum of (MAXSF-1) group/member combinations are allowed.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s), and hung (stuck busy) SFG members in the CMP.
Corrective Action Taken:	The rlSFGCOR tuple will be deleted, and any associated wide band list of RTwsfcorral[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The contents of the SFG hunt type data <i>sfgtype[]</i> from the rlSFGCOR tuple under audit ((<i>sfgtype[0]</i> <<8) <i>sfgtype[1]</i>).
<i>Good Data =</i>	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rlSFGCOR tuple under audit ((<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index =</i>	The key of the rlSFGCOR tuple (rlSFGCOR.cnum).
<i>Dump =</i>	The contents of the rlSFGCOR tuple.

Table 4 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The group number for this SFG call is not in the legal range of allowable SFG group numbers.
Possible Error Effect:	Asserts from call processing applications, and eventual illegal segment table entry errors and hung SFG members in the CMP.
Corrective Action Taken:	The SFG corral entry position for this SFG call (rlSFGCOR.sfg[position]) will be zeroed out. If this is the last marked position in the tuple, the tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rlSFGCOR tuple.
<i>Bad Data =</i>	The SFG group number found.
<i>Good Data =</i>	The SFG group member number found.

<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple.

Table 5 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SFG group type <i>sfgtype[]</i> marked in the rISFGCOR tuple is invalid for wide band. For wide band SFG calls, the SFG group type must be either SFGEDSL, SFG2WESSX, SFGOUTESSX, or SFGINESSX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and any associated wide band list of RTwsfcorral[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The position in the list of the RTwsfcorral[] array entry that was found invalid.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the rISFGCOR tuple under audit (<i>sfgtype[0]</i> <<8) <i>sfgtype[1]</i>).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple.

Table 6 Error Code: INDEX

Error Code:	INDEX
Error Description:	One or both of the wide band array linkages in the rISFGCOR tuple were found to be out of range. The <i>wsf_head</i> and <i>wsf_tail</i> linkages should contain valid values between zero and (WSFMAXCOR - 1) which is the maximum allowable entry in RTwsfcorral[] or, for the case where the RTwsfcorral[] list contains only one entry, <i>wsf_tail</i> should be marked WSFSAMEIDX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and the associated wide band list of RTwsfcorral[] entries will be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The position in the list of the RTwsfcorral[] array entry that was found invalid.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the rISFGCOR tuple under audit (<i>sfgtype[0]</i> <<8) <i>sfgtype[1]</i>).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple under audit.

Table 7 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	A rISFGCOR tuple was found with no active SFG group and member data contained in it. All rISFGCOR tuples must have at least one SFG group and member marked in them (rISFGCOR.sfg[].sfg_grp != 0).
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFGCOR tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	A flag indicating no SFG data could be found by the audit (AUFALSE).
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum).

<i>Dump</i> =	The contents of the rISFGCOR tuple under audit.

Table 8 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An SFG call in the RLSFGCOR relation was found to have an unknown or invalid SFG type. The SFG type indicates the particular feature or service which is using the SFG throttling mechanism. There are two major reasons why this error may occur: (1) A new feature has very recently been added which uses the SFG throttling mechanism, and thus, a new SFG type has also been added to identify that this feature is using SFGs. The audit, however, could not identify the new type associated with this SFG call among those it has knowledge of. (2) The SFG type field in the RLSFGCOR relation which is associated with this SFG call has become corrupted.
Possible Error Effect:	Call processing applications will not be able to release this call since it cannot determine the SFG type. This will result in hung SFG data in the CMP.
Corrective Action Taken:	The call with the invalid SFG type will be zeroed out in the appropriate rISFGCOR tuple and any other VALID SF calls in the entry will be silently repacked to their most forward locations in this entry. If, after the invalid call information has been deleted, there is no remaining SFG call data in the tuple, the tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFGCOR tuple which contained which contained the bad SFG type.
<i>Bad Data</i> =	The SFG type that was found associated with this SF call (rISFGCOR.sfgtype[]).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The address of the rISFGCOR tuple.
<i>Dump</i> =	The contents of the rISFGCOR tuple.

Table 9 Error Code: LIST

Error Code:	LIST
Error Description:	The list of RTwsfcorral[] wide band array entries linked to the rISFGCOR tuple under audit was invalid. For every rISFGCOR tuple marked for wide band (rISFGCOR.wsf_tail != 0) a list of one to AUWBLISTSZ RTwsfcorral[] wide band array entries should be linked off of the rISFGCOR tuple.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s). Hung SFG member data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple will be deleted, and any associated wide band list of RTwsfcorral[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	For cases where the head entry of the list cannot be accessed (error type is marked AUWHEAD) or cases where the list size was out of range (error type is AUWLSIZE), the value dumped will be AUWBLISTSZ, indicating the list itself could not be accessed and audited from the rISFGCOR tuple. For all other cases, the entry position in the RTwsfcorral[] wide band array list between zero and AUWBLISTSZ where the error was found will be dumped.
<i>Bad Data</i> =	The contents of the <i>prev_idx</i> and <i>next_idx</i> linkages in the RTwsfcorral[] wide band array entry which was found to be corrupt (<i>prev_idx</i> << 16) <i>next_idx</i>).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the rISFGCOR tuple under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index</i> =	The key of the rISFGCOR tuple under audit (rISFGCOR.crnun) and the error

	type ((key of tuple under audit << 16) error type). The error type indicates the specific problem that was found with the RTwsfcorral[] wide band array list. The error types are: AUWHEAD = The head entry of the RTwsfcorral[] wide band array list could not be accessed or was not found linked to the rISFGCOR tuple. For RTwsfcorral[rISFGCOR.wsf_head].cid, rISFGCOR.cnum must be equal to cid in RTwsfcorral[]. AUWTAIL = The tail entry of the RTwsfcorral[] wide band array was not found linked to the rISFGCOR tuple. For RTwsfcorral[rISFGCOR.wsf_tail].cid, rISFGCOR.cnum must equal cid in RTwsfcorral[] for cases where the wide band list contains more than one entry. AUWNRANGE = A RTwsfcorral[] list entry was found which is not the tail of the list, but with a next_idx field (index to the next entry on the list) out of range. This field must be zero or greater, but less than WSFMAXCOR. AUWPLINK = A RTwsfcorral[] list entry did not point back to its predecessor (prev_idx was invalid). AUWLSIZE = The RTwsfcorral[] wide band array list was found to contain too many array entries for this wide band call. The list must contain one to AUWBLISTSZ entries.
<i>Dump</i> =	The contents of the rISFGCOR tuple under audit.

Table 10 Error Code: NO_ICR

Error Code:	NO_ICR
Error Description:	The corresponding rICR static tuple for an ICR SFG call could not be found. For ICR SFG calls, a static rICR tuple should exist, such that rICR.sfg_num == rISFGCOR.sfg[].sfg_grp, AND rISFGCOR.sfgtype[] == SFGICR.
Possible Error Effect:	An ICR SFG call (member) is associated with a simulated facility group (SFG) which does not exist. This is an illegal condition, and the call is stuck busy, since its corresponding data cannot be properly accessed. This will result in asserts out of call processing data, and hung SFG data in the ICR tracking SM.
Corrective Action Taken:	The ICR SFG data for the call is zeroed out from the rISFGCOR tuple. If this is the last data populated in the tuple, then the tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFGCOR tuple.
<i>Bad Data</i> =	The SFG group number in RLSFGCOR associated with the ICR SFG (rISFGCOR.sfg[].sfg_grp).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISFGCOR tuple where the error was found (rISFGCOR.cnum).
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 11 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	The associated rPCBLA tuple could not be found for a 6-port call located in a rISFGCOR tuple. For all 6-port related processes, there should be a corresponding rPCBLA tuple with its key equal to the 6-port process number contained in the PID field of the rISFGCOR tuple, (where rISFGCOR.cnum ≥ DBQNCRECO, AND rISFGCOR.sfgtype[] == SFG6PORT AND rISFGCOR.pid.procno == rPCBLA.procno).
Possible Error Effect:	Possible customer reorder treatment because the 6-port SF calls located in this entry are stuck busy.

Corrective Action Taken:	The rISSFGCOR tuple associated with the 6-port conference call is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	The process number found in the rISFGCOR tuple for which there was no corresponding rIPCBLA tuple (rISFGCOR.pid.procno).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump =</i>	The contents of the rISFGCOR tuple where the error was found.

Table 12 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The SFG call information found in this entry was found in illegal location in the RLSFGCOR relation. Non-6-port type SFG calls belong in the Range indexed below the maximum call record (rISFGCOR.cnum < CRMAX and rISFGCOR.sfgtype[] != SFG6PORT). 6-port conference circuit SFG calls belong in the range indexed at or above the maximum configurable Call Record (rISFGCOR.cnum ≥ DBQNCRECOR < SFGCORMAX AND rISFGCOR.sfgtype[] == SFG6PORT).
Possible Error Effect:	Lost resources since the SFG call data cannot be correctly accessed. Hung SFG data in the CMP.
Corrective Action Taken:	The rISFGCOR tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	The SFG type found at the incorrect location (rISFGCOR.sfgtype[]).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump =</i>	The contents of the rISFGCOR tuple where the error was found.

Table 13 Error Code: PID_6PORT

Error Code:	PID_6PORT
Error Description:	The 6-port SF call(s) in a rISFGCOR tuple were found to be associated with a process which is not alive. For 6-port calls, a live process ID must be marked in rISFGCOR.pid.
Possible Error Effect:	The 6-port SF call(s) in this rISFGCOR tuple are stuck busy, and the member data in the CMP is in a hung state.
Corrective Action Taken:	The tuple found in error is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	The process number of the dead 6-port process (rISFGCOR.pid.procno).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump =</i>	The contents of the rISFGCOR tuple where the error was found.

Table 14 Error Code: SFG_CRECORD

Error Code:	SFG_CRECORD
Error Description:	SFG call(s) in a rISFGCOR tuple in the non-6-port range (rISFGCOR.cnum < CRMAX) of the RLSFGCOR relation did not have a corresponding call record tuple in use. For all rISFGCOR tuples not associated with 6-port calls, there should be a valid call record for the tuple (where rICRECORD.cnum == rISFGCOR.cnum).
Possible Error Effect:	All of the SFG calls in this entry are stuck busy, and the associated data in the CMP is hung. Possible customer reorder treatment may result.
Corrective Action Taken:	The rISFGCOR tuple where the error was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	The key of the associated rICRECORD tuple which could not be accessed

	(rICRECORD.cnum).
<i>Good Data</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump</i> =	The contents of the rISFGCOR tuple where the error was found.

Table 15 Error Code: STATE_SET

Error Code:	STATE_SET
Error Description:	A bit was marked in the RTccsfg[] 6-Port conference circuit SFG global array indicating that a 6-port SFG call is active, however no corresponding tuple in the RLSFGCOR relation could be found. For every bit marked in the RTccsfg[] array (set to 0), there should exist a corresponding tuple in the RLSFGCOR relation, such that rISFGCOR.cnum(key) == (DBQNCRECOR + bit position in RTccsfg[]).
Possible Error Effect:	The 6-port SFG call(s) associated with this entry do not have a valid call ID, or tuple in RLSFGCOR and may become stuck busy, since call processing applications will not be able to access them.
Corrective Action Taken:	The appropriate bit in the RTccsfg[] array will be marked to an idle condition (set to 1).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the RTccsfg[] global array.
<i>Bad Data</i> =	The bit position in the array which was found invalidly marked.
<i>Good Data</i> =	The correct state for the bit (DBYES).
<i>Logical Index</i> =	The corresponding key of the unpopulated rISFGCOR tuple which the bit mapped to (DBQNCRECOR + bit).
<i>Dump</i> =	The entire contents of the RTccsfg[] array.

Table 16 Error Code: STATUS

Error Code:	STATUS
Error Description:	SFG call(s) in a rISFGCOR tuple in the non-6-port range (rISFGCOR.cnum < CRMAX) of the RLSFGCOR relation did not have a corresponding call record tuple which was marked busy and active for SFG calls. For all rISFGCOR tuples not associated with 6-port calls, there should be a valid call record which is marked busy for SFG calls associated with the rISFGCOR tuple (where rICRECORD.cnum == rISFGCOR.cnum, rICRECORD.crlink.st_ownk - DBSFDB_HEA == DBCRECBUSY, AND rICRECORD.sfg_call == DBYES).
Possible Error Effect:	All of the SFG calls in this entry are stuck busy, and the associated data in the CMP is hung. Possible customer reorder treatment may result.
Corrective Action Taken:	The rISFGCOR tuple where the error was found is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFGCOR tuple.
<i>Bad Data</i> =	The owner head cell list that the rICRECORD tuple is linked to (rICRECORD.crlink.st_ownk - DBSFDB_HEA).
<i>Good Data</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Logical Index</i> =	The key of the rISFGCOR tuple (rISFGCOR.cnum) where the error was found.
<i>Dump 1</i> =	The contents of the rISFGCOR tuple where the error was found.
<i>Dump 2</i> =	The contents of the associated call record tuple where rICRECORD.cnum == rISFGCOR.cnum.

Table 17 Error Code: TRACK_SM

Error Code:	TRACK_SM
Error Description:	The tracking SM number associated with an ICR SFG call did not match the tracking SM number stored in the static ICR data relation (RLICR). For all rISFGCOR tuples used for ICR calls (rISFGCOR.sfgtype[] == SFGICR), the tracking SM marked in the rISFGCOR tuple should match the tracking SM marked in the rLICR static tuple (where rISFGCOR.pid.pcrd == rLICR.track_sm for tuples where rISFGCOR.sfg[].sfg_grp == rLICR.sfg_num).

Possible Error Effect:	Lost resources, and asserts from call processing applications. Hung SFG members on the ICR tracking SM.
Corrective Action Taken:	The tracking SM number associated with the ICR SFG call is corrected to match the one found in static data (rlICR.track sm).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	The tracking SM number found in the rISFGCOR tuple (rISFGCOR.pid.pcrd).
<i>Good Data =</i>	The tracking SM number found in static data (rlICR.track sm).
<i>Logical Index =</i>	The key of the rISFGCOR tuple (rISFGCOR.cnum).
<i>Dump =</i>	The contents of the rISFGCOR tuple where the error was found.

Table 18 Error Code: UNPACKED

Error Code:	UNPACKED
Error Description:	SFG call information was not loaded properly into a tuple in the RLSFGCOR relation. The SFG call data containing SFG group, member number, and type must be "packed" into the most forward (lowest indexed) locations of the <i>sfg[]</i> and <i>sfgtype[]</i> arrays in each rISFGCOR tuple.
Possible Error Effect:	Invalid operation out of call processing applications. Stuck SFG member data, and hung SFG data in the CMP.
Corrective Action Taken:	The SFG data is repacked to its most forward locations. Note: If the inconsistency exists because of earlier audit recovery, the audit recovers (repacks) the data without firing an error. In any other instance, this error will be fired.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFGCOR tuple.
<i>Bad Data =</i>	The empty position in the entry (rISFGCOR.sfg[position]) which preceded a non-empty position.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the rISFGCOR tuple where the error was found.
<i>Dump =</i>	The contents of the rISFGCOR tuple where the error was found.

SFGCORRAL

Software Release: 5E10 and later

1. SIMULATED FACILITY GROUP CALL INFORMATION STATUS GLOBAL ARRAY (SFGCORRAL) AUDIT

All simulated facility group (SFG) call data in the switching modules (SMs) is consolidated into one dynamic data area called the SFG CORRAL. The SFGCORRAL audit verifies the live simulated facility (SF) call data contained in this SFG CORRAL global array (RTsfgcorral[]). This array exists in all SMs.

2. DATA STRUCTURES

Each SFG CORRAL entry is indexed by a call record ID. For 6-port conference circuit calls, there is no call record and thus, no call ID, the indexes at, or above the maximum call record ID are used. All other SF call types, stored in entries, are indexed below the maximum call record ID. There is space in the SFG CORRAL for one entry for each call record, plus one entry for each possible 6-port SF, in each SM. The SFG CORRAL and its related data structures are shown in Figure 1. Each SFG CORRAL entry contains space for up to three SF calls, since each call ID can control up to three SF calls. Each SF call in an entry contains the following:

- (1) The SFG group number associated with the SF call.
- (2) The SFG member number associated with the SF call.
- (3) The SFG type (feature type) associated with the SF call. There may be only 1 intercom call restriction (ICR) type SF call per entry.
- (4) A special field that has different uses. If one of the SF calls in an entry is an ICR call, this field contains the ICR tracking SM number associated with this call. If the SF call is associated with a 6-port conference circuit, this field contains a special 6-port keep alive terminal process, which periodically checks every 5 minutes, to make sure that the 6-port terminal process on the originating SM is still alive. If not, this special keep alive process ceases, and the 6-port call is terminated. For any other types of SF calls, this field is not used.

The RTsfgcorral[] general SFG call scenario is presented in Figure 1.

For members associated with wide band SFG calls, an implied linkage exists between the RTsfgcorral[] array entry and a list of RTwsfcorral[] array entries. For these linkages, *wsf_head* is the index into the RTwsfcorral[] entry which is serving as the head of the wide band list. The *wsf_tail* field is used as the index into the RTwsfcorral[] array entry which is serving as the tail of the wide band list. For wide band lists of one RTwsfcorral[] entry, RTsfgcorral[cid].wsf_tail is always marked WSFSAMEIDX.

The RTsfgcorral[] wide band SFG call scenario is presented in Figure 2 and Figure 3.

3. AUDIT CHECKS

The audit traverses through every entry in the SFG CORRAL and checks the SF call status data contained in every entry. In doing so, the audit makes the following checks:

- (1) The audit checks that each SF call is located in the correct range of the SFG CORRAL. Any 6-port conference circuit calls should be located at or above the maximum allowable call record. All other types should be indexed below the maximum call record.
- (2) For 6-port SF calls, the audit verifies that there is a live terminal process associated with the call.
- (3) For non-6-port SF calls, including ICR, the audit verifies that there is a valid (busy) call record associated with

the call, and that the call contains a valid SFG type.

- (4) For ICR SF calls, the audit verifies the tracking SM number with the one stored in static data.
- (5) For all populated entries, the audit verifies that the SF call data is "packed" into the most forward locations in each entry, and that its group number is in the valid range of group numbers for rIRT_SFG or rLICR.
- (6) For entries associated with Wide Band SFG calls, the SFGCORRAL audit verifies the *wsf_head* and *wsf_tail* linkages to the RTwsfcorral[] array, the integrity of all linkages in the RTwsfcorral[] array, and that the SFG hunt type(s) marked in the RTsfgcorral[] array are valid for the Wide Band feature.

The SFGCORRAL audit takes the following recovery actions when any of the checks fail:

- (1) If the SF call is found in an entry out of the proper range for that particular SF call type, the entire entry corresponding to the SF call ID is zeroed out.
- (2) Any SFG CORRAL entries containing 6-port conference circuit SF calls which are found without an associated live terminal process, are zeroed out.
- (3) Any SFG CORRAL entries containing non-6-port calls, including ICR SF calls, that are found without an associated call record which is in use, or contain group numbers that are out of the legal range, are zeroed out.
- (4) Any SF calls with an unknown or invalid SFG type are zeroed from their entry, and the remaining SF calls in that entry (if any) are silently repacked into their most forward locations in that entry.
- (5) For ICR SF calls, an invalid tracking SM number is corrected to match the one found in the rLICR tuple whose key matched the ICR call's SF group number. If an associated rLICR tuple matching an ICR call's SF group number in an SFG CORRAL entry does not exist, the ICR SF call is zeroed out.
- (6) Any SFG CORRAL entries found with SF call data which is not packed into the most forward locations in an entry, and this conditions IS NOT THE RESULT OF EARLIER AUDIT RECOVERY, are repacked into their most forward locations after an error message is fired.
- (7) If the entry in RTsfgcorral[] is being used for wide band, and the SFG hunt type is invalid, then the array entry is zeroed out.
- (8) If the entry in RTsfgcorral[] is being used for wide band, and the SFG group information is invalid, then the array entry is zeroed out.
- (9) For wide band list errors, the list of associated RTwsfcorral[] array entries is quarantined for recovery, and the RTsfgcorral[] array entry is zeroed out.

4. ERROR HANDLING

The error code produced by the SFGCORRAL audit are listed below. See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CALL_ID

Error Code:	CALL_ID
Error Description:	The SFGCORRAL audit was able to access the rIPCBLA tuple, however, the call record ID in the rIPCBLA tuple did not match the index of the RTSFGCORRAL

	entry containing the 6-port SF call.
Possible Error Effect:	The 6-port SF calls associated with this entry does not have a valid call ID, and may become stuck busy, since call processing will not be able to access them.
Corrective Action Taken:	The entire SFG CORRAL entry, associated with the 6-port related array index, which does not match the rIPCBLA call ID, is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the RTSFGCORRAL entry.
<i>Bad Data</i> =	The call ID found in the rIPCBLA tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The RTSFGCORRAL entry index where the error was found.
<i>Dump 1</i> =	The RTSFGCORRAL array entry where the error was found.
<i>Dump 2</i> =	The rIPCBLA tuple where the call ID was found.

Table 2 Error Code: GRPINFO

Error Code:	GRPINFO
Error Description:	The RTsfscorrall[] array is marked with more SFG groups than is allowed for wide band. The RTsfscorrall[] array holds up to MAXSF groups, but the legal limit for Wide Band is (MAXSF - 1).
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The RTsfscorrall[] array entry will be zeroed out, and any associated wide band list of RTwsfcorrall[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the RTsfscorrall[] array entry under audit (<i>sfgtype[0]</i> << 8) <i>sfgtype[1]</i>).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the RTsfscorrall[] array entry under audit (<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index</i> =	The index if the RTsfscorrall[] array entry under audit.
<i>Dump</i> =	The contents of the RTsfscorrall[] array entry under audit.

Table 3 Error Code: GRPTYP_NUM

Error Code:	GRPTYP_NUM
Error Description:	The group number for this SF call is not in the legal range of allowable SF group numbers.
Possible Error Effect:	Asserts from call processing, and eventual illegal segment table entry errors in the AM.
Corrective Action Taken:	The SFG corral entry position for this SF call will be zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the RTSFGCORRAL entry.
<i>Bad Data</i> =	The SF group number found.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The INDEX if the RTSFGCORRAL entry.
<i>Dump</i> =	The RTSFGCORRAL entry where the error was found.

Table 4 Error Code: HTYPE

Error Code:	HTYPE
Error Description:	The SFG group type <i>sfgtype[]</i> marked in the RTsfscorrall[] array is invalid for wide band. For wide band SFG calls, the SFG group type must be either SFGEDSL, SFG2WESSX, SFGOUTESSX, or SFGINESSX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The RTsfscorrall[] array entry will be zeroed out, and any associated wide band list of RTwsfcorrall[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the RTsfgcorral[] array entry under audit ((sfgtype[0] <<8) sfgtype[1]).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the RTsfgcorral[] array entry under audit ((wsf_head << 16) wsf_tail).
<i>Logical Index</i> =	The index if the RTsfgcorral[] array entry under audit.
<i>Dump</i> =	The contents of the RTsfgcorral[] array entry under audit.

Table 5 Error Code: INDEX

Error Code:	INDEX
Error Description:	One or both of the wide band array linkages in RTsfgcorral[] were found to be out of range. The <i>wsf_head</i> and <i>wsf_tail</i> linkages should contain valid values between zero and (WSFMAXCOR - 1) which is the maximum allowable entry in RTwsfcorral[] or, for the case where the RTwsfcorral[] list contains only one entry, <i>wsf_tail</i> should be marked WSFSAMEIDX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The RTsfgcorral[] array entry will be zeroed out, associated wide band list of RTwsfcorral[] entries will be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The contents of the SFG hunt type data <i>sfgtype[]</i> from the RTsfgcorral[] array entry under audit ((sfgtype[0] <<8) sfgtype[1]).
<i>Good Data</i> =	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the RTsfgcorral[] array entry under audit ((wsf_head << 16) wsf_tail).
<i>Logical Index</i> =	The index if the RTsfgcorral[] array entry under audit.
<i>Dump</i> =	The contents of the RTsfgcorral[] array entry under audit.

Table 6 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	An SF call in the SFG CORRAL was found to have an unknown or invalid SFG type. The SFG type indicates the particular feature or service which is using the SFG throttling mechanism. There are two major reasons why this error may occur: (1) A new feature has very recently been added which uses the SFG throttling mechanism, and thus, a new SFG type has also been added to identify that this feature is using SFGs. The audit, however, could not identify the new type associated with this SF call among those it has knowledge of. (2) The SFG type field in the SFG CORRAL which is associated with this SF call has become corrupted.
Possible Error Effect:	Keep Alive will not be able to release this call since it cannot determine the SFG type.
Corrective Action Taken:	The call with the invalid SFG type will be zeroed out in the SFG CORRAL, and any other VALID SF calls in the entry will be silently repacked to their most forward locations in this entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the RTSFGCORRAL entry which contained the SF call with the bad SFG type.
<i>Bad Data</i> =	The SFG type that was found associated with this SF call.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The RTSFGCORRAL entry index where the error was found.

<i>Dump =</i>	The RTSFGCORRAL array entry where the error was found.

Table 7 Error Code: LIST

Error Code:	LIST
Error Description:	The list of RTwsfcorral[] wide band array entries linked off of the RTsfgcorral[] array entry under audit was invalid. For every RTsfgcorral[] array entry marked for wide band (RTsfgcorral[].wsf_tail != 0) a list of one to AUWBLISTSZ RTwsfcorral[] wide band array entries should be linked off of the RTsfgcorral[] entry.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The RTsfgcorral[] array entry will be zeroed out, and any associated Wide Band list of RTwsfcorral[] entries will be quarantined. These entries will then be recovered by the WSFCORRAL audit when it is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	For cases where the head entry of the list cannot be accessed (error type is marked AUWHEAD) or cases where the list size was out of range (error type is AUWLSIZE), the value dumped will be AUWBLISTSZ. For all other cases, the entry position in the RTwsfcorral[] wide band array list between zero and AUWBLISTSZ where the error was found will be dumped.
<i>Bad Data =</i>	The contents of the <i>prev_idx</i> and <i>next_idx</i> linkages in the RTwsfcorral[] wide band array entry which was found to be corrupt (<i>prev_idx</i> << 16) <i>next_idx</i>).
<i>Good Data =</i>	The contents of the <i>wsf_head</i> and <i>wsf_tail</i> linkages from the RTsfgcorral[] array entry under audit ((<i>wsf_head</i> << 16) <i>wsf_tail</i>).
<i>Logical Index =</i>	The index if the RTsfgcorral[] array entry under audit and the error type ((entry under audit << 16) Error Type). The error type indicates the specific problem that was found with the RTwsfcorral[] wide band array list. the error Types are: AUWHEAD = The head entry of the RTwsfcorral[] wide band array list could not be accessed or was not found linked to the RTsfgcorral[] array. For RTwsfcorral[RTsfgcorral[INDEX].wsf_head].cid, INDEX must be equal to <i>cid</i> in RTwsfcorral[]. AUWTAIL = The tail entry of the RTwsfcorral[] wide band array was not found linked to the RTsfgcorral array. For RTwsfcorral[RTsfgcorral[INDEX].wsf_tail].cid, INDEX must equal <i>cid</i> in RTwsfcorral[] for cases where the wide band list contains more than one entry. AUWNRANGE = A RTwsfcorral[] list entry was found which is not the tail of the list, but with a <i>next_idx</i> field (index to the next entry on the list) out of range. This field must be zero or greater, but less than WSFMAXCOR. AUWPLINK = A RTwsfcorral[] list entry did not point back to its predecessor (<i>prev_idx</i> was invalid). AUWLSIZE = The RTwsfcorral[] wide band array list was found to contain too many array entries for this wide band call. The list must contain one to AUWBLISTSZ entries.
<i>Dump =</i>	The contents of the RTsfgcorral[] array entry under audit.

Table 8 Error Code: NO_ICR

Error Code:	NO_ICR
Error Description:	The corresponding rICR static tuple for an ICR SF call could not be found.
Possible Error Effect:	An ICR SF call (member) will be associated with a simulated facility group (SFG)

	which does not exist. This is an illegal condition, and the call is stuck busy, since its corresponding data cannot be properly accessed.
Corrective Action Taken:	The ICR SF call was zeroed out from the entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the RTSFGCORRAL entry.
<i>Bad Data</i> =	The SF group (SFG) number in RTSFGCORRAL associated with the ICR SF.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The RTSFGCORRAL entry index where the error was found.
<i>Dump</i> =	The RTSFGCORRAL array entry where the error was found.

Table 9 Error Code: NOPCBLA

Error Code:	NOPCBLA
Error Description:	For all 6-port related processes, there should be a corresponding rIPCBLA tuple with its key equal to the 6-port process number contained in the pid field of the SFG CORRAL global array. The SFGCORRAL audit could not find this corresponding rIPCBLA tuple.
Possible Error Effect:	The 6-port SF calls located in this entry are stuck busy, since they have no associated call process.
Corrective Action Taken:	The entire SFG CORRAL entry associated with the 6-port process number which has no corresponding rIPCBLA is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the RTSFGCORRAL entry.
<i>Bad Data</i> =	The process number found in the entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The RTSFGCORRAL entry index where the error was found.
<i>Dump</i> =	The RTSFGCORRAL array entry where the error was found.

Table 10 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The SF call information found in this entry was found in illegal location in the SFG CORRAL global array. Non-6-port type SF calls belong in the range indexed below the maximum call record. 6-port calls belong in the range indexed at or above the maximum call record.
Possible Error Effect:	Lost resources since the SF call cannot be correctly accessed.
Corrective Action Taken:	The entire SFG CORRAL entry containing the SF call is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the RTSFGCORRAL entry.
<i>Bad Data</i> =	The SF type found at the incorrect location.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The RTSFGCORRAL entry index where the error was found.
<i>Dump</i> =	The RTSFGCORRAL array entry where the error was found.

Table 11 Error Code: PID_6PORT

Error Code:	PID_6PORT
Error Description:	The 6-port SF call(s) in this entry were found to be associated with a process which is not alive.
Possible Error Effect:	The 6-port SF call(s) in this RTSFGCORRAL entry are stuck busy.
Corrective Action Taken:	The entire entry associated with the dead 6-port process is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The Address of the RTSFGCORRAL entry.
<i>Bad Data</i> =	The process number of the dead 6-port process.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The RTSFGCORRAL entry index where the error was found.
<i>Dump</i> =	The RTSFGCORRAL array entry where the error was found.

Table 12 Error Code: SFG_CRECORD

Error Code:	SFG_CRECORD
Error Description:	An SF call(s) in the non-6-port range of the SFG CORRAL array does not have a corresponding call record in use.
Possible Error Effect:	All of the SF calls in this entry are stuck busy.
Corrective Action Taken:	The entire RTSFGCORRAL entry where the error was found is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the RTSFGCORRAL entry.
<i>Bad Data =</i>	The status field of the corresponding call record.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The RTSFGCORRAL entry index where the error was found.
<i>Dump =</i>	The RTSFGCORRAL array entry where the error was found.

Table 13 Error Code: TRACK_SM

Error Code:	TRACK_SM
Error Description:	The tracking SM number associated with an ICR SF call did not match the tracking SM number stored in static data (rllCR).
Possible Error Effect:	Lost resources, and asserts from call processing.
Corrective Action Taken:	The tracking SM number associated with the ICR SF call was corrected to match the one found in static data (rllCR).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the RTSFGCORRAL entry.
<i>Bad Data =</i>	The tracking SM number found in the RTSFGCORRAL entry.
<i>Good Data =</i>	The tracking SM number found in static data (rllCR).
<i>Logical Index =</i>	The RTSFGCORRAL entry index where the error was found.
<i>Dump =</i>	The RTSFGCORRAL array entry where the error was found.

Table 14 Error Code: UNPACKED

Error Code:	UNPACKED
Error Description:	SF call information was not loaded properly into an entry of the SFG CORRAL. The SF call data containing SF group, member number, and type must be "packed" into their most forward (lowest indexed) locations of the small arrays of structures in each CORRAL entry for efficiency. If this error is seen, it is because a call process wrote data incorrectly into the SFG CORRAL global array.
Possible Error Effect:	A process accessing the SF data might assume that an entry is empty because the first SF position is zero.
Corrective Action Taken:	If the inconsistency exists because of earlier audit recovery, the audit recovers (repacks) the data without firing an error. In any other instance, this error will be fired.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The Address of the RTSFGCORRAL entry.
<i>Bad Data =</i>	The empty position in the entry which preceded a non-empty position.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The RTSFGCORRAL entry index where the error was found.
<i>Dump =</i>	The RTSFGCORRAL array entry where the error was found.

Table 15 Error Code: ZAP

Error Code:	ZAP
Error Description:	An addressing error occurred while the SFGCORRAL audit was executing.
Possible Error Effect:	The audit was not able to finish.
Corrective Action Taken:	Zap the cursor.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of AUcursor.
<i>Bad Data =</i>	AUcursor.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

SFSTAT

Software Release: 5E10 and later

1. SIMULATED FACILITY STATUS (SFSTAT) AUDIT

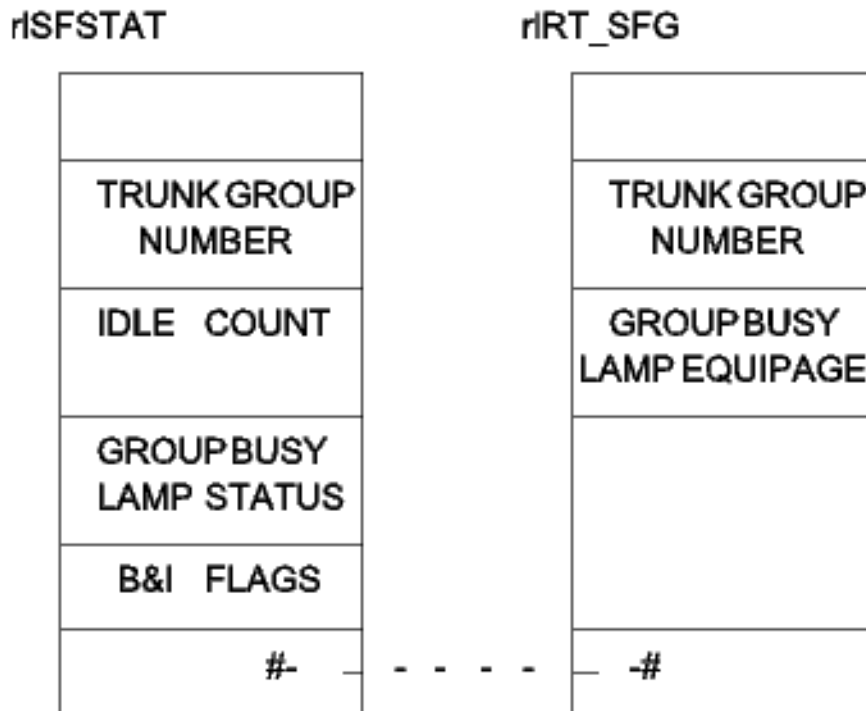
Simulated facility groups (SFGs) provide customers with the appearance of dedicated trunks without requiring the local switching office to designate specific hardware for each customer. Simulated facility groups thus increase the efficiency of the network while maintaining all the customer's features and needs. The simulated groups allow customers with private branch exchange (PBX) or centrex features status lights for each group. When the group is completely busy, the busy lamp status flag lights. Another feature, the make-busy key, allows the customers to give the appearance of a busy trunk group. This is done, for example, when a conference call needs to be placed at a specific time. Customers with ISDN attendant consoles associated with their groups may also subscribe to the warning lamp feature, whereby, as the number of available members approaches 0 (all busy) the warning lamp lights to alert the customer of this condition.

The simulated facility status audit (SFSTAT) verifies the existence and consistency of the data in small (1-96 trunks) simulated facility groups. For information about larger trunk groups (more than 96 trunks), refer to the large simulated facility status audit (LGSFSTAT).

The SFSTAT audit is part of the routing and terminal allocation (RTA) audits subgroup. It runs in the communications module processor (CMP) in the initialization and segmented modes.

2. DATA STRUCTURES

This audit checks several fields in the simulated facility group status relation (RLSFSTAT), a dynamic relation in the CMP. These fields include the key, idle count, and group busy and warning lamp fields. The rLSFSTAT relation has one tuple for each simulated facility group. Each tuple contains one bit, a busy/idle flag, for each member. The audit verifies the information in RLSFSTAT with information in the static simulated facility group relation (RLRT_SFG). The data structures of the SFSTAT audit are shown in Figure 1 .



#- - - -# = NONSTANDARD LINKAGE

Figure 1 (SFSTAT-1) Data Structures of the SFSTAT Audit

3. AUDIT CHECKS

The audit calls the static driver (DRSTATIC) to step through RLSFSTAT and perform a dynamic-to-static existence check based on the key (trunk group number) fields of RLSFSTAT and RLRT_SFG to ensure that a tuple in RLRT_SFG exists for each tuple in RLSFSTAT. If the office is equipped with the busy lamp, the audit performs the following semantic checks.

- (a) During the existence check, the audit counts the number of idle bits. When the semantic check function is invoked, the audit compares its idle bit count with the automatic count made by call processing. Unlike the trunk status audits TKSTAT, LGTKSTAT and TKOWNER. This check of the idle count is unconditional of whether or not an analog console is associated with this simulated facilities group.
- (b) If there is an analog console associated with the SFG, the audit checks that the status of the make-busy key, the idle count, and busy lamp are consistent. If the make-busy key is pressed, the lamp should be on, regardless of the idle count.
- (c) If there is an integrated services digital network (ISDN) attendant console associated with this simulated facilities group, the audit checks that the busy lamp and make-busy key of this console are consistent with the idle count. If the ISDN attendant console is equipped with a warning lamp, this lamp's state is checked against the idle count and the preset value in the simulated facilities group, information relation RLRT_SFG. If the make-busy key is pressed, the busy and warning lamps should be on regardless of the idle count.

4. ERROR HANDLING

If the dynamic-to-static existence check fails because the audit finds no static tuple corresponding to the tuple in

RLSFSTAT, the audit deletes the tuple in RLSFSTAT.

If the semantic check fails, the audit initializes the rISFSTAT tuple. The audit corrects counts and busy lamp errors.

The audit corrects ISDN attendant console busy and warning lamp inconsistencies if a simulated facilities group is equipped with such lamps.

When running in the keep alive mode, when stuck bits are detected and idled, the idle count [count of available simulated facility (SF) members left in this group] is updated (incremented) to account for all bits that were found stuck, and subsequently idled. Any analog busy lamps, and/or ISDN attendant busy or warning lamps are refreshed, as needed, to correspond to the new count of available SF members in the group.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	A count of the idle group members did not equal the stored idle count.
Possible Error Effect:	Measurements would be in error for this group. Possible mishandling of the group busy lamp -- if equipped.
Corrective Action Taken:	The idle count was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rISFSTAT tuple address.
<i>Bad Data =</i>	idle_count field.
<i>Good Data =</i>	Correct count of idle members.
<i>Logical Index =</i>	Group number.
<i>Dump =</i>	rISFSTAT tuple.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Group number not in static relation RLRT_SFG.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Group number not in static relation RLRT_SFG.
<i>Dump =</i>	SFSTAT tuple that was deleted.

Table 3 Error Code: ISDNTGBWL

Error Code:	ISDNTGBWL
Error Description:	The isdntgbwl field, which indicates if this simulated facility group (SFG) is associated with an ISDN attendant Console busy and possible warning lamp, and is a dynamic duplicate of the isdntgbwl field stored in static data (RLRT_SFG), did not match the corresponding field stored in the RLRT_SFG static relation.
Possible Error Effect:	Call processing will be unable to determine if an ISDN attendant console is associated with this SFG, and will be unable to perform manipulations on the busy lamp, warning lamp, and make-busy key for this console.
Corrective Action Taken:	The dynamic representation of this data (RLSFSTAT) was corrected to match its static counterpart in RLRT_SFG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISFSTAT tuple.

<i>Bad Data</i> =	The isdntgbwl field found in the rISFSTAT tuple.
<i>Good Data</i> =	The isdntgbwl field found in the rIRT_SFG static tuple.
<i>Logical Index</i> =	The key (SF group number) of the rISFSTAT tuple.
<i>Dump</i> =	The rISFSTAT tuple.

Table 4 Error Code: LAMP

Error Code:	LAMP
Error Description:	This SFG's analog and/or ISDN attendant console (AC) group busy lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISFSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	SF group number.
<i>Dump</i> =	rISFSTAT tuple.

Table 5 Error Code: RANGE

Error Code:	RANGE
Error Description:	The idle count (rISFSTAT.idle_count) which is the count of idle SFG members could not be audited because the hunt type in the RLRT_SFG static relation (rIRT_SFG.hunt_type) was not marked properly (where rIRT_SFG.grpnum = rISFSTAT.grpnum). For small SFGs, the hunt type must be marked RTH_SFG. This error will fire until fixed by the user.
Possible Error Effect:	The count of idle members may become corrupt causing customers to get reorder treatment and call failures.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, read the RLRT_SFG relation with key set to the key which dumped as "good data" with this error code message. Modify the hunt_type field to RTH_SFG.
Dump Description:	
<i>Error Address</i> =	The address of the rISFSTAT tuple.
<i>Bad Data</i> =	The hunt_type field found in the rIRT_SFG static tuple.
<i>Good Data</i> =	The group number (rIRT_SFG.grpnum) of the rIRT_SFG static tuple.
<i>Logical Index</i> =	The key (SF group number) of the rISFSTAT tuple (rISFSTAT.grpnum).
<i>Dump</i> =	The contents of the rISFSTAT tuple.

Table 6 Error Code: SFGBL

Error Code:	SFGBL
Error Description:	The sfgbl field, which indicates if this SFG is associated with an analog console busy lamp, and is a dynamic duplicate of the sfgbl field stored in static data (RLRT_SFG), did not match the corresponding field stored in the RLRT_SFG static relation.
Possible Error Effect:	Call processing will be unable to determine if an analog console is associated with this SFG, and will be unable to perform manipulations on the busy lamp and make-busy key for this console.
Corrective Action Taken:	The dynamic representation of this data (rISFSTAT) was corrected to match its static counterpart in rIRT_SFG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFSTAT tuple.
<i>Bad Data</i> =	The sfgbl field found in the rISFSTAT tuple.
<i>Good Data</i> =	The sfgbl field found in the rIRT_SFG static tuple.
<i>Logical Index</i> =	The key (SF group number) of the rISFSTAT tuple.
<i>Dump</i> =	The rISFSTAT tuple.

Table 7 Error Code: TLAMP

Error Code:	TLAMP
Error Description:	This SFG's ISDN AC threshold warning lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected, and the hardware state of the lamp was refreshed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISFSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	SF group number.
<i>Dump</i> =	rISFSTAT tuple.

Table 8 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not of the type used in rISFSTAT (that is, RTH_SFG) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Hunt type of group as shown in RLRT_SFG.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	SFSTAT tuple that was deleted.

Table 9 Error Code: WARNTHRESH

Error Code:	WARNTHRESH
Error Description:	The warnthresh field indicates the customer selected value for the number of idle members a customer has still available when an ISDN attendant console warning lamp (is equipped) will light. This is a dynamic duplicate of the warnthresh field stored in static data (RLRT_SFG), but it did not match the corresponding field stored in the RLRT_SFG static relation.
Possible Error Effect:	Call processing will be unable to determine if an ISDN attendant console warning lamp should be ON or OFF, and the lamp may become stuck in an incorrect state, giving the customer incorrect SF group status.
Corrective Action Taken:	The dynamic representation of this data (rISFSTAT) was corrected to match its static counterpart in rIRT_SFG.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISFSTAT tuple.
<i>Bad Data</i> =	The warnthresh field found in the rISFSTAT tuple.
<i>Good Data</i> =	The warnthresh field found in the rIRT_SFG static tuple.
<i>Logical Index</i> =	The key (SF group number) of the rISFSTAT tuple.
<i>Dump</i> =	The rISFSTAT tuple.

SGDB

Software Release: 5E10 and later

1. SERVICE GROUP DATA BLOCK (SGDB) AUDIT

The SGDB audit verifies the consistency between the dynamic RLSGDB relation and the static RLCDFI_MAP and RLCKTDATA relations for communication link digital facility interface units (CDFI). The CDFI is used for inter-remote switching module (RSM) communication in a multimodule RSM configuration. The dynamic rlsGDB tuples contain per-CDFI [associated peripheral interface control bus (PICB)] information indicating what RSM number is assigned to it and error masks and inhibit states. The RLSGDB relation is administered by peripheral control. The RLCDFI_MAP relation contains office-dependent data (ODD) configuration data indicating which RSMs are connected to which CDFIs.

The SGDB audit is part of the peripheral control audits subgroup. It runs in the segmented mode in the operating system for distributed switching-switching module (OSDS-SM) environment. The SGDB audit operates at routine level C.

2. DATA STRUCTURES

The SGDB audit uses the service group data block relation (RLSGDB), the CDFI map relation (RLCDFI_MAP), and the circuit data relation (RLCKTDATA). These data structures are shown in Figure 1 .

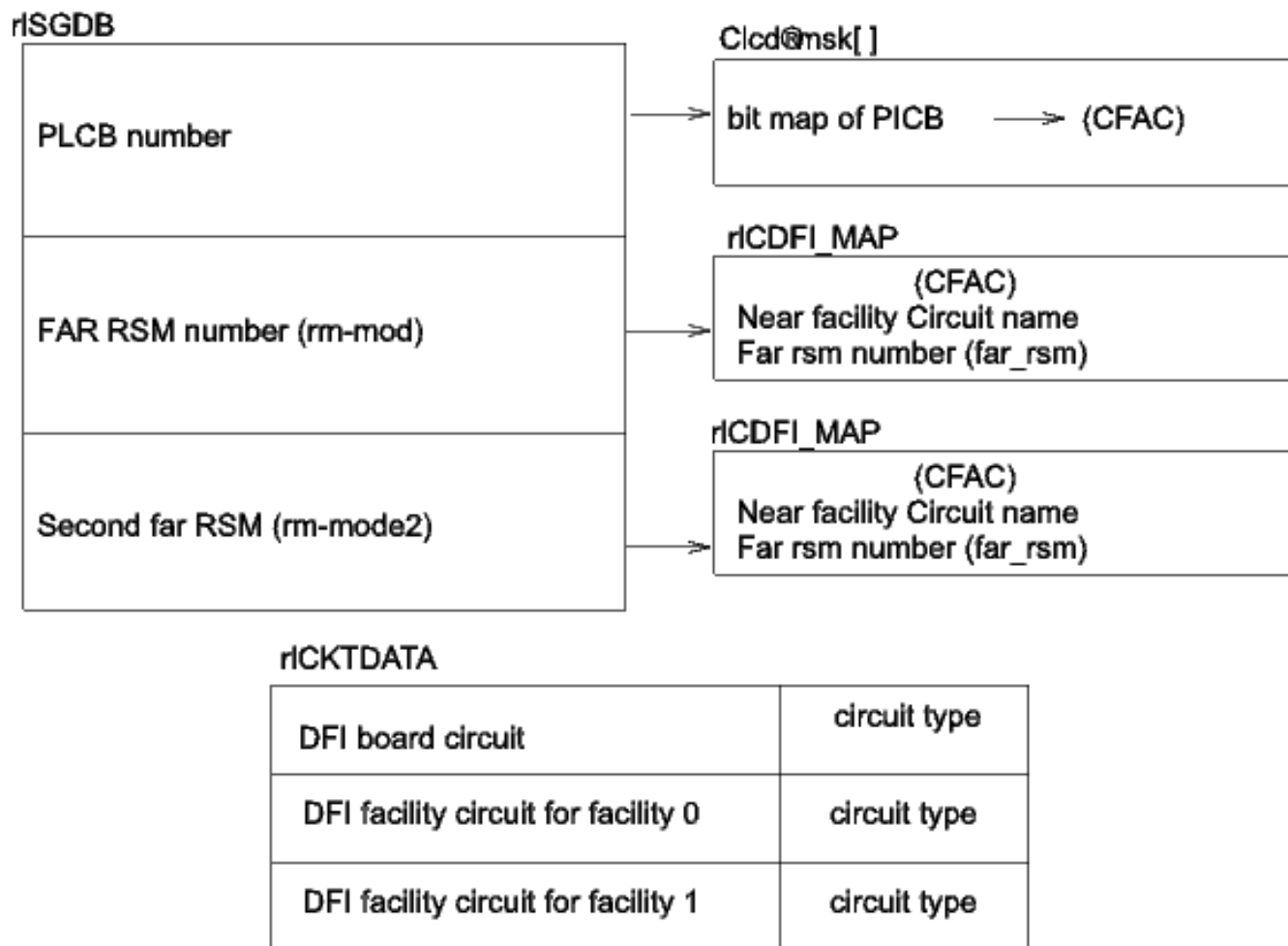


Figure 1 (SGDB-1) Data Structures Representing the Multimodule RSM Intercommunication Links (CDFIs)

3. AUDIT CHECKS

The SGDB audit invokes the DRDYN driver to step through the dynamic RLSGDB relation. For each tuple, the audit calculates the DFI board name (CDFI or RDFI), given the PICB number (key). The RLCKTDATA static relation is read for the board to verify SMDLTU or SMDLTU2 unit type. If the board is either of these unit types, the circuit name of the associated facility (one facility for SMDLTU, two facilities for SMDSTU2) is calculated. For each facility of type PCCFAC, the far RSM field is verified. For SMDLTU, the remote module field in the SGDB must match the far RSM field for facility 0 in RLCDFI_MAP. For SMDLTU2, an additional remote module field corresponding to a possible second facility, is verified in the same manner.

4. ERROR HANDLING

If errors are found, they are reported and a peripheral control function interface is used to recover the Cicdfinsk[] array. If the read of RLCKTDATA does not match, the rlSGDB tuple is recovered (far RSM number is set to zero). If the read of RLCDFI_MAP does not match, an ODD inconsistency error is reported. If the far RSM number is inconsistent between RLSGDB and RLCDFI_MAP, the rlSGDB tuple will be corrected and recovered. If the circuit type is other than a DFI and the far RSM number is not zero, it will be set to zero and recovered.

The SGDB audit schedules no related audits.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The common board associated with the SGDB does not exist in rICKTDATA, but its 'rm-mod' fields are not zero.
Possible Error Effect:	Scanning may be incorrectly handled by PC/SMIM.
Corrective Action Taken:	The PC scanning data is reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISGDB.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Zero.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	SGDB tuple.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The RLCDFI_MAP relation has no tuple for this key, yet the RLCKTDATA relation indicates there should be a tuple.
Possible Error Effect:	This will shut off message traffic on CDFIs or try to process requests from other circuits as if they were CDFIs.
Corrective Action Taken:	None.
Manual Action Required:	The RLCDFI_MAP and/or the RLCKTDATA relation should be corrected to correlate with each other.
Dump Description:	
<i>Error Address</i> =	Pointer to rISGDB tuple.
<i>Bad Data</i> =	DB return.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISGDB.

Table 3 Error Code: RMMOD

Error Code:	RMMOD
Error Description:	The rm_mod field in the RLSGDB relation for this tuple does not match the real remote module for this CDFI.
Possible Error Effect:	This will shut off message traffic on CDFIs or try to process requests from other circuits as if they were CDFIs.
Corrective Action Taken:	The rm_mod field in the rISGDB tuple is reinitialized to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to rISGDB tuple.
<i>Bad Data</i> =	Illegal rISGDB rm_mod value.
<i>Good Data</i> =	Corrected rISGDB rm_mod value.
<i>Logical Index</i> =	Circuit name.
<i>Dump</i> =	rISGDB tuple.

SILCTG

Software Release: 5E10 and later

1. SELECTIVE INCOMING LOAD CONTROL IN THE AM (SILCTG) AUDIT

Selective incoming load control (SILC) is a feature of the network management (NM) subsystem that allows local offices to ignore traffic offered to them on specified trunk groups during periods of machine congestion. Selective incoming load control allows the local office to handle its maximum number of calls without an overload occurring. By interrupting and rerouting calls before they are sent to the local office, selective incoming load control keeps the local office running efficiently when a very large number of calls are made to a certain number, such as to the Auto Club during a period of unusually bad weather.

The trunks that are given SILC treatment are listed in the selective incoming load control trunk group relation (RLSILCTG) in the administrative module (AM). The selective incoming load control trunk group audit (SILCTG) verifies information in this relation. SILCTG is part of the Administrative Services audits subgroup. It runs in the OKP environment in the initialization and segmented modes.

2. DATA STRUCTURES

The audit checks the selective incoming load control trunk group relation (RLSILCTG), a dynamic relation in the AM that contains the list of the trunk groups requiring SILC treatment. The RLSILCTG relation contains the trunk group number which, if selective incoming load control is on, prevents access to the trunk group. The audit also reads the trunk group relation (RLRT_TRKG), a static relation in the AM, to ensure the validity of the trunk group number and the group type. The audit makes a copy of RLRT_TRKG in a buffer and uses the buffer information to perform its validations. These data structures are shown in Figure 1.

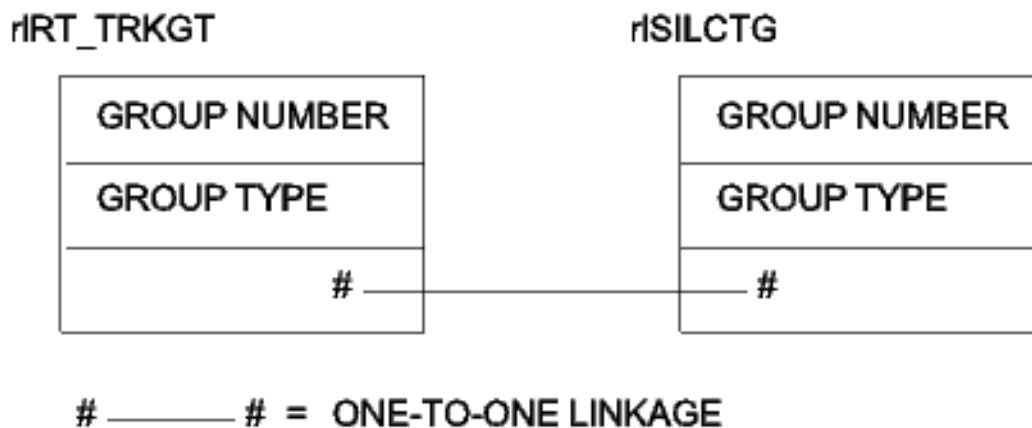


Figure 1 (SILCTG-1) Data Structures of the SILC Audit

3. AUDIT CHECKS

The SILCTG audit invokes the static driver (DRSTATIC) to perform the following checks using the group number field (grpnum) as the key:

- The grpnum and group type (grptype) fields in rLSILCTG are checked against the corresponding fields in the copy of rLRT_TRKG to ensure their validity; the group type must be incoming or two-way.
- The audit keeps a count of the rLSILCTG tuples. At its conclusion, the audit verifies the count with an internal count.

4. ERROR HANDLING

If the audit does not find the group number in rLSILCTG in the copy of the static RLRT_TRKG relation, it deletes the rLSILCTG tuple.

If the group type in rLSILCTG is neither incoming nor two-way, the erroneous tuple is removed from RLSILCTG.

If the counts do not agree, an error is reported and the internal count is updated to agree with the audit count.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra tuple was found in the SILCTG relation.
Possible Error Effect:	Restrictive actions may inadvertently be taken on a trunk group.
Corrective Action Taken:	Update the NMsilcnt and notify NM personnel of discrepancy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not used.
<i>Bad Data</i> =	Not used.
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Not used.
<i>Dump</i> =	NA.

Table 2 Error Code: KEY

Error Code:	KEY
Error Description:	The SILCTG tuple's key is incorrect.
Possible Error Effect:	Use of this tuple may propagate errors.
Corrective Action Taken:	The SILCTG key is updated and the audit will continue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SILCTG tuple.
<i>Bad Data</i> =	Invalid key value.
<i>Good Data</i> =	Valid key value.
<i>Logical Index</i> =	Valid key of the SILCTG.
<i>Dump</i> =	SILCTG tuple.

Table 3 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The external NMsilcnt reflects more silctg tuples than actually exist.
Possible Error Effect:	A trunk group could be missing from the silctg relation meaning the SILC controls would not be applied.
Corrective Action Taken:	NMsilcnt is returned to correct value. NM personnel is notified of discrepancy and we will rebuild the list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not used.
<i>Bad Data</i> =	Not used.
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Not used.
<i>Dump</i> =	NA.

Table 4 Error Code: TRKG

Error Code:	TRKG
Error Description:	A non-existent trunk group appears as if it should be having SILC controls applied.
Possible Error Effect:	Space is being wasted in the SILC relation.

Corrective Action Taken:	Tuple is deleted and the NM personnel is notified of the change in the relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SILCTG tuple.
<i>Bad Data</i> =	Trunk group number of non-existent trunk group.
<i>Good Data</i> =	Not used.
<i>Logical Index</i> =	Key of the SILCTG tuple.
<i>Dump</i> =	SILCTG tuple.

SIPCB

Software Release: 5E10 and later

1. SYSTEM INTEGRITY PROTOCOL CONTROL BLOCK (SIPCB) AUDIT

The system integrity protocol control block (SIPCB), CMsipcb, contains channel status information and controls frame ownership for this protocol on the ethernet network.

2. DATA STRUCTURES

The following data is used by the SIPCB audit:

CMsipcb = The SIP control block.
CMetfduse[] = Transmit ethernet frame descriptors.

3. AUDIT CHECKS

The SIPCB audit makes the following checks:

BADLINK = Check of SIPCB frame descriptor ownership.
BDCOUNT = Check of SIPCB frame descriptor count.
INV_TIMER = State/timer consistency check.
NULL3LINK = Consistency check of SIPCB frame descriptor ownership fields.
ZAP = Check for system interrupt handling.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between a system integrity protocol control block (SIPCB) and the ethernet frame descriptors (EFDs) it appears to own is invalid. Specifically, one or more of the following is true: - For the frame descriptor indexed by SIPCB.f_ifd, EFD.I2_prev != CMNIL_NDX. - For the frame descriptor indexed by SIPCB.I_ifd, EFD.I2_next != CMNIL_NDX.
Possible Error Effect:	Frames associated with the SIPCB are not transmitted properly.
Corrective Action Taken:	The audit sets the SIPCB EFD fields (f_ifd , I_ifd , ready , and blocked) to CMNIL_NDX.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.

<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Transmit queue count (SIPCB.xmit_q).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The SIPCB (CMSipcb).

Table 2 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	An SIPCB does not own any EFDs (elements of CMetfduse[]), but has a non-zero transmit queue count (SIPCB.xmit_q != 0).
Possible Error Effect:	SIP frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the transmit queue count (SIPCB.xmit_q = 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The transmit queue count (SIPCB.xmit_q).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 3 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	A SIPCB contains state and/or timer inconsistencies.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit requests that the SIPCB be reset to a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SIPCB.tmr_start.
<i>Bad Data</i> =	SIPCB.tmr_data.
<i>Good Data</i> =	SIPCB.tmr_hbeat.
<i>Logical Index</i> =	SIPCB.state.
<i>Dump</i> =	The SIPCB (CMSipcb).

Table 4 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One or more of an SIPCB's EFD (element of CMetfduse[]) ownership fields (f_ifd , l_ifd , ready , or blocked) contained an out-of-range value (!= CMNIL_NDX and > CM_NUMBUF).
Possible Error Effect:	Frames associated with the SIPCB are not transmitted properly.
Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to CMNIL_NDX. Any lost frame descriptors are picked up by the EFD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Transmit queue count (SIPCB.xmit_q).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	The SIPCB.

Table 5 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit requests that the system integrity protocol (SIP) be reset to a safe state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	In a message handler (MH), this is the MH identifier. In the switching module (SM), it is NA.
<i>Bad Data</i> =	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The SIPCB state (SIPCB.state).
<i>Dump</i> =	The SIPCB (CMsipcb).

SLLOC

Software Release: 5E10 and later

1. SUBSCRIBER LINE AND MEASUREMENT (SLIM) DEVICE (SLLOC) AUDIT

The purpose of this audit is to ensure the integrity of the dynamic rSLLOC tuples which describe the usage and location of SLIM device.

The SLLOC audit runs in both the operational kernel process (OKP) and in the switching module (SM).

2. DATA STRUCTURES

The RLSLLOC relation contains tuples that describe the usage and location of SLIM device. Each tuple contains the identity of the SLIM it describes, the SM the SLIM is located in, the circuit name of the SLIM, SLIM usage, and audit transient information.

The SLLOC audit verifies the validity of the following field in rSLLOC:

USAGE = indicates the usage state of SLIM device.

This audit also uses the rSL_SESSN tuple associated with an active SLLOC tuple to determine that a routine mode SLIM test session is not running outside its scheduled time.

3. AUDIT CHECKS

For each tuple in the RLSLLOC relation, the SLLOC audit makes the following checks:

BADSESSN = Verify that the process the routine mode test session is running under still exists.

BDPCR = Check that the SM number in rSLLOC is an operational SM.

CKTNAME = Verify that the circuit name stored in CDBCOM is valid.

INVSLIMTUP = Verify that a rSLLOC tuple that is not linked to the free list does not have an invalid key.

INVUSAGE = Verify that the usage field is valid.

MSGPTR = Verify that the "slim"nr" found is valid.

VALTIME_BAD = Verify that the time stored in rSL_SESSN associated with rSLLOC tuple is valid.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADSESSN

Error Code:	BADSESSN
Error Description:	An invalid routine mode test session has been detected. The test session is running without an owning process or an rSL_SESSN tuple does not exist for this test session.
Possible Error Effect:	The total number of routine mode test sessions that can be run will be limited

	due to this invalid test session.
Corrective Action Taken:	A message is sent to the routine mode control process instructing it to abort the test session.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISL_SESSN tuple.
<i>Bad Data</i> =	The identity of the test session.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	None.

Table 2 Error Code: BDPCR

Error Code:	BDPCR
Error Description:	An rISLLOC tuple has been found with the number of an SM that is not operational.
Possible Error Effect:	The availability of rISLLOC tuples for SLIM test sessions may be reduced until this invalid rISLLOC tuple is corrected.
Corrective Action Taken:	All fields in the erroneous rISLLOC tuple are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISLLOC tuple.
<i>Bad Data</i> =	The value found in the "sm" field of the rISLLOC tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The value in the "slim_nr" field of the rISLLOC tuple.
<i>Dump</i> =	The rISLLOC tuple.

Table 3 Error Code: CKTNAME

Error Code:	CKTNAME
Error Description:	An rISLLOC tuple has been found which has an associated rCDBCOM tuple with an invalid circuit name.
Possible Error Effect:	The availability of rISLLOC tuples for SLIM test sessions may be reduced until this invalid rISLLOC tuple is corrected.
Corrective Action Taken:	All fields in the erroneous rISLLOC tuple are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISLLOC tuple.
<i>Bad Data</i> =	The circuit name in the rISLLOC tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The value in the "slim_nr" field of the rISLLOC tuple.
<i>Dump</i> =	The rISLLOC tuple.

Table 4 Error Code: INVSLIMTUP

Error Code:	INVSLIMTUP
Error Description:	An rISLLOC tuple has been found with an invalid SLIM circuit name.
Possible Error Effect:	The availability of rISLLOC tuples for SLIM test sessions may be reduced until this invalid rISLLOC tuple is corrected.
Corrective Action Taken:	All fields in the erroneous rISLLOC tuple are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISLLOC tuple.
<i>Bad Data</i> =	The value found in the "usage" field of the rISLLOC tuple.
<i>Good Data</i> =	The value found in the "slim_nr" field of the rISLLOC tuple.
<i>Logical Index</i> =	The index of the rISLLOC tuple.
<i>Dump</i> =	The rISLLOC tuple.

Table 5 Error Code: INVUSAGE

Error Code:	INVUSAGE
Error Description:	An rISLLOC tuple has been found with an associated rCDBCOM tuple that indicates that this SLIM is on an idle list.
Possible Error Effect:	The availability of rISLLOC tuples for SLIM test sessions may be reduced until this invalid rISLLOC tuple is corrected.

Corrective Action Taken:	All fields in the erroneous rISLLOC tuple are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISLLOC tuple.
<i>Bad Data =</i>	The value found in the "usage" field of the rISLLOC tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The value in the "slim_nr" field of the rISLLOC tuple.
<i>Dump =</i>	The rISLLOC tuple.

Table 6 Error Code: MSGPTR

Error Code:	MSGPTR
Error Description:	An rISLLOC tuple has been found with a value in "slim_nr" that does not match the index of the rISLLOC tuple.
Possible Error Effect:	The availability of rISLLOC tuples for SLIM test sessions may be reduced until this invalid rISLLOC tuple is corrected.
Corrective Action Taken:	All fields in the erroneous rISLLOC tuple are zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISLLOC tuple.
<i>Bad Data =</i>	The circuit name in the rISLLOC tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The value in the "slim_nr" field of the rISLLOC tuple.
<i>Dump =</i>	The rISLLOC tuple.

Table 7 Error Code: VALTIME_BAD

Error Code:	VALTIME_BAD
Error Description:	The current time does not match with the date and time a routine mode test session is scheduled for in rISL_SESSN.
Possible Error Effect:	The routine mode test session is running unscheduled.
Corrective Action Taken:	A message is sent to routine mode control process instructing it to abort the invalid test session.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISL_SESSN tuple.
<i>Bad Data =</i>	The number of the SLIM device executing the test session.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The identity of the test session.
<i>Dump 1 =</i>	Current date and time.
<i>Dump 2 =</i>	The rISL_SESSN tuple.

SMEST

Software Release: 5E10 and later

1. SWITCH MAINTENANCE EQUIPMENT STATUS TABLE (SMEST) AUDIT

The maintenance request administrator program (MRA) maintains the status of the hardware in the interface modules and periphery in the switch maintenance equipment status table (rSMEST). The equipment status table contains an entry for every circuit in the switching module and periphery except for the metallic test interconnection buses (MTIBs). The MTIBs are allocated in the administrative module and audited by the administrative module equipment status table audit (CPEST). The equipment status table keeps track of the status (in-service or out-of-service) of the circuits. The SMEST audit performs a dynamic-to-static check with the circuit data relation (RLCKTDATA) to ensure the consistency of the information in rSMEST with the actual hardware states.

The SMEST audit is part of the switch maintenance audits subgroup. It runs in the OSDS-M environment in the segmented modes.

2. DATA STRUCTURES

The SMEST audit verifies the existence of tuples in the switch maintenance equipment status table relation (RLSMEST), a dynamic relation, by checking the static circuit data relation (RLCKTDATA).

3. AUDIT CHECKS

The audit first checks whether RLCKTDATA is being updated by a recent change/verify transaction. If it is being updated, the SMEST audit quits and returns control to the audit control program. If RLCKTDATA is not being updated, the audit invokes the static driver (DRSTATIC) to step through RLSMEST and read the unit ID field of each tuple. Refer to Figure 1 .

The audit compares the unit ID field of RLSMEST to the circuit name field of RLCKTDATA to determine whether a corresponding tuple exists there. If the tuple exists in RLCKTDATA, the audit checks the next tuple in RLSMEST. If the tuple does not exist in RLCKTDATA, an error is flagged.

The audit verifies that tuples having parent circuits contain information that is consistent with the parent circuit. The checks are called vertical checks. Vertical checks are made on modules equipped with a remote clock (RCLK).

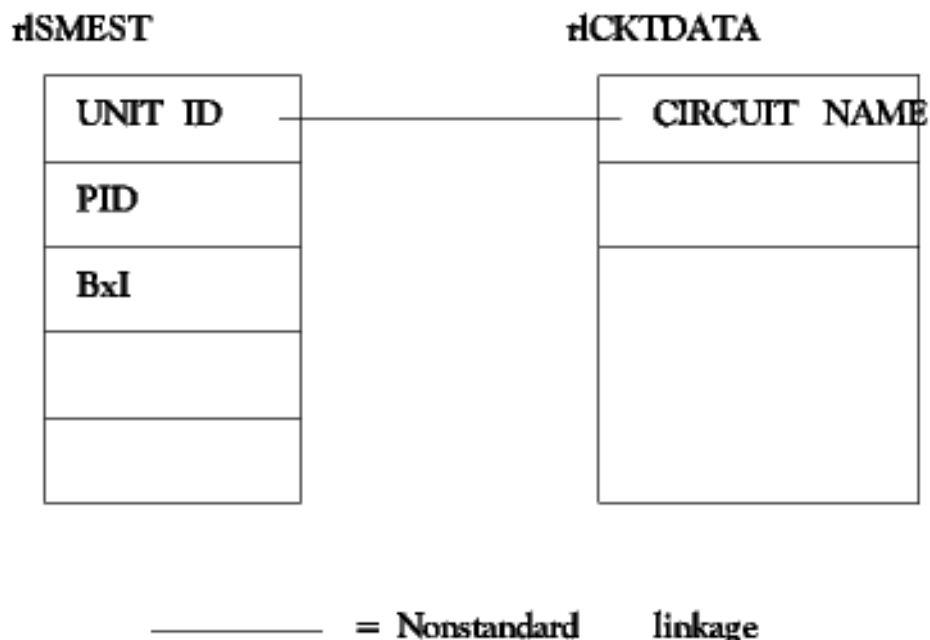


Figure 1 (SMEST-1) Fields Checked by the SMEST Audit

4. ERROR HANDLING

If the audit finds no corresponding tuple in RLCKTDATA for the rSMEST tuple being checked, the audit removes the extra the rSMEST tuple from the database.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra rSMEST tuple exists.
Possible Error Effect:	Eventual loss of call processing resources.
Corrective Action Taken:	Delete the extra rSMEST tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rSMEST tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Internal name of the circuit.
<i>Dump =</i>	rSMEST tuple.

Table 2 Error Code: RC_BADREF

Error Code:	RC_BADREF
Error Description:	The rSMEST tuple for a remote clock reference has an invalid maintenance status when compared with the parent circuit, the remote clock side, whose status is out-of-service (OOS).
Possible Error Effect:	Reference signals being sent to an OOS clock.
Corrective Action Taken:	The reference is reconfigured to agree with the parent circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of reference rSMEST tuple.
<i>Bad Data =</i>	Maintenance status of the reference side.

<i>Good Data</i> =	OOS.
<i>Logical Index</i> =	Logical key of reference rSMEST tuple.
<i>Dump 1</i> =	rSMEST tuple of the remote clock side.
<i>Dump 2</i> =	rSMEST tuple of the reference side.

Table 3 Error Code: RC_CONFIG

Error Code:	RC_CONFIG
Error Description:	Reconfiguration request of a remote clock circuit was not successful.
Possible Error Effect:	Cause of the reconfiguration request remains unresolved.
Corrective Action Taken:	All remote clock circuits generating this error, except the parent circuit, are marked out-of-service or active.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	Return value of the configuration routine.
<i>Good Data</i> =	Action performed on the circuit.
<i>Logical Index</i> =	Logical key or rSMEST tuple configured.
<i>Dump 1</i> =	rSMEST tuple of parent circuit.
<i>Dump 2</i> =	rSMEST tuple of subtending or mate side to a remote clock side.

Table 4 Error Code: RC_GLRCLKUID

Error Code:	RC_GLRCLKUID
Error Description:	A remote clock rSMEST tuple is operational but the unit identification ID global variable was not set.
Possible Error Effect:	Unable to identify the clock side, and audits cannot run. Possible office data problem.
Corrective Action Taken:	None.
Manual Action Required:	Add the remote clock global variable to the office data.
Dump Description:	
<i>Error Address</i> =	Address of remote clock rSMEST tuple.
<i>Bad Data</i> =	Logical key of rSMEST.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rSMEST.
<i>Dump</i> =	NA.

Table 5 Error Code: RC_HDWSFTW

Error Code:	RC_HDWSFTW
Error Description:	While comparing the maintenance status of a remote clock circuit, with the hardware against the software, an inconsistency was identified.
Possible Error Effect:	Improper operation of the remote clock would result.
Corrective Action Taken:	If the circuit checked is the RCLK unit and its maintenance status is active or standby, the rSMEST tuple is reconfigured in order to update the hardware status of its subtending unit - mate cross couple (XC) and references (REF). However, if the clock is out-of-service the hardware maintenance status of the XC and REF are updated to agree with the parent circuit - the RCLK. When the subtending circuits are being checked, corrections are done by updating the hardware with the inverse of the existing data which can be zero or one. The other units, OSC and OXC, get updated based on the rSMEST maintenance status of the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rSMEST tuple.
<i>Bad Data</i> =	0 (OOS), 1 (ACT), 2 (STBY), information in the configuration register.
<i>Good Data</i> =	0 (OOS), 1 (ACT), 2 (STBY).
<i>Logical Index</i> =	Logical key of rSMEST tuple.
<i>Dump</i> =	rSMEST tuple.

Table 6 Error Code: RC_SCNPTERR

Error Code:	RC_SCNPTERR
Error Description:	Unable to read the scan points for the remote clock (RCLK) when attempting to

	determine if power existed.
Possible Error Effect:	Cannot verify if the RCLK is powered up, and the display on the MCC might be inconsistent.
Corrective Action Taken:	None.
Manual Action Required:	Inspect RCLK installation for possible miswiring.
Dump Description:	
<i>Error Address =</i>	Address of rIRCLK_SC buffer.
<i>Bad Data =</i>	Scan point data.
<i>Good Data =</i>	Expected value to one.
<i>Logical Index =</i>	Logical key to access rIRCLK_SC.
<i>Dump =</i>	rIRCLK_SC buffer.

Table 7 Error Code: RC_SUBERR

Error Code:	RC_SUBERR
Error Description:	The rISMEST tuple for a subtending circuit to a remote clock side is in an invalid maintenance status.
Possible Error Effect:	If the subtending circuit is the other clock side, both clock sides may be active, or one side may be out-of-service and the other standby. When the subtending circuit is the mate cross couple (RCXC), the RCXC could cause timing problems to the active clock side.
Corrective Action Taken:	Reconfigure both clock sides if the subtending circuit is the other clock; this is done by busying, inhibiting, and removing both sides then idling, allowing, and restoring one side at a time. If the subtending circuit is the RCXC, its status is set to agree with remote clock on the given side.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of subtending/mate rISMEST tuple.
<i>Bad Data =</i>	Maintenance status of the subtending/mate side.
<i>Good Data =</i>	Maintenance status of the remote clock side. NA if the circuit is the RCXC.
<i>Logical Index =</i>	Logical key of subtending/mate rISMEST tuple.
<i>Dump 1 =</i>	rISMEST tuple of the remote clock side.
<i>Dump 2 =</i>	rISMEST tuple of the subtending/mate side.

Table 8 Error Code: RC_TMBASE

Error Code:	RC_TMBASE
Error Description:	The audit was not able to determine which oscillator it is listening to because the time base register in the communication control buffer (CCB) for a remote clock was found to be out-of-range.
Possible Error Effect:	Loss of resources for the RCLK.
Corrective Action Taken:	None.
Manual Action Required:	It is recommended that the remote clock be reconfigured manually.
Dump Description:	
<i>Error Address =</i>	Address of the rISMEST tuple for the oscillator.
<i>Bad Data =</i>	Local time base register value.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rISMEST tuple for the oscillator.
<i>Dump 1 =</i>	Tuple ofr the RCLK.
<i>Dump 2 =</i>	rISMEST tuple of the oscillator.

Table 9 Error Code: RCOSC_ACT

Error Code:	RCOSC_ACT
Error Description:	The maintenance status for an oscillator (OSC) was found to be invalidly active.
Possible Error Effect:	Improper remote clock operation.
Corrective Action Taken:	The circuit is reconfigured to agree with the configuration of the remote clock and its time base registers - loctb and remtb bits only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RCLK rISMEST tuple.
<i>Bad Data =</i>	Maintenance status of the sub-unit (OXC or OSC).
<i>Good Data =</i>	Maintenance status of the sub-unit.
<i>Logical Index =</i>	Logical key of sub-unit rISMEST tuple.
<i>Dump 1 =</i>	rISMEST tuple of the remote clock side.

<i>Dump 2 =</i>	rSMEST tuple of the sub-unit side.

Table 10 Error Code: RCOSC_OOS

Error Code:	RCOSC_OOS
Error Description:	The maintenance status for an oscillator (OSC) was found to be invalidly out-of-service.
Possible Error Effect:	Improper remote clock operation.
Corrective Action Taken:	The circuit is reconfigured to agree with the configuration of the remote clock and its time base registers - loctb and remtb bits only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RCLK rSMEST tuple.
<i>Bad Data =</i>	Maintenance status of the sub-unit (OXC or OSC).
<i>Good Data =</i>	Maintenance status of the sub-unit.
<i>Logical Index =</i>	Logical key of sub-unit rSMEST tuple.
<i>Dump 1 =</i>	rSMEST tuple of the remote clock side.
<i>Dump 2 =</i>	rSMEST tuple of the sub-unit side.

Table 11 Error Code: RCOSC_STBY

Error Code:	RCOSC_STBY
Error Description:	The maintenance status for an oscillator (OSC) was found to be invalidly stand-by.
Possible Error Effect:	Improper remote clock operation.
Corrective Action Taken:	The circuit is reconfigured to agree with the configuration of the remote clock and its time base registers - loctb and remtb bits only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RCLK rSMEST tuple.
<i>Bad Data =</i>	Maintenance status of the sub-unit (OXC or OSC).
<i>Good Data =</i>	Maintenance status of the sub-unit.
<i>Logical Index =</i>	Logical key of sub-unit rSMEST tuple.
<i>Dump 1 =</i>	rSMEST tuple of the remote clock side.
<i>Dump 2 =</i>	rSMEST tuple of the sub-unit side.

Table 12 Error Code: RCOXC_ACT

Error Code:	RCOXC_ACT
Error Description:	The maintenance status for an oscillator cross couple (OXC) was found to be invalidly active.
Possible Error Effect:	Improper remote clock operation.
Corrective Action Taken:	The circuit is reconfigured to agree with the configuration of the remote clock and its time base registers - loctb and remtb bits only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RCLK rSMEST tuple.
<i>Bad Data =</i>	Maintenance status of the sub-unit (OSC or OSC).
<i>Good Data =</i>	Maintenance status of the sub-unit.
<i>Logical Index =</i>	Logical key of sub-unit rSMEST tuple.
<i>Dump 1 =</i>	rSMEST tuple of the remote clock side.
<i>Dump 2 =</i>	rSMEST tuple of the sub-unit side.

Table 13 Error Code: RCOXC_OOS

Error Code:	RCOXC_OOS
Error Description:	The maintenance status for an oscillator cross couple (OXC) was found to be invalidly out-of-service.
Possible Error Effect:	Improper remote clock operation.
Corrective Action Taken:	The circuit is reconfigured to agree with the configuration of the remote clock and its time base registers - loctb and remtb bits only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RCLK rSMEST tuple.
<i>Bad Data =</i>	Maintenance status of the sub-unit (OXC or OSC).

<i>Good Data</i> =	Maintenance status of the sub-unit.
<i>Logical Index</i> =	Logical key of sub-unit rSMEST tuple.
<i>Dump 1</i> =	rSMEST tuple of the remote clock side.
<i>Dump 2</i> =	rSMEST tuple of the sub-unit side.

Table 14 Error Code: RCOXC_STBY

Error Code:	RCOXC_STBY
Error Description:	The maintenance status for an oscillator cross couple (OXC) was found to be invalidly stand-by.
Possible Error Effect:	Improper remote clock operation.
Corrective Action Taken:	The circuit is reconfigured to agree with the configuration of the remote clock and its time base registers - loctb and remtb bits only.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the RCLK rSMEST tuple.
<i>Bad Data</i> =	Maintenance status of the subunit (OXC or OSC).
<i>Good Data</i> =	Maintenance status of the subunit.
<i>Logical Index</i> =	Logical key of sub-unit rSMEST tuple.
<i>Dump 1</i> =	rSMEST tuple of the remote clock side.
<i>Dump 2</i> =	rSMEST tuple of the sub-unit side.

SMXRF

Software Release: 5E10 and later

1. SWITCH MAINTENANCE CROSS-REFERENCE (SMXRF) AUDIT

The switch maintenance cross-reference table relation (RLSMXRF) contains a list of all the currently active processes controlled by the maintenance request administrator (MRA) program. Each process that is created in the switching module has an entry in the cross-reference table. Up to eight concurrent processes are supported by the switch maintenance subsystem. The switch maintenance cross-reference audit (SMXRF) maintains the integrity of rLSMXRF.

This audit is part of the switch maintenance audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structure checked by the SMXRF audit is the switch maintenance cross-reference table relation (RLSMXRF). The RLSMXRF relation may have up to eight entries, one for each active process. It is a member relation of the database head cells relation (RLDB_HEADC). The RLDB_HEADC relation has one tuple for busy RLSMXRF entries and one tuple for idle RLSMXRF entries. Figure 1 shows these data structures.

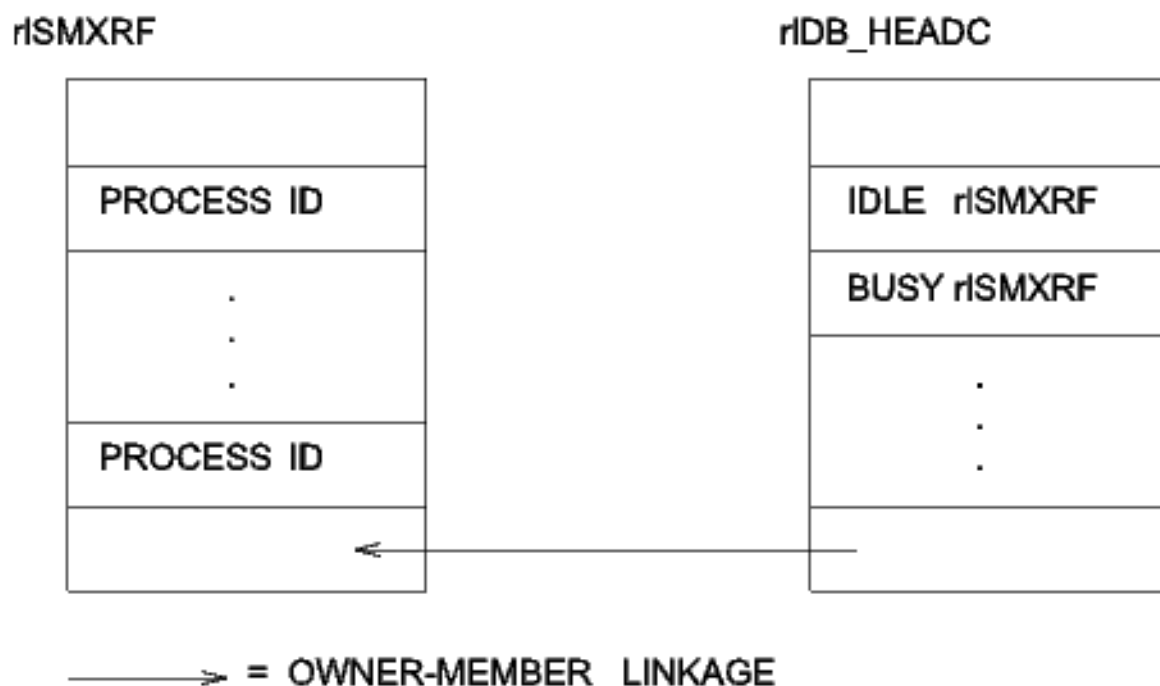


Figure 1 (SMXRF-1) Data Structures of the SMXRF Audit

3. AUDIT CHECKS

To verify the linkages between rLSMXRF tuples and rIDB_HEADC tuples, the audit calls the linkage driver (DRLINK) once for each audit segment. Refer to the Audit Drivers section of this manual.

If all the linkage checks pass, DRLINK invokes the semantic check routine. The semantic check routine checks whether the rLSMXRF tuple is on the idle list. If it is on the idle list, no other checks are made and the audit proceeds to the next rLSMXRF tuple.

If the rSMXRF tuple is not on the idle list, the audit makes the following checks:

- (a) Verifies that the terminal process associated with the rSMXRF tuple being audited is alive.
- (b) Obtains the corresponding rSMEST tuple and verifies that the terminal process ID in the rSMEST tuple matches the terminal process ID in the rSMXRF tuple. If these checks pass, the semantic routine returns to DRLINK to fetch the next rSMXRF tuple. If the tests fail, recovery is done on the rSMXRF tuple.

4. ERROR HANDLING

The audit handles errors by zeroing the appropriate tuples in the cross-reference table and linking them to the idle list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	The pid fields in the rSMXRF and rSMEST tuples do not match.
Possible Error Effect:	MRA may not properly perform maintenance requests for some circuit(s).
Corrective Action Taken:	The rSMXRF tuple is recovered to the idle rSMXRF list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rSMXRF tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rSMXRF tuple in question.
<i>Dump 1 =</i>	rSMXRF tuple contents.
<i>Dump 2 =</i>	rSMEST tuple contents.

Table 2 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	An rSMXRF tuple linked to the active list has a dead process ID (PID) in it.
Possible Error Effect:	MRA may not properly perform maintenance requests for some circuit(s).
Corrective Action Taken:	The rSMXRF tuple is recovered to the idle rSMXRF list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of bad rSMXRF tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rSMXRF tuple in question.
<i>Dump =</i>	rSMXRF tuple contents.

SOCKET-A

Software Release: 5E12 only

1. ETHERNET SOCKET PROTOCOL CONTROL BLOCK (SOCKET) AUDIT

Sockets are the interface to send and receive data using the ethernet network. They are used to represent per-connection information such as state and data buffers. The sockets are either idle and linked to the idle list or linked and linked to an internet protocol control block (INPCB). The audit verifies the linkages among CMsocket[], CMinpcb[], CMesmepipe, CMSipcb and CMesmelnk and verifies data within every entry in CMsocket[].

2. DATA STRUCTURES

The following data are used by the SOCKET audit:

CMsocket[] = Socket protocol control block.

CMinpcb[] = Internet protocol control block.

CMiddlesopcb = Socket control block idle list.

CMesmepipe = MEPIPE dynamic data.

CMSipcb = SIP control block.

CMesmelnk = MELNK dynamic data.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between a SOCKET and the ethernet frame descriptors (EFDs) it appears to own is invalid. Specifically, one or more of the following is true: - CMetfduse[CMsocket[index].so_snd.sb_unld].l2_prev does not equal QLFDNULL - CMetfduse[CMsocket[index].so_snd.sb_unld].bf_owner & CM_L2_OWNER does not equal 0 - CMetfduse[CMsocket[index].so_snd.sb_unld].usetype does not equal CMENXMT/CMENRCV - CMetfduse[CMsocket[index].so_snd.sb_load].l2_next does not equal QLFDNULL - CMetfduse[CMsocket[index].so_snd.sb_load].bf_owner & CM_L2_OWNER does not equal 0

	CMetfduse[CMsocket[index].so_snd.sb_load].usetype does not equal CMENXMT/CMENRCV
Possible Error Effect:	Frames associated with the SOCKET are not transmitted properly.
Corrective Action Taken:	The audit sets CMsocket[index].so_snd.sb_unld and .so_snd.sb_load to QLFDNULL and .so_snd.sb_mbcnt to 0 for SOCKETSND or sets CMsocket[index].so_rcv.sb_unld and .so_rcv.sb_load to QLFDNULL and so_rcv.sb_mbcnt to 0 for SOCKETRCV.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Two for SOCKETSND or three for SOCKETRCV.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	CMetfduse[CMsocket[index].so_snd/rcv.sb_mbcnt.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	CMsocket[index].

Table 2 Error Code: BADPROTOCOL

Error Code:	BADPROTOCOL
Error Description:	The so_systype of a CMsocket[] entry is set to CMSOST_RTCD, theso_protocol is set to CMIPRIP. One of the following has occurred: - CMsocket[index].so_melnk is not 0, - CMesmelnk[0][0].so_idx[0] does not equal the index into CMsocket[] or - CMesmelnk[0][1].so_idx[0] does not equal the index into CMsocket[].
Possible Error Effect:	The MELNK is removed and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_cbstate.
<i>Bad Data =</i>	CMsocket[index].so_systype.
<i>Good Data =</i>	CMsocket[index].so_protocol.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump 1 =</i>	Contents of CMsocket[index].
<i>Dump 2 =</i>	Contents of CMsocket[index].
<i>Dump 3 =</i>	Contents of CMesmelnk[0][0].ping_so_idx.
<i>Dump 4 =</i>	Contents of CMesmelnk[0][1].ping_so_idx.

Table 3 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	This CMsocket[] entry should be linked to an entry in CMinpcb[]. One of the following has occurred: - CMsocket[index].so_qindx is not CMNIL_NDX, - CMsocket[index].so_pcb is out of range or - CMinpcb[CMsocket[index].so_pcb].inp_socket does not equal the index into CMsocket[].
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	CMsocket[index].so_pcb.
<i>Bad Data</i> =	CMsocket[index].so_qindx.
<i>Good Data</i> =	CMinpcb[CMsocket[index].so_pcb].inp_socket.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 4 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The state of an entry in CMsocket[] is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMsocket[index].so_cbstate.
<i>Good Data</i> =	CMINCBMAX
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 5 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The protocol for a CMsocket[] entry whose so_systype is set to CMSOST_RTCD is not recognized by the audit.
Possible Error Effect:	The MELNK is removed.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMsocket[index].so_cbstate.
<i>Bad Data</i> =	CMsocket[index].so_systype.
<i>Good Data</i> =	CMsocket[index].so_protocol.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 6 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	A SOCKET does not own any EFDs (elements of CMetfduse[]), but has a non-zero transmit queue count (SOCKET.xmit_q != 0).
Possible Error Effect:	The SOCKET frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the transmit queue count (SOCKET.xmit_q = 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Two for AUCMSOCKETSND or three for AUCMSOCKETRCV.
<i>Bad Data</i> =	The transmit queue count (SOCKET.xmit_q).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	NA.

Table 7 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The CMsocket[] is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	CMsocket[index].so_cbstate.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 8 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An SOCKET in-use idle stack entry (index $\leq$ idle stack <i>toppt</i>) is corrupt. One or more of the following is true ("idle_socket" refers to the contents of the idle stack entry <i>sopcb[index]</i>): - idle_socket $\geq$ max - The SOCKET indexed the contents of idle stack entry <i>sopcb[idle_socket]</i> and does not point back to the idle list (SOCKET.so_qindx $\neq$ idle_socket).
Possible Error Effect:	SOCKET may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry (index = idle stack <i>toppt</i>), the audit clears the idle stack entry (<i>sopcb[index]</i> = CMNIL_NDX) and decrements the count of idle SOCKETS (<i>toppt</i> is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets <i>sopcb[index]</i> = <i>sopcb[toppt]</i>, relinks (SOCKET.so_qindx = index), clears the old entry (<i>so_pcb[CMidlesopcb.toppt]</i> = CMNIL_NDX), and decrements the number of idle SOCKETS (decrements <i>toppt</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Indicates SOCKET idle list (CMidlesopcb).
<i>Bad Data</i> =	Value from idle stack entry (SOCKET index).
<i>Good Data</i> =	Value of "max".
<i>Logical Index</i> =	Index into idle stack.
<i>Dump</i> =	NA.

Table 9 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMsocket[] idle stack is corrupt. One or more of the following is true: - The idle stack <i>toppt</i> value is out-of-range ($>$ max or $<$ min). - The idle stack entry <i>sopcb[toppt]</i> = CMNIL_NDX. - The idle stack <i>toppt</i> is $<$ max and the idle stack entry <i>sopcb[toppt+1]</i> $\neq$ CMNIL_NDX.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit clears the CMsocket[] idle stack (<i>toppt</i> = 0) and rebuilds the idle list (links all entries whose <i>so_cbstate</i> is CMINCBFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMidlesopcb.sopcb[CMidlesopcb.toppt]
<i>Bad Data</i> =	CMidlesopcb.sopcb[CMidlesopcb.toppt+1].
<i>Good Data</i> =	CM_NUMSOPCB.
<i>Logical Index</i> =	CMidlesopcb.toppt

<i>Dump</i> =	NA.

Table 10 Error Code: INDEX

Error Code:	INDEX
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is set to CMSOST_VCDX. One of the following has occurred: - CMsocket[index].so_melnk does not equal (DMELNKNUM)-1. - CMsocket[index].so_protocol does not equal CMIPUDP, - CMsipcb.sip_soindx[0].sip_sndso does not equal the index into CMsocket[], - CMsipcb.sip_soindx[0].sip_rcvso does not equal the index into CMsocket[], - CMsipcb.sip_soindx[1].sip_sndso does not equal the index into CMsocket[] or - CMsipcb.sip_soindx[1].sip_rcvso does not equal the index into CMsocket[].
Possible Error Effect:	Ethernet link to workstation is restarted. Transient calls needing workstation may fail.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMsocket[index].so_cbstate.
<i>Bad Data</i> =	CMsocket[index].so_systype.
<i>Good Data</i> =	CMsocket[index].so_protocol.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump 1</i> =	Contents of CMsocket[index].
<i>Dump 2</i> =	Contents of CMsipcb.

Table 11 Error Code: INV

Error Code:	INV
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is not recognized by the audit.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMsocket[index].so_systype.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 12 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The state of a CMsocket[] is one not recognized by the audit.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle

	stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMsocket[index].so_cbstate.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 13 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused SOCKET idle stack entry is not null (index < <i>toppt</i> and <i>sopcb[index]</i> != CMNIL_NDX).
Possible Error Effect:	SOCKETs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the SOCKET idle stack entry (<i>sopcb[index]</i> = CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Indicate SOCKET idle list (CMidlesopcb).
<i>Bad Data =</i>	Contents of the idle stack entry (<i>sopcb[index]</i>).
<i>Good Data =</i>	CMNIL_NDX.
<i>Logical Index =</i>	Index into the idle stack.
<i>Dump =</i>	NA.

Table 14 Error Code: MARKED_INDEX

Error Code:	MARKED_INDEX
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is set to CMSOST_RTCD and the <i>so_protocol</i> is set to CMIPTCP. One of the following has occurred: - CMsocket[index].so_melnk does not equal 0, - CMesmelnk[0][0].so_idx[0] does not equal the index into CMsocket[] or - CMesmelnk[0][1].so_idx[0] does not equal the index into CMsocket[].
Possible Error Effect:	The MELNK is removed.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_cbstate.
<i>Bad Data =</i>	CMsocket[index].so_systype.
<i>Good Data =</i>	CMsocket[index].so_protocol.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump 1 =</i>	Contents of CMsocket[index].
<i>Dump 2 =</i>	Contents of CMesmelnk[0][0].so_idx[0].
<i>Dump 3 =</i>	Contents of CMesmelnk[0][1].so_idx[0].

Table 15 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	One of the fields in an idle CMsocket[] entry is not its default value.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMidlesopcb.sopcb[CMsocket[so_qindx]].
<i>Good Data =</i>	CMsocket[index].so_cbstate.

<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 16 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One or more of an MHATPL3Q's EFD (element of CMetfduse[]) ownership fields (<i>f_ifd</i> , <i>l_ifd</i> , <i>ready</i> , or <i>blocked</i>) contained an out-of-range value (!= CMNIL_NDX and > CM_NUMBUF).
Possible Error Effect:	Frames associated with the SOCKET are not transmitted properly.
Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to CMNIL_NDX. Any lost frame descriptors are picked up by the EFD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Two for AUCMSOCKETSND or three for AUCMSOCKETRCV.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Transmit queue count (SOCKET.xmit_q).
<i>Logical Index</i> =	Index into CMmhatpl3q[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 17 Error Code: TYPE

Error Code:	TYPE
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is set to CMSOST_HUB. One of the following has occurred: - CMsocket[index].so_melnk does not equal SMELNUMELNKS, - CMsocket[index].so_protocol does not equal CMIPRIP, - CMesmepipe.ping_so_idx[0] does not equal the index into CMsocket[] or - CMesmepipe.ping_so_idx[1] does not equal the index into CMsocket[].
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMsocket[index].so_cbstate.
<i>Bad Data</i> =	CMsocket[index].so_systype.
<i>Good Data</i> =	CMsocket[index].so_protocol.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump 1</i> =	Contents of CMsocket[index].
<i>Dump 2</i> =	Contents of CMesmepipe.ping_so_idx[0].
<i>Dump 3</i> =	Contents of CMesmepipe.ping_so_idx[1].

Table 18 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: - Clears the idle stack entry, if any, that was being audited when the stimulus occurred. - Recovers the CMsocket[] entry, if any, which was being audited when

	the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMidlesopcb.sopcb[CM_NUMSOPCB - AUcur_audstat.position].
<i>Good Data</i> =	CM_NUMSOPCB - AUcur_audstat.position.
<i>Logical Index</i> =	AUcur_audstat.position
<i>Dump</i> =	Contents of CMsocket[AUcur_audstat.position].

SOCKET-B

Software Release: 5E13 and later

1. ETHERNET SOCKET PROTOCOL CONTROL BLOCK (SOCKET) AUDIT

Sockets are the interface to send and receive data using the ethernet network. They are used to represent per-connection information such as state and data buffers. The sockets are either idle and linked to the idle list or linked and linked to an internet protocol control block (INPCB). The audit verifies the linkages among CMsocket[], CMinpcb[], CMesmepipe, CMSipcb and CMesmelnk and verifies data within every entry in CMsocket[].

2. DATA STRUCTURES

The following data are used by the SOCKET audit:

CMsocket[] = Socket protocol control block.

CMinpcb[] = Internet protocol control block.

CMidlesopcb = Socket control block idle list.

CMesmepipe = MEPIPE dynamic data.

CMSipcb = SIP control block.

CMesmelnk = MELNK dynamic data.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The linkage between a SOCKET and the ethernet frame descriptors (EFDs) it appears to own is invalid. Specifically, one or more of the following is true: - CMetfduse[CMsocket[index].so_snd.sb_unld].l2_prev does not equal QLFDNULL - CMetfduse[CMsocket[index].so_snd.sb_unld].bf_owner & CM_L2_OWNER does not equal 0 - CMetfduse[CMsocket[index].so_snd.sb_unld].usetype does not equal CMENXMT/CMENRCV - CMetfduse[CMsocket[index].so_snd.sb_load].l2_next does not equal QLFDNULL - CMetfduse[CMsocket[index].so_snd.sb_load].bf_owner & CM_L2_OWNER does not equal 0

	CMetfduse[CMsocket[index].so_snd.sb_load].usetype does not equal CMENXMT/CMENRCV
Possible Error Effect:	Frames associated with the SOCKET are not transmitted properly.
Corrective Action Taken:	The audit sets CMsocket[index].so_snd.sb_unld and .so_snd.sb_load to QLFNULL and .so_snd.sb_mbcnt to 0 for SOCKETSND or sets CMsocket[index].so_rcv.sb_unld and .so_rcv.sb_load to QLFNULL and so_rcv.sb_mbcnt to 0 for SOCKETRCV.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Two for SOCKETSND or three for SOCKETRCV.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	CMetfduse[CMsocket[index].so_snd/rcv.sb_mbcnt.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	CMsocket[index].

Table 2 Error Code: BADPROTOCOL

Error Code:	BADPROTOCOL
Error Description:	The so_systype of a CMsocket[] entry is set to CMSOST_RTCD or CMSOST_EEC, theso_protocol is set to CMIPRIP and one or more of the following is true: - CMsocket[index].so_melnk is not in range ($0 \leq \text{melnk} < \text{SMELNUMELNK}$ is in-range). - CMesmelnk[so_melnk][0].so_idx[0] does not equal the index into CMsocket[] or - CMesmelnk[so_melnk][1].so_idx[0] does not equal the index into CMsocket[].
Possible Error Effect:	The MELNK is removed and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_cbstate.
<i>Bad Data =</i>	CMsocket[index].so_systype.
<i>Good Data =</i>	CMsocket[index].so_protocol.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump 1 =</i>	Contents of CMsocket[index].
<i>Dump 2 =</i>	Contents of CMsocket[index].
<i>Dump 3 =</i>	Contents of CMesmelnk[so_melnk][0].ping_so_idx if so_melnk is in range. Otherwise, NA.
<i>Dump 4 =</i>	Contents of CMesmelnk[so_melnk][1].ping_so_idx if so_melnk is in range. Otherwise, NA.

Table 3 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	This CMsocket[] entry should be linked to an entry in CMinpcb[]. One of the following has occurred: - CMsocket[index].so_qindx is not CMNIL_NDX, - CMsocket[index].so_pcb is out of range or - CMinpcb[CMsocket[index].so_pcb].inp_socket does not equal the index into CMsocket[].

Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_pcb.
<i>Bad Data =</i>	CMsocket[index].so_qindx.
<i>Good Data =</i>	CMinpcb[CMsocket[index].so_pcb].inp_socket.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 4 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The state of an entry in CMsocket[] is out of range.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMsocket[index].so_cbstate.
<i>Good Data =</i>	CMINCBMAX
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 5 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The protocol for a CMsocket[] entry whose so_systype is set to CMSOST_RTCD is not recognized by the audit.
Possible Error Effect:	The MELNK is removed.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_cbstate.
<i>Bad Data =</i>	CMsocket[index].so_systype.
<i>Good Data =</i>	CMsocket[index].so_protocol.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 6 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	A SOCKET does not own any EFDs (elements of CMetfduse[]), but has a non-zero transmit queue count (SOCKET.xmit_q != 0).
Possible Error Effect:	The SOCKET frame handling code may take incorrect actions.
Corrective Action Taken:	The audit clears the transmit queue count (SOCKET.xmit_q = 0).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Two for AUCMSOCKETSND or three for AUCMSOCKETRCV.
<i>Bad Data =</i>	The transmit queue count (SOCKET.xmit_q).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	NA.

Table 7 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The CMsocket[] is allocated but not linked. Application code allocated the entry

	but was not able to complete linking it.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	CMsocket[index].so_cbstate.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 8 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An SOCKET in-use idle stack entry ($\text{index} \leq \text{idle stack toppt}$) is corrupt. One or more of the following is true ("idle_socket" refers to the contents of the idle stack entry <i>sopcb[index]</i>): - idle_socket $\geq$ max - The SOCKET indexed the contents of idle stack entry <i>sopcb[idle_socket]</i> and does not point back to the idle list (SOCKET.so_qindx $\neq$ idle_socket).
Possible Error Effect:	SOCKET may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry ($\text{index} = \text{idle stack toppt}$), the audit clears the idle stack entry (<i>sopcb[index]</i> = CMNIL_NDX) and decrements the count of idle SOCKETS (<i>toppt</i> is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets <i>sopcb[index]</i> = <i>sopcb[toppt]</i> , relinks (SOCKET.so_qindx = index), clears the old entry (<i>so_pcb[CMidlesopcb.toppt]</i> = CMNIL_NDX), and decrements the number of idle SOCKETS (decrements <i>toppt</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Indicates SOCKET idle list (CMidlesopcb).
<i>Bad Data</i> =	Value from idle stack entry (SOCKET index).
<i>Good Data</i> =	Value of "max".
<i>Logical Index</i> =	Index into idle stack.
<i>Dump</i> =	NA.

Table 9 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMsocket[] idle stack is corrupt. One or more of the following is true: - The idle stack <i>toppt</i> value is out-of-range ($> \text{max}$ or $< \text{min}$). - The idle stack entry <i>sopcb[toppt]</i> = CMNIL_NDX. - The idle stack <i>toppt</i> is $< \text{max}$ and the idle stack entry <i>sopcb[toppt+1]</i> $\neq$ CMNIL_NDX.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit clears the CMsocket[] idle stack (<i>toppt</i> = 0) and rebuilds the idle list

	(links all entries whose <i>so_cbstate</i> is CMINCBSFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMidlesopcb.sopcb[CMidlesopcb.toppt]
<i>Bad Data</i> =	CMidlesopcb.sopcb[CMidlesopcb.toppt+1].
<i>Good Data</i> =	CM_NUMSOPCB.
<i>Logical Index</i> =	CMidlesopcb.toppt
<i>Dump</i> =	NA.

Table 10 Error Code: INDEX

Error Code:	INDEX
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is set to CMSOST_VCDX. One of the following has occurred: - CMsocket[index].so_melnk does not equal (DMELNKNUM)-1. - CMsocket[index].so_protocol does not equal CMIPUDP, - CMsipcb.sip_soindx[0].sip_sndso does not equal the index into CMsocket[], - CMsipcb.sip_soindx[0].sip_rcvso does not equal the index into CMsocket[], - CMsipcb.sip_soindx[1].sip_sndso does not equal the index into CMsocket[] or - CMsipcb.sip_soindx[1].sip_rcvso does not equal the index into CMsocket[].
Possible Error Effect:	Ethernet link to workstation is restarted. Transient calls needing workstation may fail.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMsocket[index].so_cbstate.
<i>Bad Data</i> =	CMsocket[index].so_systype.
<i>Good Data</i> =	CMsocket[index].so_protocol.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump 1</i> =	Contents of CMsocket[index].
<i>Dump 2</i> =	Contents of CMsipcb.

Table 11 Error Code: INV

Error Code:	INV
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is not recognized by the audit.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMsocket[index].so_systype.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into CMsocket[].
<i>Dump</i> =	Contents of CMsocket[index].

Table 12 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The state of a C_Msocket[] is one not recognized by the audit.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMsocket[index].so_cbstate.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 13 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused SOCKET idle stack entry is not null (<i>index < topp</i> and <i>sopcb[index] != CMNIL_NDX</i>).
Possible Error Effect:	SOCKETs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the SOCKET idle stack entry (<i>sopcb[index] = CMNIL_NDX</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Indicate SOCKET idle list (CMidlesopcb).
<i>Bad Data =</i>	Contents of the idle stack entry (<i>sopcb[index]</i>).
<i>Good Data =</i>	CMNIL_NDX.
<i>Logical Index =</i>	Index into the idle stack.
<i>Dump =</i>	NA.

Table 14 Error Code: MARKED_INDEX

Error Code:	MARKED_INDEX
Error Description:	The <i>so_systype</i> for a C_Msocket[] entry is set to CMSOST_RTCD or CMSOST_EEC and the <i>so_protocol</i> is set to CMIPTCP and one or more of the following is true: - CMsocket[index].so_melnk is not in range ($0 \leq \text{melnk} < \text{SMELNUMELNK}$ is in-range). - CMesmelnk[so_melnk][0].so_idx[0] does not equal the index into CMsocket[] or - CMesmelnk[so_melnk][1].so_idx[0] does not equal the index into CMsocket[].
Possible Error Effect:	The MELNK is removed.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_cbstate.
<i>Bad Data =</i>	CMsocket[index].so_systype.
<i>Good Data =</i>	CMsocket[index].so_protocol.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump 1 =</i>	Contents of CMsocket[index].
<i>Dump 2 =</i>	Contents of CMesmelnk[so_melnk][0].so_idx[0] if <i>so_melnk</i> is in range. Otherwise, NA.
<i>Dump 3 =</i>	is in range. Otherwise, NA.

Table 15 Error Code: NON_NULL

Error Code:	NON_NULL

Error Description:	One of the fields in an idle CMsocket[] entry is not its default value.
Possible Error Effect:	For VCDX, ethernet link to workstation is restricted, transient calls needing workstation may fail. For RTCD, MELNK is restarted and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMidlesopcb.sopcb[CMsocket[so_qindx]].
<i>Good Data =</i>	CMsocket[index].so_cbstate.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 16 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	One or more of an MHATPL3Q's EFD (element of CMetfduse[]) ownership fields (<i>f_ifd</i> , <i>l_ifd</i> , <i>ready</i> , or <i>blocked</i>) contained an out-of-range value (!= CMNIL_NDX and > CM_NUMBUF).
Possible Error Effect:	Frames associated with the SOCKET are not transmitted properly.
Corrective Action Taken:	The audit sets all of the frame descriptor ownership fields (listed in the error description) to CMNIL_NDX. Any lost frame descriptors are picked up by the EFD audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Two for AUCMSOCKETSND or three for AUCMSOCKETRCV.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Transmit queue count (SOCKET.xmit_q).
<i>Logical Index =</i>	Index into CMmhatpl3q[].
<i>Dump =</i>	Contents of CMsocket[index].

Table 17 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An entry in CMsocket[] which is in use for RTCD (CMSOCKET.so_systype = CMSOST_RTCD) contains an <i>so_melnk</i> value other than 0.
Possible Error Effect:	Frames associated with the SOCKET are not transmitted properly or the CMsocket[] entry is not available for use when it should be.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMSOCKET.so_systype.
<i>Bad Data =</i>	CMSOCKET.so_melnk.
<i>Good Data =</i>	CMSOCKET.so_protocol.
<i>Logical Index =</i>	Index into CMsocket[].
<i>Dump 1 =</i>	CMsocket[index].
<i>Dump 2 =</i>	CMsipcb.

Table 18 Error Code: TYPE

Error Code:	TYPE
Error Description:	The <i>so_systype</i> for a CMsocket[] entry is set to CMSOST_HUB. One of the following has occurred: - CMsocket[index].so_melnk does not equal SMELNUMELNKS, - CMsocket[index].so_protocol does not equal CMIPRIP, - CMesmepipe.ping_so_indx[0] does not equal the index into CMsocket[] or

	CMesmepipe.ping_so_idx[1] does not equal the index into CMssocket[].
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidlesopcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMsocket[index].so_cbstate.
<i>Bad Data =</i>	CMsocket[index].so_systype.
<i>Good Data =</i>	CMsocket[index].so_protocol.
<i>Logical Index =</i>	Index into CMssocket[].
<i>Dump 1 =</i>	Contents of CMssocket[index].
<i>Dump 2 =</i>	Contents of CMesmepipe.ping_so_idx[0].
<i>Dump 3 =</i>	Contents of CMesmepipe.ping_so_idx[1].

Table 19 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: - Clears the idle stack entry, if any, that was being audited when the stimulus occurred. - Recovers the CMssocket[] entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMidlesopcb.sopcb[CM_NUMSOPCB - AUcur_audstat.position].
<i>Good Data =</i>	CM_NUMSOPCB - AUcur_audstat.position.
<i>Logical Index =</i>	AUcur_audstat.position
<i>Dump =</i>	Contents of CMssocket[AUcur_audstat.position].

SPB

Software Release: 5E10 and later

1. SUPPLEMENTARY PROTOCOL BLOCK (SPB) AUDIT

The operator services position system (OSPS) message transport feature provides reliable message communication between OSPS directory assistance (DA) feature-level software in the integrated services digital network (ISDN) switching module processor (SMP) and OSPS terminal equipment. Two audits are needed to maintain the integrity of the dynamic data in the switching modules (SMs) for the network side, the packets buffers audit (PKB) and the supplementary protocol block audit (SPB).

The SPB audit is part of the OSPS audits subgroup. It is part of the OSPS SM directory assistance automatic call handling (OPSMDAAC) loadable package. It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The audit uses the following data structures: the dynamic supplementary protocol block relation (RLSPB), the dynamic packet buffers relation (RLPKB), the local port firmware relation (RLLPRT_FW), the D-channel application linkage block relation (RLDALB), and the database head cells relation (RLDB_HEADC). The RLDB_HEADC relation contains head cells for idle rISPB tuples and rISPB tuples linked to a timer. Idle rISPB tuples should be linked to the idle list. Tuples of rISPB that are not idle may or may not be linked to the list of timers; they must be linked to rIDALB. A tuple on the timer list means that it is waiting for a message associated with a digital subscriber line (DSL). Figure 1 shows these data structures and their linkages.

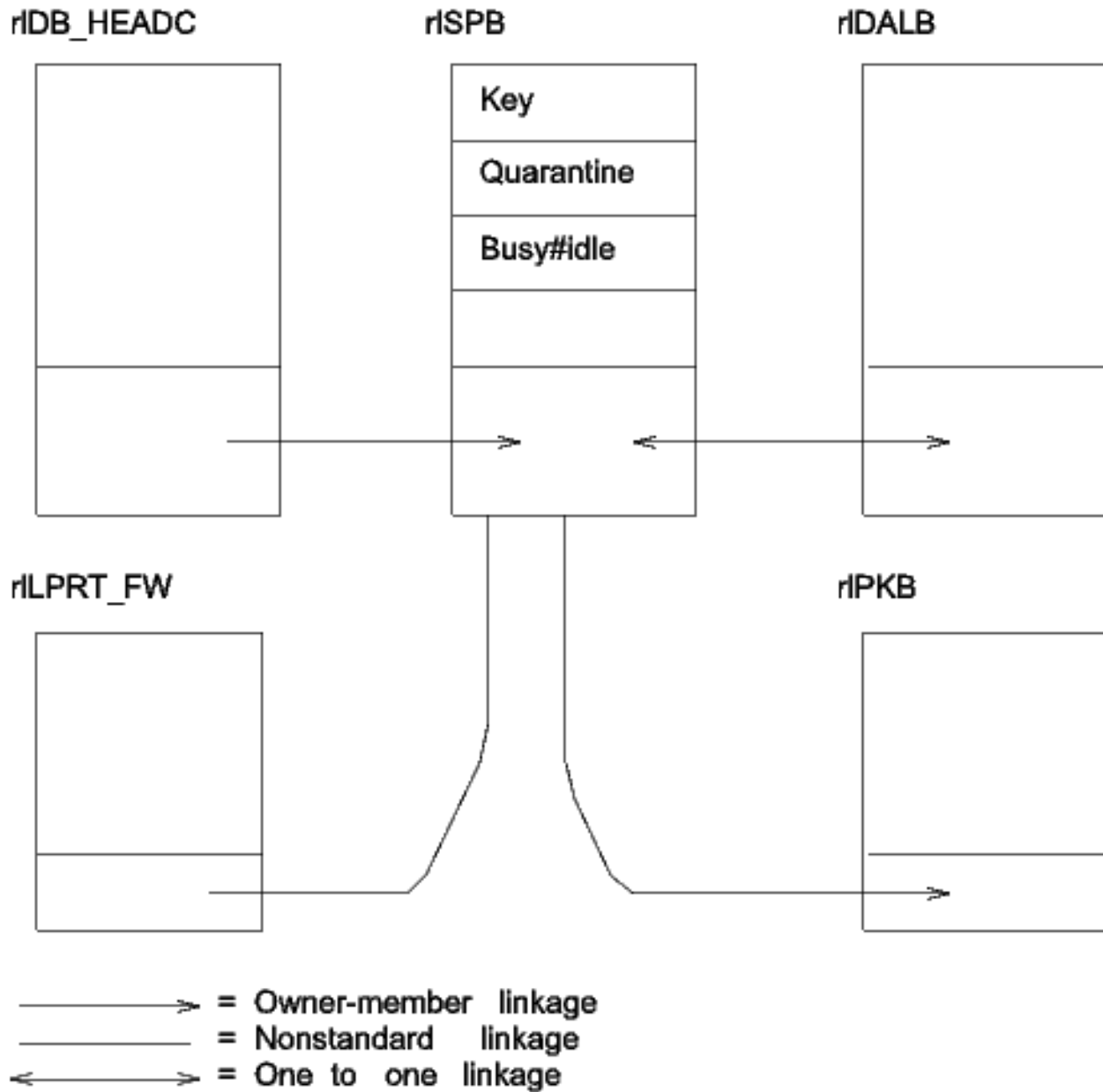


Figure 1 (SPB-1) Data Structures of the SPB Audit

3. AUDIT CHECKS

The SPB audit uses the DRINASEK driver to step through the tuples of RLSPB. The audit checks the point-to/point-back linkages between rISPB and rIDALB, rISPB and rIDB_HEADC, and rISPB and rIPKB. It also examines the quarantine and key fields. If the linkage checks pass, the audit calls its semantic check function to make the following checks.

- When the busy/idle state is idle, the tuple must be owned by the idle head cell in rIDB_HEADC.
- When the tuple is busy, it must be linked to a tuple of rIDALB and the rIDALB tuple must be to a rIDPB tuple.
- If the rISPB tuple is on the idle list, the DSL is not in service. If the rIDALB tuple is linked to an rISPB tuple that

is on the idle list, there is an error that indicates the need for a level three restart.

- (d) If the `busy_idle` field is set to `OATIMER`, the tuple should be linked to the timer list.
- (e) If the `busy_idle` field is set to something other than `busy`, `idle`, or `timer`, the audit will report an error.
- (f) If the logical channel state field (`lc_state`) is set to an invalid state, the audit will report an error.
- (g) The audit will check for inconsistencies in the `servclass` (service class) and `fwcli` attributes (firmware change level indicator) between `rISPB` and `rLPRT_FW`.
- (h) The audit will report an error if the packet layer entity state (`plestate`) is set to an invalid state.
- (i) The audit checks for valid data in the `wind_sz` (window size) and `numpvcs` (number of permanent virtual circuits) fields.
- (j) The audit counts the number of `rIPKB` tuples linked to `rISPB` tuple and verifies that this count matches the value in the `pktsqd` (packets queued) attribute of `rISPB`.

4. ERROR HANDLING

When the audit detects errors, it prints the contents of `rISPB` and initializes the bad tuple.

The SPB audits schedules the `DALB`, `PKB`, and `DBHEADC` audits as related audits when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The number of transmit counter <code>rIPKB</code> tuples linked to this <code>rISPB</code> owning tuple is inconsistent with the value marked in the <code>rISPB</code> transmit counter (<code>rISPB.pktsqd</code>) field. The transmit counter should contain a value equal to the number of transmit <code>rIPKB</code> tuples linked to the <code>rISPB</code> tuple under audit.
Possible Error Effect:	Interruption of Level 3 message traffic, and incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The value in the <code>pktsqd</code> field is corrected to the number of transmit counter <code>rIPKB</code> tuples the audit found linked to this owning <code>rISPB</code> tuple, and a reset of a B X.25 level 3 logical channel is performed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the <code>rISPB</code> tuple.
<i>Bad Data</i> =	The contents of the transmit counter which was found in error (<code>rISPB.pktsqd</code>).
<i>Good Data</i> =	The number of transmit counter <code>rIPKB</code> tuples found linked to the owning <code>rISPB</code> tuple.
<i>Logical Index</i> =	The key of the <code>rISPB</code> tuple (<code>rISPB.spbkey</code>).
<i>Dump</i> =	The contents of the <code>rISPB</code> tuple.

Table 2 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	A busy <code>rISPB</code> tuple (<code>rISPB.busy_idle == OABUSY, OATIMER, OR OAITIMER</code>) did not have an associated <code>rDALB</code> tuple linked to it. Busy <code>rISPB</code> tuple must have an associated <code>rDALB</code> tuple linked such that <code>rISPB.spb_dalb.link - DBSFDALB == rDALB.dalbkey</code> .

Possible Error Effect:	Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The rISPB tuple is recovered to an idle condition, and the DALB audit is scheduled as a related audit to recover any associated rIDALB tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISPB tuple.
<i>Bad Data =</i>	If the linkage to RLDALB is zero, the rISPB tuple's linkage to RLDALB (rISPB.spb_dalb.link - DBSFDALB) is dumped. Otherwise, the key of the rIDALB tuple the audit attempted to access is dumped.
<i>Good Data =</i>	If the linkage to RLDALB is zero, the contents of the <i>busy_idle</i> field and the value the linkage to RLDALB is recovered to (rISPB.busy_idle << 16 DALNKNUL) ID dumped. Otherwise, the contents of the RLDALB linkage and the contents of the <i>busy_idle</i> field is dumped (rISPB.spb_dalb.link - DBSFDALB << 16 rISPB.busy_idle).
<i>Logical Index =</i>	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump =</i>	The contents of the rISPB tuple.

Table 3 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The link from the rISPB tuple to the rIDALB tuple is intact, but the audit cannot get from the rIDALB tuple to the rIDPB tuple using the linkage marked in rIDALB. A relationship for busy rISPB tuple should exist where rIDPB.port == rIDALB.adpm.st_ownk - DBSFDPB where rISPB.spb_dalb.link - DBSFDALB == rIDALB.dalbkey.
Possible Error Effect:	Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The rISPB tuple is recovered to an idle condition, and the DALB and DPB audit are scheduled as related audits to recover any additional data in the RLDALB and RLDPB relations as needed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIDALB tuple that should have been linked to a rIDPB tuple.
<i>Bad Data =</i>	The invalid key of the rIDPB tuple the audit accessed (rIDPB.port).
<i>Good Data =</i>	The key of the rIDPB tuple marked in RLDALB which the audit tried to access, and the contents of the <i>busy_idle</i> field in the rISPB tuple (rIDALB.dadpm.st_ownk - DBSFDPB << 16 rISPB.busy_idle).
<i>Logical Index =</i>	The key of the rISPB tuple.
<i>Dump 1 =</i>	The contents of the rISPB tuple.
<i>Dump 2 =</i>	The contents of the related rIDALB tuple.

Table 4 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The service class (rISPB.servclass) or the firmware change level indicator (rISPB.fwcli) attributes in the rISPB tuple are not equal to their static counterparts in the rLPRT_FW tuple. These fields should be equal, where rLPRT_FW.ldport (key) == rIDPB.port (key), where rIDALB.dadpm.st_ownk - DBSFDPB == rIDPB.port, and rIDALB.dalbkey (key) == rISPB.spb_dalb.link - DBSFDALB.
Possible Error Effect:	Level three messages cannot be processed. Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The rISPB tuple will be recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISPB tuple.
<i>Bad Data =</i>	The service class and the firmware channel level indicator as listed in the rISPB tuple (rISPB.servclass << 16 rISPB.fwcli).
<i>Good Data =</i>	The service class and the firmware chane level indicator as listed in the rLPRT_FW static tuple (rLPRT_FW.servclass << 16 rLPRT_FW.fwcli).

<i>Logical Index</i> =	The key of the rISPB tuple.
<i>Dump 1</i> =	The contents of the rISPB tuple.
<i>Dump 2</i> =	The contents of the rLPRT_FW static tuple.

Table 5 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	The <i>busy_idle</i> indicator in the rISPB tuple is in an invalid state. This field must be marked to the appropriate idle or busy status of the rISPB tuple.
Possible Error Effect:	Any associated calls, as well as the rISPB tuple, will be lost to the system.
Corrective Action Taken:	Any associated calls will be torn down and the corresponding rISPB tuple will be recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The contents of the <i>busy_idle</i> field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 6 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A rISPB tuple is marked busy (rISPB.busy_idle == OABUSY) but its idle list head cell linkage is being used. Busy rISPB tuple should not have their member linkages (rISPB.dasphcm.st_ownk) used when the tuple are busy.
Possible Error Effect:	Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	Any associated calls are torn down and the corresponding rISPB is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISPB tuple.
<i>Bad Data</i> =	The member linkage field contents (rISPB.dasphcm.st_ownk - DBSFDB_HEA).
<i>Good Data</i> =	The contents of the <i>busy_idle</i> flag and the idle list head cell key for the rIDB_HEADC head cell tuple (rISPB.busy_idle << 16 DBSPBIDLE).
<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 7 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rISPB tuple is marked idle but is not linked to the idle list head cell tuple in the RLDB_HEADC relation. For rISPB tuples which are marked idle (rISPB.busy_idle == OAIIDLE), a linkage should exist to the rIDB_HEADC idle list head cell tuple (rISPB.dasphcm.st_ownk - DBSFDB_HEA == DBSPBIDLE).
Possible Error Effect:	Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	Any associated calls are torn down and the corresponding rISPB tuple is recovered to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISPB tuple.
<i>Bad Data</i> =	Contents of the member linkage to RLDB_HEADC (rISPB.dasphcm.st_ownk - DBSFDB_HEA).
<i>Good Data</i> =	Expected value of the member linkage (DBSPBIDLE).
<i>Logical Index</i> =	Key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	Contents of the rISPB tuple found in error.

Table 8 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The static rLPRT_FW tuple does not exist for the dynamic rISPB tuple. A static rLPRT_FW tuple should exist where rLPRT_FW.Idport (key) == rIDPB.port

	(key), where rIDALB.dadpm.st_ownk - DBSFDPB == rIDPB.port, and rIDALB.dalbkey (key) == rISPB.spb_dalb.link - DBSFDALB.
Possible Error Effect:	Level three messages cannot be processed. Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	None.
Manual Action Required:	(1) Using ODBE, read the corresponding rIFC_PORTTYP static tuple using the key of the rIDPB tuple dumped as part of the third dump of this message, and the SM number of the SM where this audit error appeared (rIFC_PORTTYP.port.member == rIDPB.port rIFC_PORTTYP.port.module == GI_pcrd). Log the value of the <i>servclass</i> field in the rIFC_PORTTYP tuple. (2) Insert the missing rLLPRT_FW tuple using the key of the rIDPB tuple dumped as part of the third dump of this message (rIDPB.port) as the key of the rLLPRT_FW tuple (rLLPRT_FW.lidport). This should be the same as the port that was used to read RLFC_PORTTYP. Set the <i>servclass</i> in RLLPRT_FW to the value obtained by reading RLFC_PORTTYP in step 1, above. Set all other values to their defaults.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The key of the rIDPB tuple (rIDPB.port).
<i>Good Data</i> =	The relation ID of the static tuple (RLLPRT_FW).
<i>Logical Index</i> =	The key of the rISPB tuple.
<i>Dump 1</i> =	The contents of the rISPB tuple.
<i>Dump 2</i> =	The contents of the related rIDALB tuple.
<i>Dump 3</i> =	The contents of the related rIDPB tuple.

Table 9 Error Code: PLE_STATE

Error Code:	PLE_STATE
Error Description:	The packet layer entity (PLE) state (rISPB.plestate) marked in the rISPB tuple is incorrect. If the rISPB tuple is not being used for interflow pseudo operator sending OSPS of EDSL, and not being used for an interflow originating treatment process receiving OSPS of EDSL, (rISPB.servclass != OA_IFOPER AND rISPB.servclass != OA_IFTERM), then the PLE state should be marked packet layer entity unavailable (PSL3UNAVAIL), packet layer ready (PSL3R1STATE) or restart requested (PSL3R3STATE).
Possible Error Effect:	Interruption of Level 3 message traffic, and incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The corresponding rISPB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The rISPB tuple's <i>plestate</i> field.
<i>Good Data</i> =	The service class marked in the rISPB tuple (rISPB.servclass).
<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 10 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The <i>lc_state</i> (logical channel state) field in the rISPB tuple has an invalid value. For busy rISPB tuples, this field must be marked PSLCUNAV, PSLCD1, OR PSLCD2.
Possible Error Effect:	Level three messages cannot be processed. Incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The rISPB tuple will be recovered to an idle state.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The value found in the logical state field (rISPB.lc_state).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 11 Error Code: RPQ_COUNTS

Error Code:	RPQ_COUNTS
Error Description:	The number of receiving counter rIPKB tuples linked to this rISPB owning tuple is inconsistent with the value marked in the rISPB receiving counter (rISPB.rcvpktsqd) field. The receiving counter should contain a value equal to the number of receiving rIPKB tuples linked to the rISPB tuple under audit.
Possible Error Effect:	Interruption of Level 3 message traffic, and incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The value in the <i>rcvpktsqd</i> field is corrected to the number of receiving counter rIPKB tuples the audit found linked to this owning rISPB tuple, and a reset of a B X.25 level 3 logical channel is performed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The rISPB.rcvpktsqd field (receiving counter value found incorrect).
<i>Good Data</i> =	The number of receiving rIPKB tuples found linked to the owning rISPB tuple.
<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 12 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	A rISPB tuple is marked as being on a timer list (rISPB.busy_idle == OATIMER OR rISPB.busy_idle == OAITIMER) but the tuple is not actually linked to a RLDB_HEADC timer list head cell tuple (rISPB.dasphcm.st_ownk - DBSFDB_HEA != DBSPBITMR AND rISPB.dasphcm.st_ownk - DBSFDB_HEA != DBSPBTIMER).
Possible Error Effect:	Timers cannot be decremented, and level three messages may not be transmitted correctly causing associated calls to be lost to the system.
Corrective Action Taken:	The rISPB tuple will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The rISPB tuple's member linkage contents (rISPB.dasphcm.st_ownk - DBSFDB_HEA).
<i>Good Data</i> =	The contents of the <i>busy_idle</i> files and the timer head cell key values (rISPB.busy_idle << 30 DBSPBITMR << 24 DBSPBTIMER << 8).
<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 13 Error Code: UNEQUAL

Error Code:	UNEQUAL
Error Description:	The window size (rISPB.wind_sz) or the number of PVCs (rISPB.numpvcs) marked in the rISPB tuple is out of range. The window size must not be zero, and the number of PVCs should not be 1.
Possible Error Effect:	Interruption of Level 3 message traffic, and incorrect OSPS operations, measurements and billing. Possible exhaustion of OSPS data resources.
Corrective Action Taken:	The rISPB tuple will be recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The window size and the number of PVCs found marked in the rISPB tuple (rISPB.wind_sz << 16 rISPB.numpvcs).
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	The key of the rISPB tuple (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

Table 14 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rISPB tuple was found marked idle, and found linked to the RLDB_HEADC idle list head cell, but one or more of the other linkages in the tuple were found non-zero. For all rISPB tuples which are marked idle (rISPB.busy_idle == OAIDLE AND rISPB.dasphcm.st_ownk - DBSFDB_HEA == DBSPBIDLE), all other linkages in the tuple should be marked to zero.
Possible Error Effect:	Incorrect OSPA operations, measurements and billing. Possible exhaustion of OSPA data resources.
Corrective Action Taken:	The rISPB tuple is recovered to an idle condition.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISPB tuple.
<i>Bad Data</i> =	The contents of the first non-zero linkage found.
<i>Good Data</i> =	The contents of the <i>busy_idle</i> flag and the idle list head cell key for the rIDB_HEADC head cell tuple (rISPB.busy_idle << 16 DBSPBIDLE).
<i>Logical Index</i> =	The key of the rISPB tuple found corrupted (rISPB.spbkey).
<i>Dump</i> =	The contents of the rISPB tuple.

SPID

Software Release: 5E10 and later

1. SERVICE PROFILE IDENTIFIER (SPID) AUDIT

The SPID audit performs a static-to-dynamic existence check against the dynamic terminal equipment data block (TEDB) relation for multipoint DSL subscribers. This audit is part of ISDNBASE package and resides on the operating system for distributed switching-switching module (OSDS-SM). The SPID audit is running at the segmented mode as well as the initialization mode, during selective initialization.

2. DATA STRUCTURES

The RLSPID is a static relation with the spid as the key. Each SPID tuple represents the service profile subscribed by the user, and each SPID tuple may have circuit-switch, on-demand-packet, and/or permanent-packet services. Figure 1 shows the Data Structures of the SPID audit.

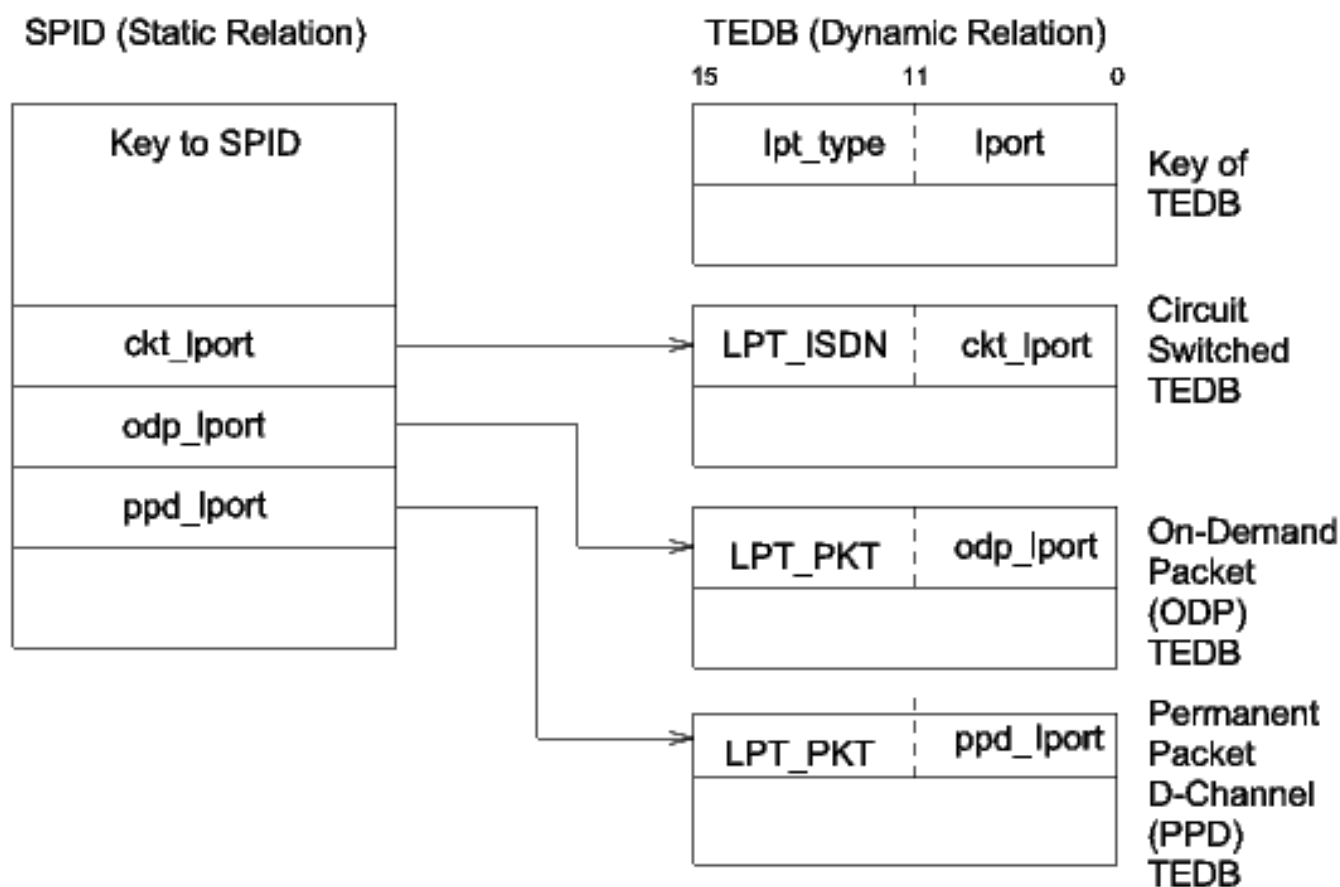


Figure 1 (SPID-1) Data Structures of the SPID Audit

3. AUDIT CHECKS

The SPID audit checks three attributes in rISPID (ckt_lport, odp_lport, and ppd_lport), to identify which service is subscribed by the user. For every service order profile (SOP), the TEDB tuple should be initialized and linked to the D-channel port block (DPB). If any TEDB is missing, the SPID audit invokes the recovery action to insert the TEDB and link it to the DPB.

4. ERROR HANDLING

Any missing TEDB tuple is initialized and linked to the DPB.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	There is a tuple missing in the dynamic RLTEDB relation; the corresponding static tuple is in RLSPID relation.
Possible Error Effect:	Application codes will ASSERT and calls cannot originate or terminate to this user.
Corrective Action Taken:	The tuple is initialized and linked to DPB (D-channel process block).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rISPID tuple.
<i>Bad Data =</i>	Counter of SOP in SPID. Valid values are: 1 = Circuit TEDB. 2 = On demand packet TEDB. 3 = Permanent packet TEDB.
<i>Good Data =</i>	Expected TEDB key.
<i>Logical Index =</i>	Expected TEDB key.
<i>Dump =</i>	Dump of the rISPID tuple.

SPRCRD

Software Release: 5E10 and later

1. SPARE CARD (SPRCRD) AUDIT

The integrated services line unit (ISLU) Z analog interface (ISLU-Z) provides analog call processing on the ISLU, including plain old telephone service (POTS) calls, multiparty calls, and coin phone operations on the ISLU. Several audits check the data structures used by the ringing generators (RGs), metallic access network (MAN), and sparing of faulty line cards. The audits include the peripheral control (PC) ringing occupancy table audit (RGOcup), the PC ringing task data block audit (RGTDDB), the ringing generator bus audit (RGBUS), the channel control block user audit (CCBUSR), the ISLU-Z access network bus audit (ISANBUS), the spare line card audit (SPRCRD), and the ISLU transaction audit (ISLUTRAN). The SPRCRD audit ensures that the sparing of ISLU-Z cards is handled correctly.

The SPRCRD audit is part of the peripheral control audits subgroup. It is part of the ISLU application loadable package (ISLUAPP). It runs in the operating system for distributed switching-switching module (OSDS-SM) environment in the initialization and segmented modes.

2. DATA STRUCTURES

The SPRCRD audit uses the following data structures. Sparing status is contained in the switch maintenance equipment status relation (RLSMEST). When sparing occurs, a recent change takes place to update the sparing information (faulty port name and spared port name) in the static active spare relation (RLACT_SPR). These data structures are shown in Figure 1.

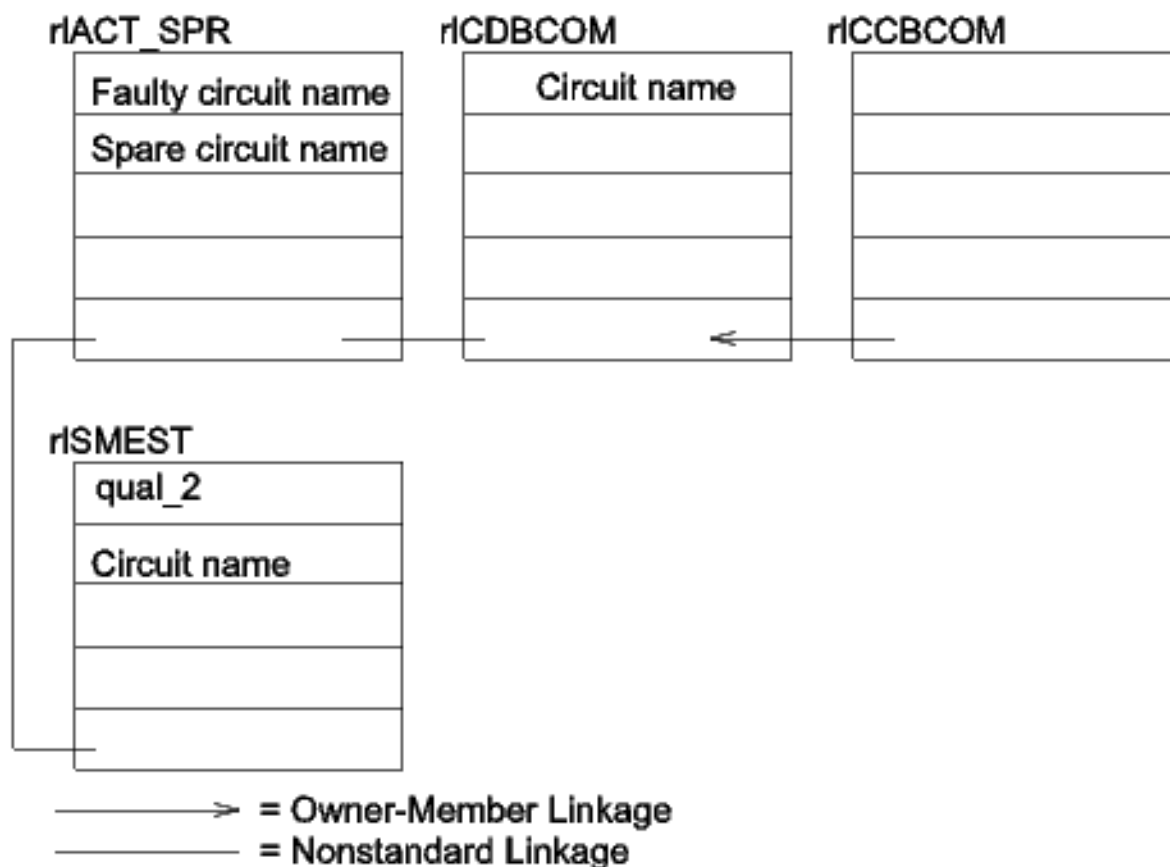


Figure 1 (SPRCRD-1) Data Structures of the SPRCRD Audit

3. AUDIT CHECKS

The SPRCRD audit verifies its input data and invokes its semantic check function. The semantic checks include the following.

The SPRCRD audit goes through each RISLU/ISLU in the switching module (SM) and reads the two static rIACT_SPR tuples associated with them. If the information in rIACT_SPR indicates that there is active sparing in that ISLU, the SPRCRD audit reads rISMEST with key the faulty port and spare port. Also, from the port name of the faulty port in rIACT_SPR, the audit derives the line group bus (LG) and the test bus (TST). From the spare port, it derives the spare bus (SPR) that is involved in this sparing connection.

Each bus (LG, TST, SPR) has an rICDBCOM tuple associated with it. If the bus is involved in a sparing connection, its rICDBCOM tuple should be owned by a head cell in rCCBCOM called DBSPARING. If not, the audit reports the error and sends a message to integrated services port administration (ISPA) to refresh that sparing connection and condition the ports.

The audit also checks the qual_2 in rISMEST. For the faulty port, it should be SMSPR. For the spare port, it should be SMDSPR. If not, the audit reports the error, sets the correct qual_2 in rISMEST, and sends a message to ISPA to refresh the sparing connection and condition the ports.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: INV

Error Code:	INV
Error Description:	The qual_2 state of the spare port is not SMDSPR.
Possible Error Effect:	Sparing SMEST data has been garbaged.
Corrective Action Taken:	The qual_2 state in rISMEST tuple has been corrected to SMAUTO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the sparing info.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Line unit number.
<i>Dump 1 =</i>	The rISMEST tuple of the spare port.
<i>Dump 2 =</i>	The rISMEST tuple of the faulty port.

Table 2 Error Code: QUAL

Error Code:	QUAL
Error Description:	The qual_2 state of the faulty port is not SMSPR.
Possible Error Effect:	Sparing SMEST data has been garbaged.
Corrective Action Taken:	The qual_2 state in rISMEST tuple has been corrected to SMAUTO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the sparing info.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Line unit number.
<i>Dump 1 =</i>	The rISMEST tuple of the spare port.
<i>Dump 2 =</i>	The rISMEST tuple of the faulty port.

Table 3 Error Code: SMEST

Error Code:	SMEST

Error Description:	DBfrdtup failed to read rISMEST for the faulty port.
Possible Error Effect:	Sparing data has been garbaged.
Corrective Action Taken:	Ignore this record and go to the next.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Return from DBfrdtup.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Faulty port circuit.
<i>Dump 1 =</i>	NA.

Table 4 Error Code: STATE

Error Code:	STATE
Error Description:	The base state of the faulty port and spare port indicate sparing, yet the cdb is not linked to the sparing head cell.
Possible Error Effect:	Sparing data has been garbaged.
Corrective Action Taken:	The CDBCOM tuple is quarantined to be cleaned up.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the sparing info.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Line unit number.
<i>Dump 1 =</i>	The rISMEST tuple of the spare port.
<i>Dump 2 =</i>	The rISMEST tuple of the faulty port.
<i>Dump 3 =</i>	The rCDBCOM tuple for the bus.

Table 5 Error Code: STATUS

Error Code:	STATUS
Error Description:	DBfrdtup failed to read rISMEST for the spare port.
Possible Error Effect:	Sparing data has been garbaged.
Corrective Action Taken:	Ignore this record and go the next.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Return from DBfrdtup.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Spare port circuit.
<i>Dump 1 =</i>	NA.

Table 6 Error Code: ZAP

Error Code:	ZAP
Error Description:	The SPRCRD has been entered in ZAP mode. The audit will be terminated and will start from the beginning the next time it is scheduled to run.
Possible Error Effect:	The SPRCRD audit is terminated before it completes all its checks.
Corrective Action Taken:	The audit will start from its beginning the next time it is scheduled to run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	0.
<i>Dump 1 =</i>	NA.

SRVOT

Software Release: 5E10 and later

1. LISTING SERVICES CALL DATA FOR OT & EIS INTERFACE (SRVOT) AUDIT

After originating treatment (OT) seizes a rISRVOT tuple for listing services (LS) originations from the RLDB_HEADC list to handle a call, it will link idle rISRVOT tuple one to one with rIPCBLA corresponding tuple to the originating treatment (OT) operating system for distributed switching (OSDS) process number.

2. DATA STRUCTURES

The data structure for SRVOT audit is the listing services call data for OT and external information system (EIS) interface relation (RLSRVOT). The rISRVOT tuples are linked to the RLDB_HEADC relation in a owner/member relationship. Two rIDB_HEADC tuples are defined for maintaining rISRVOT idle and busy list tuples.

This audit runs on switching modules (SM) which contain the OPSMOT package.

3. AUDIT CHECKS

The SRVOT audit invokes the DRINASEK driver to execute the syntactic checks and invokes the semantic function.

For each tuple, the SRVOT audit makes the following checks after the driver has verified the linkages. For details regarding each error code, see the comment blocks given.

ZERO_LNK = Check if the tuple should be on the idle list.

NOTBUSY = Check if the tuple should be on the busy list.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rISRVOT tuple is not busy when it should be.
Possible Error Effect:	The rISRVOT call tuple will be lost to the system.
Corrective Action Taken:	The corresponding rISRVOT tuple is initialized and returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISRVOT tuple.
<i>Bad Data</i> =	Actual owner's (RLSRVOT) invalid physical key (st link).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rISRVOT (srvot_num) tuple.
<i>Dump</i> =	rISRVOT tuple.

Table 2 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The rISRVOT tuple is in an indeterminate state. The tuple should be linked to an idle or busy list head cell (rISRVOT.dbheadc_lnk.st_ownk == DBSRVOTIDLE or DBSRVOTBUSY).
Possible Error Effect:	Lost OSPS calls, stuck operator positions, and invalid OSPS call measurement data.
Corrective Action Taken:	The corresponding rISRVOT tuple is initialized and returned to the idle list.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISRVOT tuple.
<i>Bad Data</i> =	Actual owner's (RLSRVOT) invalid physical key (st_ownk).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rISRVOT (srvot_num) tuple.
<i>Dump</i> =	The contents of the rISRVOT tuple.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	The rISRVOT tuple is not idle when it should be.
Possible Error Effect:	The rISRVOT tuple will be lost to the system.
Corrective Action Taken:	The corresponding rISRVOT tuple is initialized and returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rISRVOT tuple.
<i>Bad Data</i> =	Actual owner's (RLSRVOT) invalid physical key (st_ownk).
<i>Good Data</i> =	Expected owner's (RLSRVOT) physical key.
<i>Logical Index</i> =	Logical key of the rISRVOT (srvot_num) tuple.
<i>Dump</i> =	rISRVOT tuple.

SRVQDB

Software Release: 5E11 and later

1. SERVING TEAM QUEUE DATA BLOCKS (SRVQDB) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. Each serving team position is stored in the RLSRVQDB (serving team queue data blocks) relation. Data in this relation is audited by the SRVQDB audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLSRVQDB relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTESTPOS

Error Code:	BADTESTPOS
Error Description:	The audit found that the position number, posnum, value was inconsistent between a dynamic rISRVQDB tuple and the corresponding dynamic rIAGENTPOS (agent position) tuple.
Possible Error Effect:	This serving team position may not be used by this agent.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	This tuple's position number (rISRVQDB.posnum).
<i>Good Data =</i>	The rIAGENTPOS tuple's position number (rIAGENTPOS.posnum).
<i>Logical Index =</i>	The logical key of the rISRVQDB tuple.
<i>Dump 1 =</i>	The rISRVQDB tuple.
<i>Dump 2 =</i>	The rIAGENTPOS tuple.

Table 2 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	The audit found a rISRVQDB tuple corresponding to an available position to handle calls, but either this tuple's serving team number, serv_tm, represents the unoccupied list or the owning rIHC_SRVQ (head cell of serving team queues) tuple's serving team number, serv_tm, represents the unoccupied list.
Possible Error Effect:	This may prevent an agent using this serving team position from processing calls.

Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	The key of the unoccupied rIHC SRVQ tuple.
<i>Good Data =</i>	0. This distinguishes this IDLE LINKED error from others.
<i>Logical Index =</i>	The logical key of the rISRVQDB tuple.
<i>Dump 1 =</i>	The rISRVQDB tuple.
<i>Dump 2 =</i>	The rIHC SRVQ tuple.

Table 3 Error Code: IDLE_LINKED

Error Code:	IDLE_LINKED
Error Description:	The audit found a rISRVQDB tuple corresponding to a busy position, but either this tuple's serving team number, serv_tm, represents the unoccupied list or the owning rIHC SRVQ (head cell of serving team queues) tuple's serving team number, serv_tm, represents the unoccupied list.
Possible Error Effect:	This may prevent an agent using this serving team position from processing any more calls.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	The key of the unoccupied rIHC SRVQ tuple.
<i>Good Data =</i>	1. This distinguishes this IDLE LINKED error from others.
<i>Logical Index =</i>	The logical key of the rISRVQDB tuple.
<i>Dump 1 =</i>	The rISRVQDB tuple.
<i>Dump 2 =</i>	The rIHC SRVQ tuple.

Table 4 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a dynamic rISRVQDB tuple that does not have a corresponding dynamic rIHC SRVQ (head cell for serving team queues) tuple.
Possible Error Effect:	This position has no corresponding serving team, therefore the queued call may never get processed.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	Return value from DB (DBNO MATCH).
<i>Good Data =</i>	The tuple's serving team number (rISRVQDB.serv_tm).
<i>Logical Index =</i>	The logical key of the rIHC SRVQ tuple.
<i>Dump =</i>	The rISRVQDB tuple.

Table 5 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a rISRVQDB tuple representing an available or busy position that does not have a corresponding rAGENTPOS (agent position) tuple.
Possible Error Effect:	This may prevent an agent from using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	Return value from DB (DBNO MATCH).
<i>Good Data =</i>	The tuple's agent identification number (rISRVQDB.agent_id).
<i>Logical Index =</i>	The logical key of the rAGENTPOS tuple.
<i>Dump =</i>	The rISRVQDB tuple.

Table 6 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The audit found a rISRVQDB tuple corresponding to an unoccupied position, but either this tuple's serving team number, serv_tm, doesn't represent the unoccupied list or the owning rLHC_SRVQ (head cell of serving team queues) tuple's serving team number, serv_tm, doesn't represent the unoccupied list.
Possible Error Effect:	This may prevent an agent from using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	The tuple's serving team number (rISRVQDB.serv_tm).
<i>Good Data =</i>	The key of the unoccupied rLHC_SRVQ tuple.
<i>Logical Index =</i>	The logical key of the idle rISRVQDB tuple.
<i>Dump 1 =</i>	The rISRVQDB tuple.
<i>Dump 2 =</i>	The rLHC_SRVQ tuple.

Table 7 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The audit found a rISRVQDB tuple corresponding to an available position to handle calls, but this tuple's supporting queue number field, supq, was not zero. The supporting queue number field should be zero since the agent at this position is not currently handling any calls.
Possible Error Effect:	This position may not be able to process calls from a supporting queue.
Corrective Action Taken:	The audit will set the rISRVQDB tuple's supq field to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	The tuple's supporting queue number (rISRVQDB.supq).
<i>Good Data =</i>	The value the tuple's supporting queue number (rISRVQDB.supq) should be, 0.
<i>Logical Index =</i>	The logical key of the rISRVQDB tuple.
<i>Dump =</i>	The rISTPARM tuple.

Table 8 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The audit found an inconsistency between the dynamic rISRVQDB tuple and the static rLACD_POS tuple.
Possible Error Effect:	This may prevent an agent from using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	The value of the posdn_error flag.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The logical key of the rISRVQDB tuple.
<i>Dump 1 =</i>	The rISRVQDB tuple.
<i>Dump 2 =</i>	The rLACD_POS tuple.

Table 9 Error Code: STATE

Error Code:	STATE
Error Description:	The audit found a rISRVQDB tuple with an invalid position state pos_state value.
Possible Error Effect:	This may prevent an agent from using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rISRVQDB tuple.
<i>Bad Data =</i>	The tuple's position state value (rISRVQDB.pos_state).
<i>Good Data =</i>	NA.

<i>Logical Index</i> =	The logical key of the rISRVQDB tuple.
<i>Dump</i> =	The rISRVQDB tuple.

Table 10 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The audit found a rISRVQDB tuple corresponding to an unoccupied position that was not linked properly to the unoccupied rIHC_SRVQ (head cell of serving team queues) tuple.
Possible Error Effect:	This may prevent an agent from using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISRVQDB tuple.
<i>Bad Data</i> =	Return value from AU (AUFALSE).
<i>Good Data</i> =	DB linkage definition (DBHQSQ).
<i>Logical Index</i> =	The logical key of the rISRVQDB tuple.
<i>Dump 1</i> =	The rISRVQDB tuple.
<i>Dump 2</i> =	The rIHC_SRVQ tuple.

Table 11 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The audit found a rISRVQDB tuple corresponding to an available position that was not linked properly to a rIHC_SRVQ (head cell of serving team queues) tuple.
Possible Error Effect:	This may prevent an agent from using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISRVQDB tuple.
<i>Bad Data</i> =	Return value from AU (AUFALSE).
<i>Good Data</i> =	DB linkage definition (DBHQSQ).
<i>Logical Index</i> =	The logical key of the rISRVQDB tuple.
<i>Dump 1</i> =	The rISRVQDB tuple.
<i>Dump 2</i> =	The rIHC_SRVQ tuple.

Table 12 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The audit found a rISRVQDB tuple corresponding to a busy position that was not linked properly to a rIHC_SRVQ (head cell of serving team queues) tuple.
Possible Error Effect:	This may prevent an agent from further using this serving team position.
Corrective Action Taken:	The rISRVQDB tuple is recovered to the unoccupied list defined in the RLHC_SRVQ relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISRVQDB tuple.
<i>Bad Data</i> =	Return value from AU (AUFALSE).
<i>Good Data</i> =	DB linkage definition (DBHBSQ).
<i>Logical Index</i> =	The logical key of the rISRVQDB tuple.
<i>Dump 1</i> =	The rISRVQDB tuple.
<i>Dump 2</i> =	The rIHC_SRVQ tuple.

STINFO

Software Release: 5E10 and later

1. SERVING TEAM INFORMATION (STINFO) AUDIT

Features of the Operator Services Position System (OSPS) include basic automatic call distribution (ACD) and call handling. These features contain the call processing software needed for an operator to log onto a basic services terminal (BST) and handle requests for operator services. The ACD device routes calls to available positions. Audits are needed to maintain the integrity of the data structures needed to perform these OSPS functions. The STINFO audit verifies information used by the dynamic head cells serving team audit (HCSERVTM).

It is part of the OSPS audits subgroup. The audit runs in the OKP environment in the segmented modes.

2. DATA STRUCTURES

The STINFO audit uses two data structures, the static serving team information relation (RLSTINFO) and the dynamic serving team head cells relation (RLHCSERVTM).

3. AUDIT CHECKS

The STINFO audit calls the static driver (DRSTATIC) to perform static-to-dynamic existence checks on the tuples of RLSTINFO and RLHCSERVTM. It ensures that all tuples in RLSTINFO have a corresponding tuple in the dynamic relation. If there is no corresponding tuple, the audit creates one.

4. ERROR HANDLING

If the audit does not find a corresponding tuple in RLHCSERVTM, it creates one.

The STINFO audit schedules the HCSERVTM audit as a related audit when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	An rHCSERVTM tuple does not exist when it should.
Possible Error Effect:	This condition results in lost resources.
Corrective Action Taken:	The rHCSERVTM tuple will be inserted into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rHCSERVTM.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rHCSERVTM.
<i>Dump =</i>	NA.

STPARM

Software Release: 5E11 and later

1. SERVING TEAM PARAMETERS (STPARM) AUDIT

The automatic call distributor (ACD) management information system presents queued calls for processing to available agents who are logged into the switch through an integrated digital services (ISDN) position having full business and residential customer services (BRCS) features. The serving team data is stored in the RLSTPARM (call queue parameters) relation. Data in this relation is audited by the STPARM audit which runs in routine and elevated segmented modes.

2. DATA STRUCTURES

For details on the fields contained in the RLSTPARM relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found a static rIStPARM tuple that does not have a corresponding dynamic rIHC_SRVQ (head cell for serving team queues) tuple. As part of this error report a rIStPARM tuple is dumped. This is what distinguishes this LOSTTUP error from the others in this audit.
Possible Error Effect:	This may prevent queued calls from being presented to the corresponding serving team represented by this rIStPARM tuple.
Corrective Action Taken:	The audit will insert the missing dynamic rIHC_SRVQ tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rIStPARM tuple.
<i>Bad Data =</i>	The key of the rIHC_SRVQ tuple.
<i>Good Data =</i>	The key of the rIStPARM tuple.
<i>Logical Index =</i>	The logical key of the rIStPARM tuple.
<i>Dump =</i>	The rIStPARM tuple.

Table 2 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit found the dynamic idle rIHC_SRVQ (head cell for serving team queues) tuple to be missing. This LOSTTUP error does not dump any tuples. This is what distinguishes this LOSTTUP error from the others in this audit.
Possible Error Effect:	This may prevent queued calls from being processed by any serving team.
Corrective Action Taken:	The audit will insert the missing dynamic idle rIHC_SRVQ tuple.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	The address of the rIHC_SRVQ tuple.
<i>Bad Data</i> =	The key of the rIHC_SRVQ tuple.
<i>Good Data</i> =	The key of the idle rIHC_SRVQ tuple.
<i>Logical Index</i> =	The logical key of the idle rIHC_SRVQ tuple.
<i>Dump</i> =	NA.

Table 3 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit found the static principle serving team rICQMASK (call queue bit masks) tuple missing when trying to reconstruct a lost rIHC_SRVQ tuple.
Possible Error Effect:	The lost rIHC_SRVQ tuple will be inserted with an invalid count of call queues associated with the principle serving team cq_count.
Corrective Action Taken:	The audit will insert the missing dynamic rIHC_SRVQ tuple with a zero cq_count value.
Manual Action Required:	The missing static rICQMASK tuple must be inserted for the principle serving team represented by the rIStPARM tuple. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.
Dump Description:	
<i>Error Address</i> =	The role of serving team key of the rICQMASK tuple, 0.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rICQMASK tuple.
<i>Logical Index</i> =	The logical key of the rIStPARM tuple.
<i>Dump</i> =	The rIStPARM tuple.

Table 4 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit found the static supporting serving team rICQMASK (call queue bit masks) tuple missing when trying to reconstruct a lost rIHC_SRVQ tuple.
Possible Error Effect:	The lost rIHC_SRVQ tuple will be inserted with an invalid count of call queues associated with the supporting serving team scq_count.
Corrective Action Taken:	The audit will insert the missing dynamic rIHC_SRVQ tuple with a zero scq_count value.
Manual Action Required:	The missing static rICQMASK tuple must be inserted for the supporting serving team represented by the rIStPARM tuple. It should be inserted by recent change or ODBE with the key being the value which is dumped in good data.
Dump Description:	
<i>Error Address</i> =	The role of serving team key of the rICQMASK tuple, 1.
<i>Bad Data</i> =	Return value from DB (DBNO_MATCH).
<i>Good Data</i> =	The serving team key of the rICQMASK tuple.
<i>Logical Index</i> =	The logical key of the rIStPARM tuple.
<i>Dump</i> =	The rIStPARM tuple.

STQDB

Software Release: 5E10 and later

1. SERVING TEAM QUEUE DATA BLOCK (STQDB) AUDIT

The STQDB audit runs in both the automatic call distributor management information system (ACD/MIS) and the operator services position system (OSPS). Both the ACD/MIS and OSPS features are similar. Both features are used to present queued calls to agents that are logged into the 5ESS[®] switch. In the OSPS feature, the operator would log into the 5ESS[®] switch through a basic services terminal (BST) and handle requests for operator services. In the ACD/MIS feature, the operator would log into the 5ESS[®] switch through an integrated services digital network (ISDN) position having full business and residential customer services (BRCS) features. In both features, calls are routed to available agent positions for processing. In addition, supervisors are able to monitor all agent activities. Each agent is part of a serving team that is capable of handling certain types of calls.

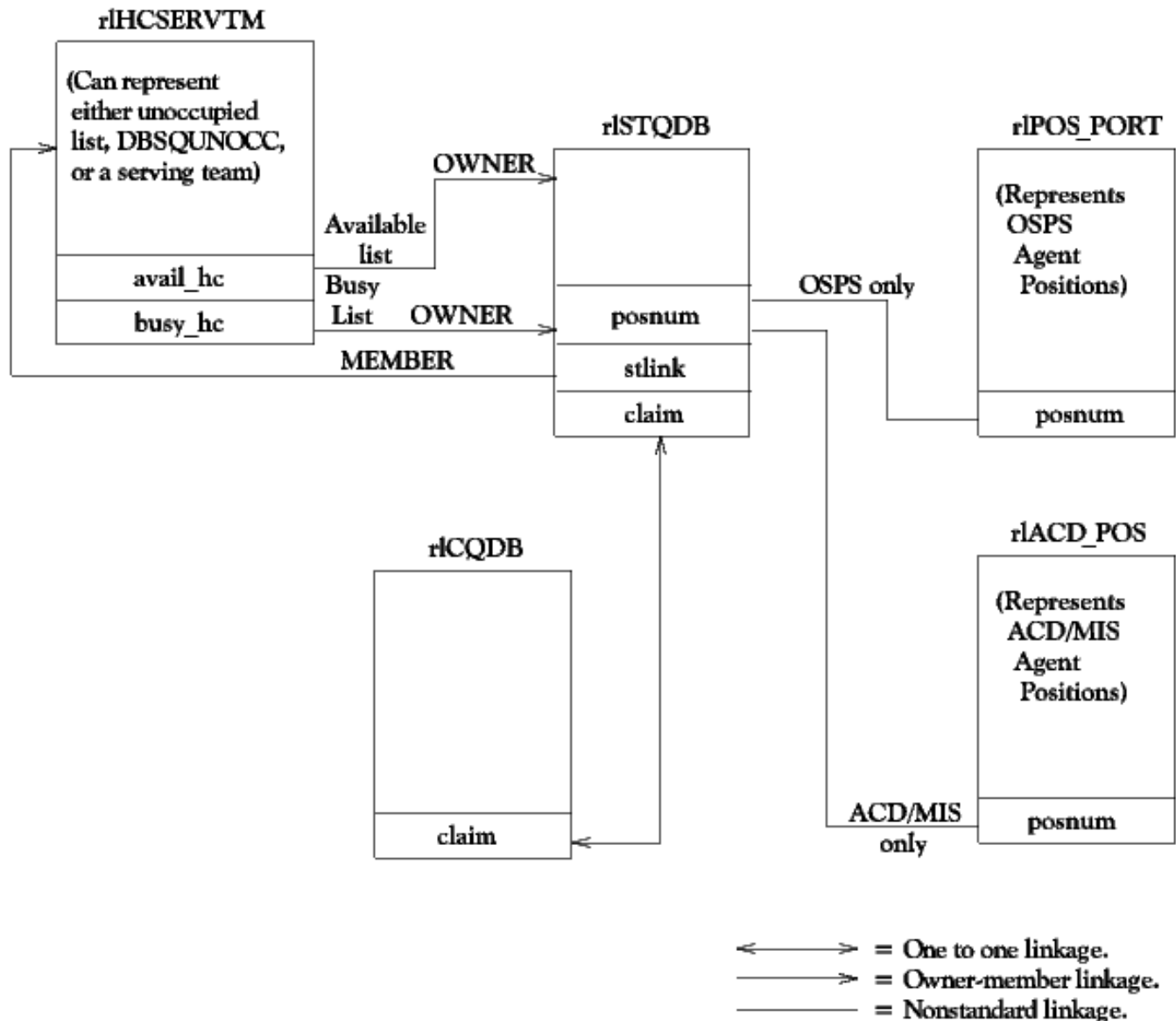
The purpose of the STQDB audit is to ensure the integrity of the dynamic rISTQDB tuples that describe agent positions. The STQDB audit runs in the operational kernel process (OKP) in the communications module processor (CMP) in segmented modes. It is part of the OPAMBACD software package.

2. DATA STRUCTURES

The RLSTQDB relation contains tuples that describe both OSPS and ACD/MIS operator positions in the 5ESS[®] office. Each tuple contains information about the agent's position number, the position's state, the position type, the agent's ID, the agent's serving team, and the agent's related port number for OSPS agents only. For ACD/MIS agents, the tuples contain the logical termination index and the position directory number. The *claim* field is used to contain the logical key of a call queued data block (CQDB) representing a call queued to this position. The *supq* field indicate the supporting call type queue that is being handled by this agent position. The rISTQDB tuples contain other fields as well.

Depending on the state of the rISTQDB tuples, they are linked to either a busy or idle list on their owning head cell serving team position queue (rHCSESVTM) tuple. Idle rISTQDB tuples that do not represent logged in agents are linked off a special rHCSESVTM tuple (key = DBSQUNOCC) that represents the unoccupied list. As an agent logs into the 5ESS[®] switch, an rISTQDB is removed from the unoccupied list defined in the RLHCSESVTM relation and linked to the rHCSESVTM tuple representing the serving team assigned to that agent. Once an agent logs out of the 5ESS[®] switch, its corresponding rISTQDB is moved back to the unoccupied list defined in the RLHCSESVTM relation.

Each rISTQDB tuple has a nonstandard linkage to either a corresponding position port (rIPOS_PORT) tuple for OSPS positions or an automatic call distributor position (rIACD_POS) tuple for ACD/MIS positions. The linkages to the rISTQDB tuples are shown in Figure 1.



Note: The rISTQDB tuple can only be linked to one of the fields, *avail_hc* or *busy_hc*, at a time.

Figure 1 (STQDB-1) RLSTQDB Linkages

3. AUDIT CHECKS

The STQDB audit uses the DRINDEX audit driver to traverse through each tuple in the RLSTQDB relation. For each tuple, the STQDB audit will make the following checks:

If the STQDB audit is entered:

TYPE = Check to make sure that the *pos_typ* type field in an rISTQDB tuple is not corrupt.

BADTYPE = Check to make sure that the *pos_typ* field in an rISTQDB tuple representing an OSPA agent position is a valid OSPA type.

NOTIDLE	= Check to make sure that an rISTQDB tuple marked as unoccupied is linked to the unoccupied list defined as the rIHCSERVMTM tuple with key DBSQUNOCC.
BDLINK	= Check to make sure that an rISTQDB tuple on the unoccupied list has a <i>claim</i> field of 0.
NOTZERO	= Check to make sure that an rISTQDB tuple marked as available has corresponding <i>supq</i> field set to 0.
SIMSTM	= Check to make sure that an rISTQDB tuple marked as available has a corresponding <i>serv_tm</i> field that points to the proper rIHCSERVMTM tuple representing that serving team.
UNAVAIL	= Check to make sure the member linkage off the RLHCSERVMTM relation is good for rISTQDB tuples.
NOTBUSY	= Check to make sure that an rISTQDB tuple marked as being busy or made busy is linked to the correct serving team head cell defined in the RLHCSERVMTM relation.
STATE	= Check that the rISTQDB tuple is in a valid position state as indicated in the <i>pos_state</i> field.
SEMANTIC	= Check to make sure the semantic fields defined in an rISTQDB match with their counterparts in the corresponding static rIACD_POS or rIPOS_PORT tuples.
CDTYPE	= Check to make sure the <i>pos_typ</i> field in the rISTQDB represents a valid OSPS or ACD/MIS agent type.
LOGBID	= Check to make sure that an rISTQDB tuple representing an agent not on the unoccupied list has a valid corresponding rLOGINID tuple.
BADTESTPOS	= Check to make sure that the position numbers match between an rISTQDB tuple representing a logged in agent and its corresponding rLOGINID tuple.
TSNO_MATCH	= Check to make sure that the serving team, <i>serv_tm</i> , in an rISTQDB representing a logged in agent matches the value found in the corresponding rIOLOGINID tuple.
LGLOSTTUP	= Check to make sure that an rISTQDB tuple representing a logged in agent has a corresponding static rIOLOGINID tuple.

4. ERROR HANDLING

Note that when an rISTQDB tuple that represents an agent position logged into the 5ESS® switch is recovered to the unoccupied list by the STQDB audit, call processing is not notified. Call processing will search the unoccupied list for the proper rISTQDB tuple if it is not currently linked to the correct serving team in the RLHCSERVMTM relation. Any communication from the switching module (SM) for this agent position will cause call processing to unlink the recovered rISTQDB tuple from the unoccupied list and link it to the correct rIHCSERVMTM tuple representing the agent's serving team. This eliminates the need for an interprocessor audit.

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADTESTPOS

Error Code:	BADTESTPOS
Error Description:	For an rISTQDB tuple that represents an agent position that is logged into the

	5ESS [®] switch, the audit has found that the position number, <i>posnum</i> field, in this rISTQDB tuple does not match the position number found in its corresponding rLOGINID tuple.
Possible Error Effect:	Possible mis-handling of calls presented to this agent position.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSEVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of <i>posnum</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The value of <i>posnum</i> in the rLOGINID tuple.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	The rLOGINID tuple.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	While reading the corresponding static rIPOS_PORT tuple for this rISTQDB tuple representing an OSPS agent, the audit found that the position type, <i>pos_typ</i> field, in the rISTQDB tuple does not match the corresponding value in its static rIPOS_PORT tuple.
Possible Error Effect:	Possible loss of the rISTQDB tuple resource.
Corrective Action Taken:	The position number in the rISTQDB tuple is updated to the value found in the corresponding rIPOS_PORT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of <i>pos_typ</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The value of <i>pos_typ</i> in the rIPOS_PORT tuple.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	The rIPOS_PORT tuple.

Table 3 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The audit found that the <i>claim</i> field in an rISTQDB tuple representing an unoccupied position was not zero. No rISTQDB tuple on the unoccupied list should have a call queued to it.
Possible Error Effect:	Possible loss of the rISTQDB tuple resource.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list and the <i>claim</i> field is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>claim</i> field in the rISTQDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 4 Error Code: CDTYPE

Error Code:	CDTYPE
Error Description:	The audit found an rISTQDB tuple that contained an unknown position type in the <i>pos_typ</i> field. Either the <i>pos_typ</i> field is corrupted or a new position type was added without modifying this audit.
Possible Error Effect:	Possible loss of the rISTQDB tuple resource.
Corrective Action Taken:	None. However, this error means that either the <i>pos_typ</i> field has been corrupted or a new position type was added without modifying this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>pos_typ</i> field in the rISTQDB tuple.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 5 Error Code: LGLOSTTUP

Error Code:	LGLOSTTUP
Error Description:	The audit could not read the corresponding rILOGINID tuple for this rISTQDB tuple representing an agent position logged into the 5ESS [®] switch. Either the static rILOGINID tuple is missing or the <i>opr_id</i> field in the corresponding dynamic rILOGINID tuple is corrupt.
Possible Error Effect:	Possible mis-handling of call processing for this agent position.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESRVTM relation.
Manual Action Required:	None. However, if the error persists, it may mean that the static rILOGINID tuple needs to be inserted for this agent by recent change or ODBE.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>opr_id</i> field in the rISTQDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	The rILOGINID tuple.

Table 6 Error Code: LOGBID

Error Code:	LOGBID
Error Description:	The audit was unable to obtain a pointer to the corresponding rILOGINID tuple for an rISTQDB tuple that represents an agent position that is logged into the 5ESS [®] switch. This may be due to a corrupt operator ID field, <i>opr_id</i> , in the rISTQDB tuple.
Possible Error Effect:	Possible mis-handling of call processing for this agent position.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESRVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>opr_id</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The value of the <i>intopid</i> returned from the DA primitive.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 7 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The position state field, <i>pos_state</i> , in an rISTQDB tuple indicates a busy agent position. However the rISTQDB tuple is linked to the unoccupied list defined in the RLHCSESRVTM relation.
Possible Error Effect:	Possible mis-handling of call processing for this agent position.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESRVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of DBSQUNOCC representing the key of the unoccupied list.
<i>Good Data</i> =	The logical key of the owning rIHCSESRVTM tuple.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	The rIHCSESRVTM tuple.

Table 8 Error Code: NOTIDLE

Error Code:	NOTIDLE

Error Description:	The audit found an rISTQDB tuple whos position state field, <i>pos_state</i> , indicates an unoccupied agent position. However, the rISTQDB tuple is not linked to the unoccupied list defined in the RLHCSESVTM relation.
Possible Error Effect:	Possible lost of the rISTQDB resource.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of DBSQUNOCC representing the key of the unoccupied list.
<i>Good Data</i> =	The logical key of the rIHCSERVMT tuple representing the unoccupied list.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	The owning rIHCSERVMT tuple.

Table 9 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	The support queue field, <i>supq</i> , is not zero for an rISTQDB that represents an available agent position.
Possible Error Effect:	Possible mis-handling of this agent position by call processing.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>supq</i> field in the rISTQDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 10 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	Various semantic fields in the rISTQDB tuple were found to be inconsistent with the values found in the corresponding rIACD_POS static tuple (for ACD/MIS agents) or the rIPOS_PORT static tuple (for OSPs agents). The audit dump with dump out either the rIACD_POS tuple for ACD/MIS agent positions or the rIPOS_PORT for OSPs agent positions. The type of agent position is indicated in the Bad Data portion of the audit dump.
Possible Error Effect:	Possible mis-handling of this agent position by call processing.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>pos_typ</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The value of the <i>locdn_flg</i> local variable flag.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	Based on position type, either the rIACD_POS tuple or the rIPOS_PORT tuple.

Table 11 Error Code: SIMSTM

Error Code:	SIMSTM
Error Description:	The rISTQDB under audit indicates an available agent position in the <i>pos_state</i> field. However, this tuple is linked off the unoccupied list defined in the RLHCSESVTM relation.
Possible Error Effect:	Possible mis-handling of this agent position by call processing.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSESVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.

<i>Bad Data</i> =	The value of the <i>serv_tm</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The logical key of the owning rLHCSEVTM tuple.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 12 Error Code: STATE

Error Code:	STATE
Error Description:	The position state field, <i>pos_state</i> , in the rISTQDB is not a known state or has been corrupted to an unknown state.
Possible Error Effect:	Possible mis-handling of this agent position by call processing.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSEVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>pos_state</i> field in the rISTQDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 13 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The serving team field, <i>serv_tm</i> , does not match between the dynamic rISTQDB tuple and its corresponding static rIOLOGINID tuple.
Possible Error Effect:	Possible mis-handling of this agent position by call processing.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list defined in the RLHCSEVTM relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>serv_tm</i> field in the rISTQDB tuple.
<i>Good Data</i> =	The value of the <i>serv_tm</i> field in the rIOLOGINID tuple.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump 1</i> =	The rISTQDB tuple.
<i>Dump 2</i> =	The rIOLOGINID tuple.

Table 14 Error Code: TYPE

Error Code:	TYPE
Error Description:	While the audit was trying to read the corresponding static rIPOS_PORT tuple or static rIACD_POS tuple for an rISTQDB tuple, it discovered that the position type field, <i>pos_typ</i> , has been corrupted.
Possible Error Effect:	Possible loss of the rISTQDB tuple resource.
Corrective Action Taken:	The <i>pos_typ</i> field is corrected from either the rIACD_POS tuple or the rIPOS_PORT tuple depending on which static read was successful using the <i>posnum</i> field as key. If the value of the Good Data field in the dump has a value of OA_ACDBRCS, then the <i>pos_typ</i> field is corrected from the rIACD_POS tuple. If it is any other value, then we have an OSPS position, and the <i>pos_typ</i> field is corrected from the rIPOS_PORT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of the <i>pos_typ</i> field in the rISTQDB tuple before correction.
<i>Good Data</i> =	The value of the <i>pos_typ</i> field in the corresponding static rIACD_POS or rIPOS_PORT tuples.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

Table 15 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The member linkage between the rISTQDB tuple and its owning rLHCSEVTM

	tuple has been corrupted.
Possible Error Effect:	Possible loss of the rISTQDB tuple resource.
Corrective Action Taken:	The rISTQDB tuple is recovered to the unoccupied list off the RLHCSEVTM relation. The BAD DATA field in the dump indicates which member linkage has been corrupted. If the unoccupied or available linkage has been corrupted, the BAD DATA field will have a value of DBASRVTMST. If the busy member linkage is corrupt, this will have a value of DBBSRVTMST.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rISTQDB tuple.
<i>Bad Data</i> =	The value of DBASRVTMST or DBBSRVTMST.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The logical key of the rISTQDB tuple.
<i>Dump</i> =	The rISTQDB tuple.

31. T1 - TE

T1USG

Software Release: 5E10 and later

1. T1 USAGE (T1USG) AUDIT

The *SLC*® 96 remote terminals connect to the digital carrier line units (DCLUs) in the interface module over T1 trunks. The availability of the time slots in T1 trunks needs to be checked continually to ensure that they are used efficiently. The T1 usage audit (T1USG) performs these checks.

The audit is part of the peripheral control audits subgroup. It runs in the segmented, nonsegmented, and initialization modes.

2. DATA STRUCTURES

The T1USG audit refers to the static port type (RLFC_PORTTYPE) and *SLC*® 96 equipment (RLS96EQUIP) relations while checking the dynamic *SLC*® 96 T1 usage relation (RLS96T1USG). Each tuple of the RLS96T1USG relation has 24 time slots. The tuples differ depending on the type of line they represent. As Figure 1 shows, special mode II lines have a one in the highest bit field and the time assignment unit (TAU) line number in the lowest bit field. Special mode II lines are dedicated for use by coin, PBX, and WATS lines; they are not concentrated as are other mode II lines. The lines that are not special mode II lines have zeroes in the highest bit field, and the service group number, peripheral interface data bus (PIDB) number, and PIDB time slot number in the remaining fields.

The *SLC*® 96 equipment relation tells how each remote terminal is equipped. The tuples of the relation contain the remote terminal number, the shelves to which each *SLC*® 96 digital facility interface (S-DFI) connects, and the modes of the two banks per remote terminal.

The T1USG audit also checks the peripheral control data path array (refer to the PATHMAP audit description) and, for mode II *SLC*® 96 lines, the stuffing pattern stored in the circuit data relation (RLCKTDATA).

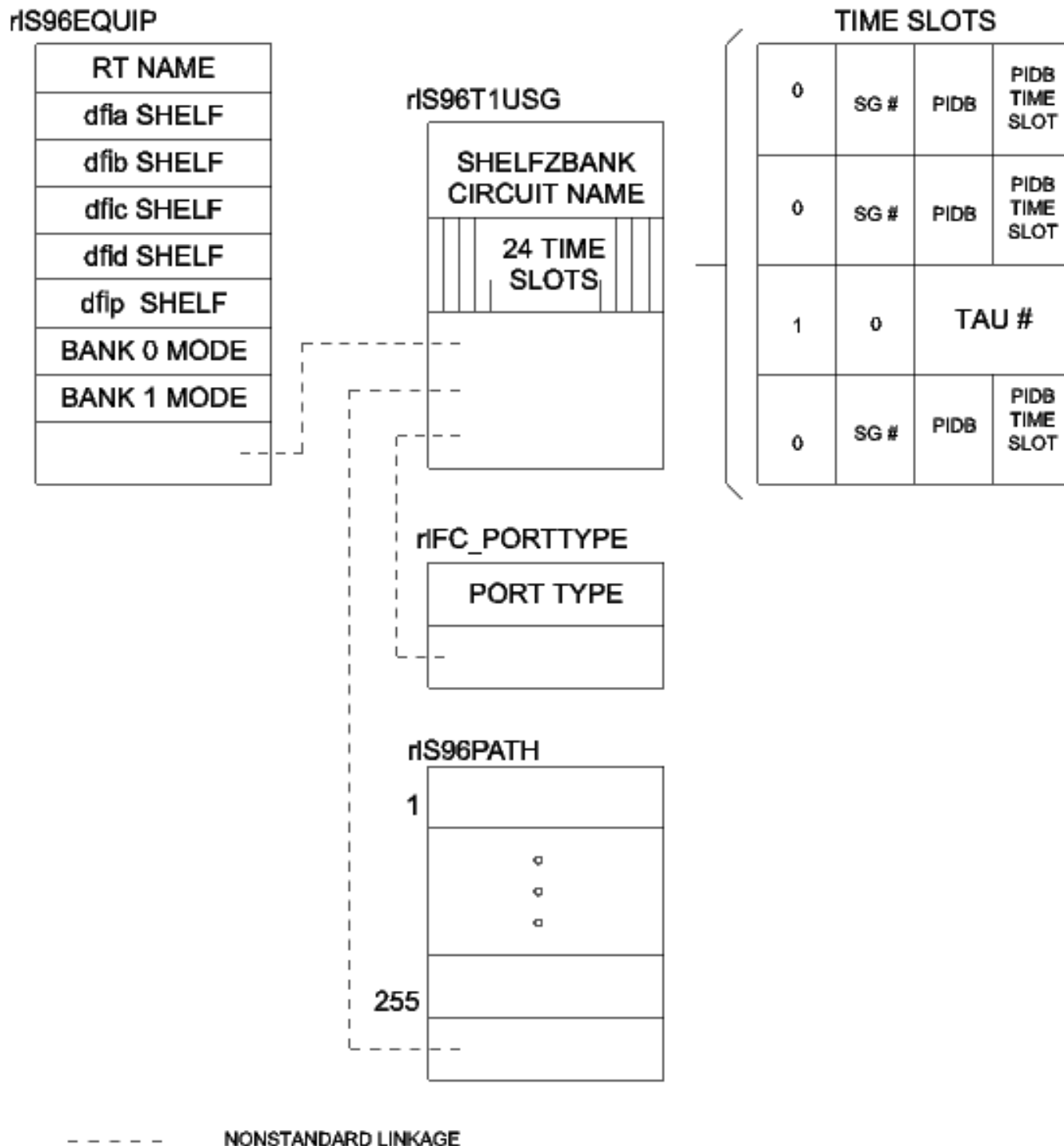


Figure 1 (T1USG-1) Data Structures of the T1USG Audit

3. AUDIT CHECKS

The audit invokes the static driver (DRSTATIC) and steps through RLS96T1USG. It checks with the RLS96EQUIP static relation and determines which tuples should exist in RLS96T1USG. For each tuple in the RLS96T1USG relation, the corresponding shelf should be listed in RLS96EQUIP. The EQUIP audit makes the static-to-dynamic check from RLS96EQUIP to RLS96T1USG.

The audit checks each time slot in rLS96T1USG.

- (a) If the time slot is idle, the audit proceeds to the next time slot.
- (b) If the time slot is not idle, the audit checks whether the tuple is a special mode II line. If the line is a special mode II line, the audit read RLFC_PORTTYPE and checks the port type listed there to ensure that the line should be special mode II. Valid entries include concentrated mode II coin, PBX, and WATS lines.
- (c) If the line is not special mode II, the audit determines the DCLU service group, PIDB time slot, and PIDB number. It accesses pcS96PATH to determine the S-DFI number and circuit name, and reads RLS96DFI to obtain the shelf name. The audit ensures that the shelf circuit name matches the circuit name in the key of RLS96T1USG.
- (d) Mode II SLC[®] 96 lines have a fixed correspondence between the PIDB time slots and the T1 time slot numbers. The relationship between the time slots is given by the stuffing pattern stored in RLCKTDATA. The T1USG audit verifies the relationship between the time slot data.

4. ERROR HANDLING

If the audit cannot find a tuple in RLS96T1USG corresponding to the information in RLS96EQUIP, it creates a tuple for the appropriate time slot in RLS96T1USG. The audit inhibits internal PIDB parity reporting, idles the corresponding hardware connections, and sets the erroneous entry in RLS96T1USG to idle. The T1USG audit then requests the SLC[®] 96 path map (PATHMAP), circuit data (CKTDATA), and port linkage area (PORTLA) audits.

The audit inhibits internal PIDB parity reporting because the sequence of events in taking down a SLC[®] 96 call processing path through the DCLU cannot be guaranteed. The DCLU data multiplexer should be taken down before the transmit and receive control RAMs. If it is not, parity errors occur. To nullify the consequences of the parity errors, the T1USG audit inhibits parity error reporting when it takes down the transmit and receive control RAMs. Parity reporting is reenabled when the CKTDATA audit runs.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: CONC_SLC

Error Code:	CONC_SLC
Error Description:	The T1 time slot passed into the audit and the derived T1 time slot are inconsistent.
Possible Error Effect:	Call processing may not be able to use this T1 time slot.
Corrective Action Taken:	This audit masks out the PIDB parity interrupt before it takes down the transmit control RAM in the S-DFI time slot interchanger and idle the corresponding software representation in the T1 time slot array in RLS96T1USG. For MODE II (CONCENTRATED) SLC [®] 96, it also removes the T1 time slot in the update assignment table so that the path through the time assignment unit (TAU) in the remote terminal is torn down. The BASE, DFISHELF, PATHMAP, CKTDATA and PORTLA audits are requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLS96T1USG tuple under audit.
<i>Bad Data =</i>	T1 time slot number currently being audited.
<i>Good Data =</i>	The derived T1 time slot number.
<i>Logical Index =</i>	Key of the rLS96T1USG tuple.
<i>Dump =</i>	The rLS96T1USG tuple under audit.

Table 2 Error Code: DBREAD

Error Code:	DBREAD
Error Description:	Cannot access the static relation rIFC_PORTTYP using the port name key constructed based on the time assignment unit number in the T1 time slot entry in the T1 time slot array and the shelf name of the tuple in the dynamic relation RLS96T1USG.
Possible Error Effect:	The T1 time slot may be lost. (Call processing cannot reuse this T1 time slot).
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple being audited.
<i>Bad Data =</i>	T1 time slot entry.
<i>Good Data =</i>	The relation ID of rIFC_PORTTYP.
<i>Logical Index =</i>	The T1 time slot number index into the T1 time slot array of the tuple in RLS96T1USG.
<i>Dump =</i>	Dump of the rIS96T1USG tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rIS96T1USG tuple exists but its static (rIS96EQUIP) tuple does not.
Possible Error Effect:	This error will result in call processing erroneously using rIS96T1USG dynamic data.
Corrective Action Taken:	The rIS96T1USG tuple will be deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple buffer.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of the rIS96T1USG tuple.
<i>Dump 1 =</i>	Dump of the rIS96T1USG tuple buffer.
<i>Dump 2 =</i>	Dump of the rIS96EQUIP tuple buffer.

Table 4 Error Code: GRBG

Error Code:	GRBG
Error Description:	The DFI number in PCS96PATH was found to be greater than or equal to the maximum number of DFI's possible or was found to be greater than or equal to the maximum possible service group or DFI's.
Possible Error Effect:	Call processing may not function properly.
Corrective Action Taken:	This audit masks out the PIDB parity interrupt before it takes down the transmit control RAM in the S-DFI time slot interchanger and idle the corresponding software representation in the T1 time slot array in RLS96T1USG. For MODE II (CONCENTRATED) SLC [®] 96, it also removes the T1 time slot in the update assignment table so that the path through the TAU in the Remote Terminal is torn down. The BASE, DFISHELF, PATHMAP, CKTDATA and PORTLA audits are requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple under audit.
<i>Bad Data =</i>	Value of the path data.
<i>Good Data =</i>	The value should be between 0 and 30 (except 15).
<i>Logical Index =</i>	The T1 time slot number being audited.
<i>Dump =</i>	The t1usg tuple under audit.

Table 5 Error Code: INV

Error Code:	INV
Error Description:	The S-DFI circuit name that is stored in AUs96inhp[] global array is not a valid S-DFI circuit name. The global array AUs96inhp[] is used to stored the S-DFI circuit name whose interrupt and the internal PIDB parity error mask of the

	corresponding DCLU is turned off while a SLC [®] 96 call which goes through that S-DFI is torn down.
Possible Error Effect:	Some program performed a wild write into AUs96inhp[] and it is not clear what effect it will have.
Corrective Action Taken:	The entry in AUs96inhp[] is zeroed.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the element of inhibit global array AUs96inhp[] that is wild written.
<i>Bad Data =</i>	Value found in that address.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The digital carrier line unit number which is used to index the array AUs96inhp[].
<i>Dump =</i>	The whole array AUs96inhp[].

Table 6 Error Code: LOST_CKTDATA

Error Code:	LOST_CKTDATA
Error Description:	The audit cannot read the static relation RLCKTDATA. The audit will repeatedly report until the error is fixed.
Possible Error Effect:	Call processing in CONCENTRATED SLC [®] -96 cannot work and asserts will fire.
Corrective Action Taken:	The PATHMAP, BASE, DFI_SHELF, CKTDATA and PORTLA audits are requested.
Manual Action Required:	The maintenance personnel need to recent change the RLCKTDATA relation to correct or prevent future errors.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple under audit.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Relation ID of relation trying to be read.
<i>Logical Index =</i>	Shelf name.
<i>Dump =</i>	None.

Table 7 Error Code: LOST_SHELF

Error Code:	LOST_SHELF
Error Description:	The audit is not successful in reading the dynamic relation RLS96SHELF.
Possible Error Effect:	Calls may not be able to terminate.
Corrective Action Taken:	The BASE, DFISHELF, PATHMAP, CKTDATA and PORTLA audits are requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of buffer used to store the rIS96SHELF tuple being read.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Relation ID of rIS96SHELF (this field should not be interpreted as good data but as an information field).
<i>Logical Index =</i>	S-DFI SHELF name that is used as the key to read the dynamic relation RLS96SHELF.
<i>Dump =</i>	NA.

Table 8 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The audit cannot read the RLS96T1USG relation after it comes back from a time break.
Possible Error Effect:	Some process performs a wild write to AUt1usg_buf while the audit is taking a time break. The effect of a wild write is hard to determine. In this case, it causes the audit to skip checking this tuple.
Corrective Action Taken:	The audit skip auditing this tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the RLS96T1USG tuple being audited.
<i>Bad Data =</i>	Not used.
<i>Good Data =</i>	Not used.
<i>Logical Index =</i>	The key of the SHELF name used to read the rIS96T1USG relation.
<i>Dump =</i>	Dump of the rIS96T1USG tuple whose read is not successful.

Table 9 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	Wrong subscriber digital facility interface (S-DFI) BANK mode. The audit will report the error until the correct mode is put in.
Possible Error Effect:	Cannot initialize the S-DFI.
Corrective Action Taken:	No automatic action can be taken.
Manual Action Required:	The maintenance personnel has to put in the proper mode in RLS96EQUIP.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple under audit.
<i>Bad Data =</i>	The value of the bank mode.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The t1tsno currently being audited.
<i>Dump 1 =</i>	The rIS96T1USG tuple under audit.
<i>Dump 2 =</i>	The rIS96EQUIP tuple associated with the rIS96T1USG tuple under audit.

Table 10 Error Code: OWNER

Error Code:	OWNER
Error Description:	The owner of a busy rICDBCOM is an rICBCOM that belongs to a head cell.
Possible Error Effect:	When the call is released, a link problem is found and ASSERTS will be fired.
Corrective Action Taken:	The entry that is found in error is idled in PCs96path[] and PSc96map[], the rICDBCOM associated with the PIDB time slot is quarantined; any valid terminal process is purged, and the BASE, DFISHELF, T1USG and CDBCOM audits are requested. That is, all the data structures and hardware associated with this call will be idled by this audit and other audits requested by this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM associated with the PIDB time slot.
<i>Bad Data =</i>	Owner of this rICDBCOM (a physical key of CCBCOM).
<i>Good Data =</i>	Any logical key in the rICBCOM that is greater than 0 is considered not a head cell.
<i>Logical Index =</i>	Peripheral interface data bus (PIDB) circuit name.
<i>Dump =</i>	Dump of rICDBCOM associated with the PIDB time slot.

Table 11 Error Code: SHELF_SLC

Error Code:	SHELF_SLC
Error Description:	The SHELF/SHELF GROUP name derived by the corresponding entry in the PCs96path[] global array is not the same as the SHELF/SHELF GROUP name of the tuple in RLS96T1USG being audited.
Possible Error Effect:	Call processing might not be able to reuse the T1 time slot.
Corrective Action Taken:	This audit masks out the PIDB parity interrupt before it takes down the transmit control RAM in the S-DFI time slot interchanger and idle the corresponding software representation in the T1 time slot array in RLS96T1USG. For MODE II (CONCENTRATED) SLC [®] 96, it also removes the T1 time slot in the update assignment table so that the path through the TAU in the remote terminal is torn down. The BASE, DFISHELF, PATHMAP, CKTDATA and PORTLA audits are requested.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple under audit.
<i>Bad Data =</i>	The shelf name extracted from the rIS96T1USG tuple.
<i>Good Data =</i>	The derived shelf name.
<i>Logical Index =</i>	T1 time slot number being audited.
<i>Dump 1 =</i>	The rIS96T1USG tuple under audit.
<i>Dump 2 =</i>	The rIS96DFI tuple associated with this rIS96T1USG tuple.

Table 12 Error Code: SLC_PORT

Error Code:	SLC_PORT
Error Description:	The logical port type of this SLC [®] 96 special line is not a special line type in the static relation RLFC PORTTYP.

Possible Error Effect:	The T1 time slot may be lost.
Corrective Action Taken:	The bank mode of this T1 time slot will be set to (PCMODE1) non-special line.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple being audited.
<i>Bad Data =</i>	Bank mode of the T1 time slot under audit.
<i>Good Data =</i>	Logical port type of port found in rIFC_PORTTYP.
<i>Logical Index =</i>	The T1 time slot number index into the T1 time slot array of the tuple in rIS96T1USG.
<i>Dump =</i>	Dump of the rIS96T1USG tuple.

Table 13 Error Code: SPCL_LINE

Error Code:	SPCL_LINE
Error Description:	A SLC [®] 96 special line bit is set on the entry of the T1 time slot array corresponding to a SHELF/SHELF GROUP that is not CONCENTRATED.
Possible Error Effect:	The T1 time slot may be lost.
Corrective Action Taken:	The T1 time slot entry for this T1 time slot will be set to idle.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIS96T1USG tuple being audited.
<i>Bad Data =</i>	The bank mode of the T1 time slot under audit.
<i>Good Data =</i>	PCMODE2.
<i>Logical Index =</i>	The T1 time slot number.
<i>Dump 1 =</i>	Dump of the rIS96T1USG tuple.
<i>Dump 2 =</i>	Dump of the rIS96EQUIP tuple.

TABGSM

Software Release: 5E11 and later

1. TABLE TO GLOBAL SM (TABGSM) AUDIT

TABGSM audit steps through the static relation RL GSM OPC (global SM to OPC and network). For each tuple in the rL GSM OPC, the audit checks the corresponding entry in the CCopc_gsm[]. The error will be corrected per the static RL GSM OPC relation.

2. DATA STRUCTURES

For details on the fields contained in the RL GSM OPC relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_CONG

Error Code:	BAD_CONG
Error Description:	The congestion type in the dynamic array which maps originating point codes to global switching modules does not match the corresponding field in the static relation RL GSM OPC for this global switching module (GSM).
Possible Error Effect:	Congestion control is not handled as the customer expects.
Corrective Action Taken:	The audit calls a common channel signaling 7 (CCS7) function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rL GSM OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rL GSM OPC.gsmi].congtype.
<i>Good Data =</i>	rL GSM OPC.congtype.
<i>Logical Index =</i>	rL GSM OPC.gsm.
<i>Dump =</i>	NA.

Table 2 Error Code: CCINDEX

Error Code:	CCINDEX
Error Description:	The index which CCS7 uses to index into CCopc_gsm[] for this GSM is not correct for this GSM.
Possible Error Effect:	Incorrect data for this GSM is accessed and CCS7 calls are not completed.
Corrective Action Taken:	The audit sets the CCS7 index to the correct index for this GSM.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rL GSM OPC.gsmi].
<i>Bad Data =</i>	CCS7 index.
<i>Good Data =</i>	rL GSM OPC.gsmi.
<i>Logical Index =</i>	rL GSM OPC.gsm.

<i>Dump</i> =	NA.

Table 3 Error Code: CIC_LENGTH

Error Code:	CIC_LENGTH
Error Description:	The circuit identification length in bits in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this global switching module.
Possible Error Effect:	CCS7 calls are set up for the incorrect circuit. In some cases, call cannot be completed.
Corrective Action Taken:	The audit calls a CCCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCopc_gsm[rlGSM_OPC.gsmi].
<i>Bad Data</i> =	CCopc_gsm[rlGSM_OPC.gsmi].cic_length.
<i>Good Data</i> =	rlGSM_OPC.cic_length.
<i>Logical Index</i> =	rlGSM_OPC.gsm.
<i>Dump</i> =	NA.

Table 4 Error Code: GSMI_BAD

Error Code:	GSMI_BAD
Error Description:	The abbreviated specification for this GSM in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this GSM.
Possible Error Effect:	CCS7 trunk maintenance performs incorrect operations. CCS7 calls cannot be made.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCopc_gsm[rlGSM_OPC.gsmi].
<i>Bad Data</i> =	CCopc_gsm[rlGSM_OPC.gsmi].gsmi.
<i>Good Data</i> =	rlGSM_OPC.gsmi.
<i>Logical Index</i> =	rlGSM_OPC.gsm.
<i>Dump</i> =	NA.

Table 5 Error Code: INT_DATA_BAD

Error Code:	INT_DATA_BAD
Error Description:	The network indicator in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this GSM.
Possible Error Effect:	CCS7 messages may be dropped and CCS7 calls cannot be made due to the dropped messages.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CCopc_gsm[rlGSM_OPC.gsmi].
<i>Bad Data</i> =	CCopc_gsm[rlGSM_OPC.gsmi].ni.
<i>Good Data</i> =	rlGSM_OPC.ni.
<i>Logical Index</i> =	rlGSM_OPC.gsm.
<i>Dump</i> =	NA.

Table 6 Error Code: INV_LENGTH

Error Code:	INV_LENGTH
Error Description:	The CCopc_gsm[].pc_length does not equal rlGSM_OPC.pc_length for the current GSM. The pc_length is the length of the point code in bits and maps originating point codes to GSMs.
Possible Error Effect:	CCS7 messages may be dropped, and CCS7 calls cannot be made due to the dropped messages.
Corrective Action Taken:	Since CCopc_gsm[] is built from relation RL GSM OPC, the audit requests CCS7 to rebuild CCopc_gsm[].
Manual Action Required:	None.

Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rIGSM_OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rIGSM_OPC.gsmi].pc_length.
<i>Good Data =</i>	rIGSM_OPC.pc_length.
<i>Logical Index =</i>	rIGSM_OPC.gsm.
<i>Dump =</i>	NA.

Table 7 Error Code: MTP_ACC

Error Code:	MTP_ACC
Error Description:	The CCopc_gsm[] maps global switching information to point codes and networks. It also has a field (CCopc_gsm[].mtp_acc) which indicates if the current global switching module is being used for message transfer part (MTP) accounting. The audit finds that CCopc_gsm[rIGSM_OPC.gsmi].mtp_acc does not equal rIGSM_OPC.mtp_acc in an rIGSM_OPC tuple whose key is the global switching module number (rIGSM_OPC.gsm).
Possible Error Effect:	Incorrect billing may result.
Corrective Action Taken:	The audit sets CCopc_gsm[rIGSM_OPC.gsmi].mtp_acc equal to rIGSM_OPC.mtp_acc.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIGSM_OPC.gsm.
<i>Bad Data =</i>	CCopc_gsm[rIGSM_OPC.gsmi].mtp_acc.
<i>Good Data =</i>	rIGSM_OPC.mtp_acc.
<i>Logical Index =</i>	rIGSM_OPC.gsmi.
<i>Dump =</i>	NA.

Table 8 Error Code: N7VAR_BAD

Error Code:	N7VAR_BAD
Error Description:	The National 7 variance type in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM_OPC for this GSM.
Possible Error Effect:	Message transfer part features may be activated/deactivated incorrectly.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rIGSM_OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rIGSM_OPC.gsmi].n7var.
<i>Good Data =</i>	rIGSM_OPC.n7var.
<i>Logical Index =</i>	rIGSM_OPC.gsm.
<i>Dump =</i>	NA.

Table 9 Error Code: NETWORK

Error Code:	NETWORK
Error Description:	The network to which this signaling point code belongs in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM_OPC for that GSM.
Possible Error Effect:	CCS7 calls cannot be completed. CCS7 trunk maintenance performs incorrect operations.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rIGSM_OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rIGSM_OPC.gsmi].network.
<i>Good Data =</i>	rIGSM_OPC.network.
<i>Logical Index =</i>	rIGSM_OPC.gsmi.
<i>Dump =</i>	NA.

Table 10 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	An entry in the CCopc_gsm array is not being used, but is not set to null values.

Possible Error Effect:	Invalid data may be accidentally accessed and CCS7 calls do not complete.
Corrective Action Taken:	The audit sets the array entry to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[i].
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into the array.
<i>Dump =</i>	NA.

Table 11 Error Code: OPC_BAD

Error Code:	OPC_BAD
Error Description:	The originating point code in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this GSM.
Possible Error Effect:	CCS7 calls cannot be completed. CCS7 trunk maintenance performs incorrect operations.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rl GSM OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rl GSM OPC.gsmi].opc.
<i>Good Data =</i>	rl GSM OPC.opc.
<i>Logical Index =</i>	rl GSM OPC.gsm.
<i>Dump =</i>	NA.

Table 12 Error Code: PC_FLD1

Error Code:	PC_FLD1
Error Description:	The number of bits in the highest ordered bits in the point code in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this GSM.
Possible Error Effect:	Input/output reports related to point codes may be inoperable.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rl GSM OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rl GSM OPC.gsmi].pc fld1.
<i>Good Data =</i>	rl GSM OPC.pc fld1.
<i>Logical Index =</i>	rl GSM OPC.gsm.
<i>Dump =</i>	NA.

Table 13 Error Code: PC_FLD2

Error Code:	PC_FLD2
Error Description:	The number of bits in the second highest ordered bits in the point code in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this GSM.
Possible Error Effect:	Input/output reports related to point codes may be inoperable.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rl GSM OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rl GSM OPC.gsmi].pc fld2.
<i>Good Data =</i>	rl GSM OPC.pc fld2.
<i>Logical Index =</i>	rl GSM OPC.gsm.
<i>Dump =</i>	NA.

Table 14 Error Code: PC_FLD3

Error Code:	PC_FLD3
Error Description:	The number of bits in the lowest ordered bits in the point code in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RL GSM OPC for this GSM.

Possible Error Effect:	Input/output reports related to point codes may be inoperable.
Corrective Action Taken:	The audit calls a CCS7 function to rebuild the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CCopc_gsm[rIGSM_OPC.gsmi].
<i>Bad Data =</i>	CCopc_gsm[rIGSM_OPC.gsmi].pc fld3.
<i>Good Data =</i>	rIGSM_OPC.pc fld3.
<i>Logical Index =</i>	rIGSM_OPC.gsm.
<i>Dump =</i>	NA.

Table 15 Error Code: PTCODE_BAD

Error Code:	PTCODE_BAD
Error Description:	The alias point code in the dynamic array which maps originating point codes to GSMs does not match the corresponding field in the static relation RLGSM_OPC for this global switching module.
Possible Error Effect:	CCS7 messages may be dropped, and CCS7 calls cannot be made due to the dropped messages.
Corrective Action Taken:	The audit rebuilds the array.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rIGSM_OPC.gsm.
<i>Bad Data =</i>	CCopc_gsm[rIGSM_OPC.gsmi].alias pc.
<i>Good Data =</i>	rIGSM_OPC.alias pc.
<i>Logical Index =</i>	rIGSM_OPC.gsmi.
<i>Dump =</i>	NA.

TALLOOPDB

Software Release: 5E10 and later

1. T&A LOOP DATA BLOCK (TALLOOPDB) AUDIT

The toll and assistance loop data block audit (TALLOOPDB) verifies the consistency of the toll and assistance loop data block relation (RLTALLOOPDB). It performs validity checks on the key, linkages, and quarantine field. It also verifies that each rTALLOOPDB tuple is linked to the correct list.

The TALLOOPDB audit must perform syntactic and semantic checks similar to the LOOPDB audit. In addition, it examines the member linkages from the rTALLOOPDB tuple to the rTAPOSDB tuple. It will also recover any lost rTALLOOPDB tuples by initializing them and linking them on the idle list. Any errors found with the rTALLOOPDB tuples are reported and the tuple is recovered. This audit runs in the routine mode in the normal audit cycle, and in elevated mode when scheduled by a related audit or by a user request.

2. DATA STRUCTURES

The TALLOOPDB audit structure is shown in Figure 1 .

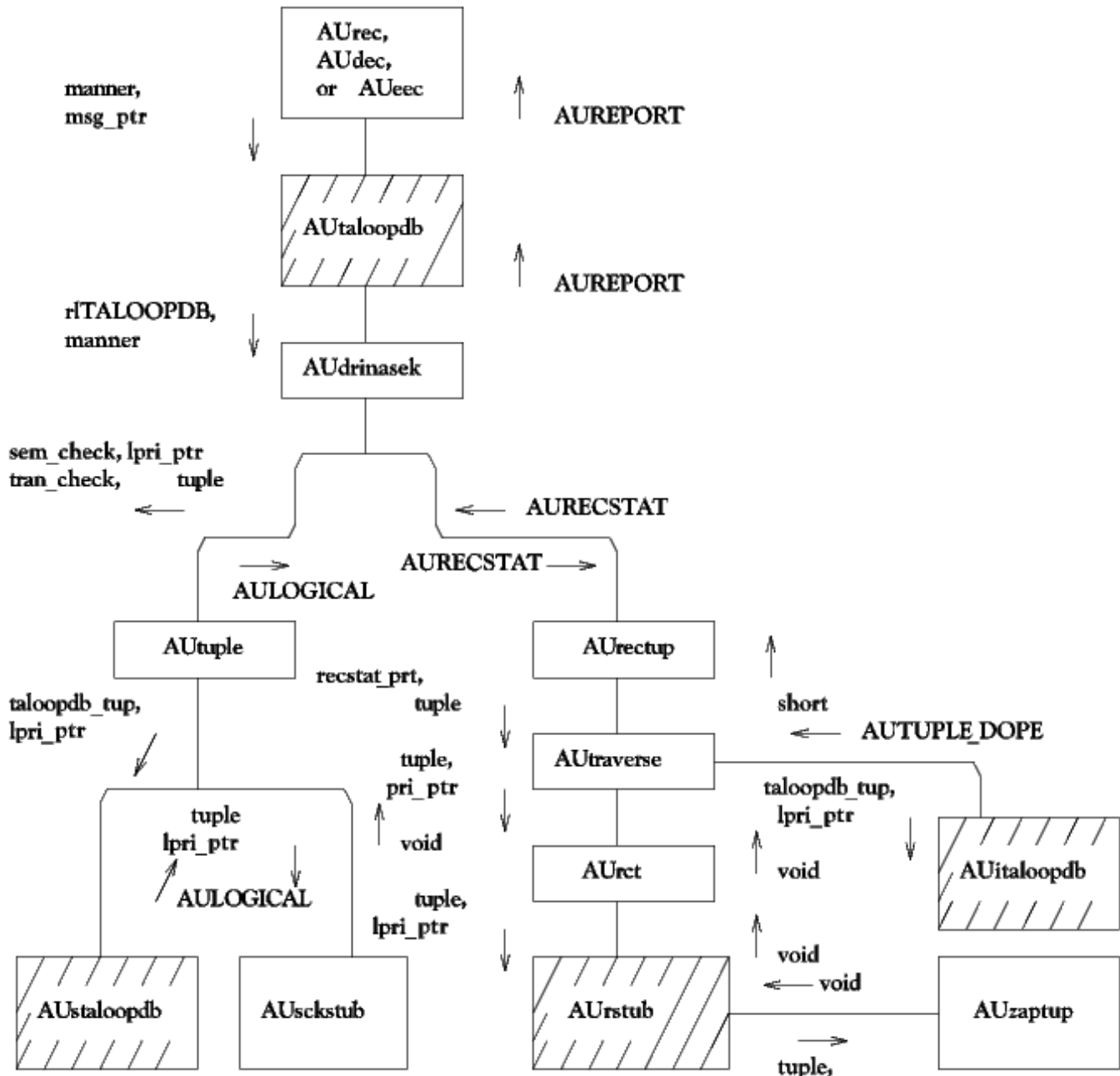


Figure 1 (TALoopDB-1) TALoopDB Audit Structure Chart

3. AUDIT CHECKS

This audit verifies the linkages to the TAPOSDB data block checking for lost data blocks and quarantined data blocks which are being used. The loop number is checked. If the loop number is 0 the audit will verify that the data block is on the idle list. If the loop number is 1-3, it will verify that the data block is on the busy list and linked to a TAPOSDB data block. If the loop number is not 0-3 it will report an error and tear down the call.

4. ERROR HANDLING

The TALoopDB audit calls the AUdrinasek driver to step through the rTALoopDB tuples. The driver performs syntactic checks on the key field, the quarantine field and any linkage fields. If the syntactic checks pass, the

semantic checks are performed.

Upon detection of any error by either the driver or the semantic routine, the contents of rTALLOOPDB are printed and the bad tuple is zeroed out and linked to the idle list.

It also schedules the TAPOSDB audit to be run in the elevated mode. The TALLOOPDB audit uses these error codes and also those used by the driver AUdrinasek.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The rTALLOOPDB is not busy when it should be.
Possible Error Effect:	The rTALLOOPDB will be lost to the system.
Corrective Action Taken:	The corresponding rTALLOOPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTALLOOPDB.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	rTALLOOPDB number.
<i>Dump</i> =	rTALLOOPDB.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rTALLOOPDB is not idle when it should be.
Possible Error Effect:	The rTALLOOPDB will be lost to the system.
Corrective Action Taken:	The corresponding rTALLOOPDB is returned to the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTALLOOPDB.
<i>Bad Data</i> =	Actual owner's invalid physical key.
<i>Good Data</i> =	Expected owner's physical key.
<i>Logical Index</i> =	rTALLOOPDB number.
<i>Dump</i> =	rTALLOOPDB.

Table 3 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The LOOPNO range has a invalid status.
Possible Error Effect:	If LOOPNO is invalid, operators may get indications for calls that are not really there.
Corrective Action Taken:	The corresponding position terminal process will traverse the call and purge it from the system.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTALLOOPDB.
<i>Bad Data</i> =	LOOPNO value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	rTALLOOPDB.

TAPOSDB

Software Release: 5E10 and later

1. TOLL AND ASSISTANCE (T&A) POSITION DATA BLOCK (TAPOSDB) AUDIT

The toll and assistance (T&A) position data block audit (TAPOSDB) verifies the consistency of the T&A position data block relation. It performs validity checks on linkages, verifies the consistency between the dynamic and the static attributes, checks for extra tuples and that idle TAPOSDBs are owned by the correct head cell.

The TAPOSDB audit verifies the physical linkages to the TALOOPDB data blocks and that the current loop number is valid and that the loop number is equal to the number of TALOOPDBs.

2. DATA STRUCTURES

The TAPOSDB audit structure chart is shown in Figure 1 .

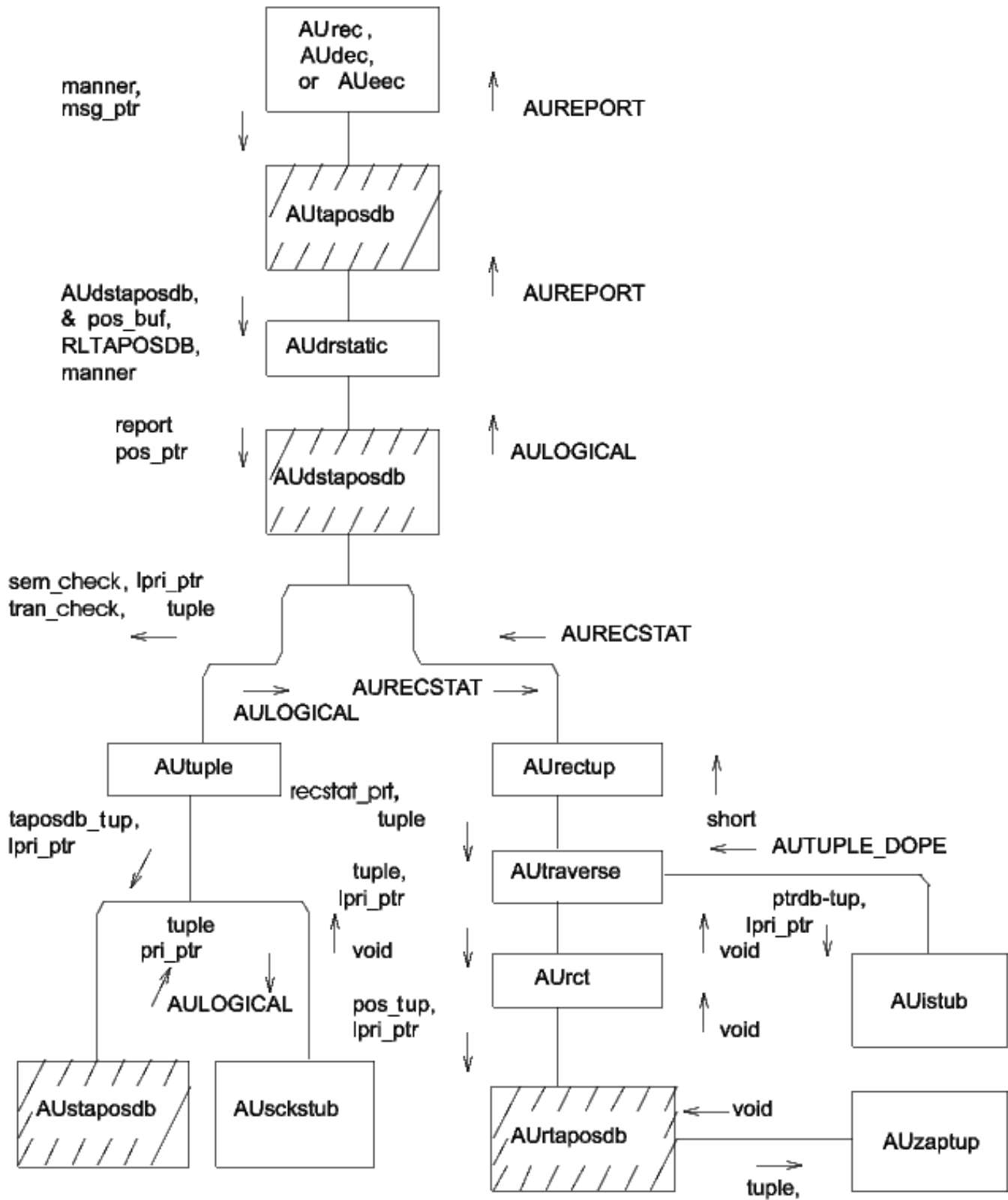


Figure 1 (TAPOSDB-1) TAPOSDB Audit Structure Chart

3. AUDIT CHECKS

The TAPOSDB audit examines each RLTAPOSDB tuple. It must perform all the syntactic and semantic checks done by the POSDB audit. There should be only one RLTAPOSDB per video display terminal (VDT) position in the SM. The LOOPDB_CT attribute is checked to insure that it corresponds to the number of valid rTALLOOPDB linkages. The audit will allow up to three linkages to rTALLOOPDBs. The intopid and serv_tm attributes are compared to the corresponding values stored in the static tuple rIOLOGINID. posnum is compared with the static tuple rILPRT_POS. The owner linkages to rTALLOOPDB and rIAPB, and the one-to-one linkage to rIPTRDB are validated. Any errors found are reported and the tuple is recovered. This audit will run in the routine mode in the normal audit cycle, and in elevated mode when scheduled by a related audit or by a user request.

4. ERROR HANDLING

The TAPOSDB audit calls the driver AUdrstatic to step through the rITAPOSDB tuples. The driver performs syntactic checks on the key field, the quarantine field and any linkage fields. If the syntactic checks pass, the semantic checks are performed.

Upon detection of errors in either the syntactic or semantic data, recovery is invoked. Whenever an error is found in a rITAPOSDB tuple, the rITAPOSDB tuple contents will be printed and the related audits PTRDB, TALLOOPDB, and APB will be scheduled to run in the elevated mode.

When an error is detected in a rITAPOSDB tuple, the audit will recover the bad tuple and send a message to the ACD/AM to notify it that the position has been taken out of service. The ACD primitive is called to take the appropriate action to log off the operator.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	An extra tuple was found.
Possible Error Effect:	Garbage left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The extra rITAPOSDB tuple that was found is deleted. A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rITAPOSDB.
<i>Dump =</i>	rItAPOSDB tuple.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	An extra tuple was found.
Possible Error Effect:	Garbage left in the relation, which may be invalidly accessed and taken to be valid data.
Corrective Action Taken:	The extra rITAPOSDB tuple that was found is deleted. A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rITAPOSDB.
<i>Dump =</i>	rItAPOSDB tuple.

Table 3 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The ODD for this port in rITAPOSDB are corrupted, it is compared against the static rIOLOGINID.
Possible Error Effect:	The operator-id, (op_id), the internal operator id, (intopid), and the serving team, (serv_team) calls the ACsferrct function which will call an assert. The function will abort causing a halt to handling the message.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB tuple.
<i>Bad Data =</i>	NA (see dump of rITAPOSDB).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rITAPOSDB.
<i>Dump 1 =</i>	rITAPOSDB.
<i>Dump 2 =</i>	rIOLOGINID.

Table 4 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	A static tuple exists for rILPRT_POS but the static tuple for rIOLOGINID doesn't exist.
Possible Error Effect:	rITAPOSDB will be updated from the rILPRT_POS, but the attributes related to rIOLOGINID won't be updated.
Corrective Action Taken:	The rIOLOGINID must be created and initialized. A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	Action must be taken to create and initialize the OLOGINID relation.
Dump Description:	
<i>Error Address =</i>	Address of rILPRT_POS tuple.
<i>Bad Data =</i>	NA (see dump of rILPRT_POS).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key of rILPRT_POS.
<i>Dump =</i>	rILPRT_POS.

Table 5 Error Code: PSTATUS

Error Code:	PSTATUS
Error Description:	The status of LOOP_CT linkages to the rITAPOSDB is invalid.
Possible Error Effect:	This may attempt to serve calls that don't exist. It may also cause calls not to be served that do exist.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB.
<i>Bad Data =</i>	LOOP_CT value.
<i>Good Data =</i>	Correct number of rTALOOPDB's linked to the rITAPOSDB.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	rITAPOSDB.

Table 6 Error Code: PTR1

Error Code:	PTR1
Error Description:	The pointer from the TAPOSDB to the first TALOOPDB is bad.
Possible Error Effect:	The direct link from the TALOOPDB to the TAPOSDB is missing, but the TALOOPDB is linked to the TAPOSDB indirectly. This could cause the PTP to use the wrong rTALOOPDB tuple.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITAPOSDB tuple.

<i>Bad Data</i> =	loopladdr.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITAPOSDB.
<i>Dump</i> =	rITAPOSDB.

Table 7 Error Code: PTR2

Error Code:	PTR2
Error Description:	The pointer from the TAPOSDB to the second TALOOPDB is bad.
Possible Error Effect:	The direct link from the TALOOPDB to the TAPOSDB is missing, but the TALOOPDB is linked to the TAPOSDB indirectly. This could cause the PTP to use the wrong rITALOOPDB tuple.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITAPOSDB tuple.
<i>Bad Data</i> =	loop2addr.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITAPOSDB.
<i>Dump</i> =	rITAPOSDB.

Table 8 Error Code: PTR3

Error Code:	PTR3
Error Description:	The pointer from the TAPOSDB to the third TALOOPDB is bad.
Possible Error Effect:	The direct link from the TALOOPDB to the TAPOSDB is missing, but the TALOOPDB is linked to the TAPOSDB indirectly. This could cause the PTP to use wrong rITALOOPDB tuple.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITAPOSDB tuple.
<i>Bad Data</i> =	loop3addr.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITAPOSDB.
<i>Dump</i> =	rITAPOSDB.

Table 9 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The ODD for this port kept in the rITAPOSDB are corrupted. The position number is compared with static rLPRTPOS.
Possible Error Effect:	If the position number is occupied and shouldn't be it will want to access a call. If the position number is unoccupied and should be it won't access the call.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITAPOSDB tuple.
<i>Bad Data</i> =	NA (see dump of rITAPOSDB).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITAPOSDB.
<i>Dump</i> =	rITAPOSDB.

Table 10 Error Code: TYPE

Error Code:	TYPE
Error Description:	The sanity between initial position terminal process and initial originating terminal process is lost. The communication between the interflow B-channel is lost.
Possible Error Effect:	The communication between the interflow B-channel is lost.
Corrective Action Taken:	The associated B-channel will be removed and restored. The initial position

	terminal process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITAPOSDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rITAPOSDB tuple logical key (posdb_key).
<i>Dump</i> =	rITAPOSDB tuple.

Table 11 Error Code: UNAVAIL

Error Code:	UNAVAIL
Error Description:	The loop number in the TALOOPDB is not within the valid range of 1 - 3 for active loops.
Possible Error Effect:	This could be a loop left over from a previous call or a valid loop with a bad loop number. This means that the TAPOSDB data connected to this PTP may be corrupt.
Corrective Action Taken:	A log off message is sent to the AM and the corresponding PTP process is PURGED.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITAPOSDB tuple.
<i>Bad Data</i> =	loopno field in rITALOOPDB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITAPOSDB.
<i>Dump</i> =	rITAPOSDB.

TATDB

Software Release: 5E10 and later

1. TERMINATING ATTEMPT TRIGGER DATA BLOCK (TATDB) AUDIT

The advanced system platform (ASP) family of services has an effect on the terminating side of a call. terminating attempt triggers (TATs) allow the service switching point (SSP) to launch queries, based on the terminating directory number (DN), prior to determining and seizing the physical port to which the incoming call would normally be terminated. As a result of the query, the originator may receive busy or reorder, the call may be forwarded, or an announcement sequence may be started to collect more information from the originator. TATs are assigned to analog lines only, and may be assignable to MLHGs as well as DNs. Trunks, ISDN attendants (ISATs) and other ISDN lines will not be allowed to have TATs assigned to them but may originate calls to TATs.

2. DATA STRUCTURES

The RLATATDB dynamic relation serves as a holding area for the MGLN_TERM message while the terminating side waits for the response to the TCAP query and sometimes as well for the play and collect announcements. It contains a tuple for each TAT terminating process and is a compacted indexed relation.

The TATDB audit references the following dynamic relations:

RLDB_HEADC = One tuple of this relation acts as the idle list headcell for relation RLATATDB, linked through *nsidx_mem*.

RLNS_IDX = A busy rLATATDB tuple is linked through *nsidx_mem.st_link* to an rINS_IDX tuple.

Figure 1 shows the relationship of these data structures.

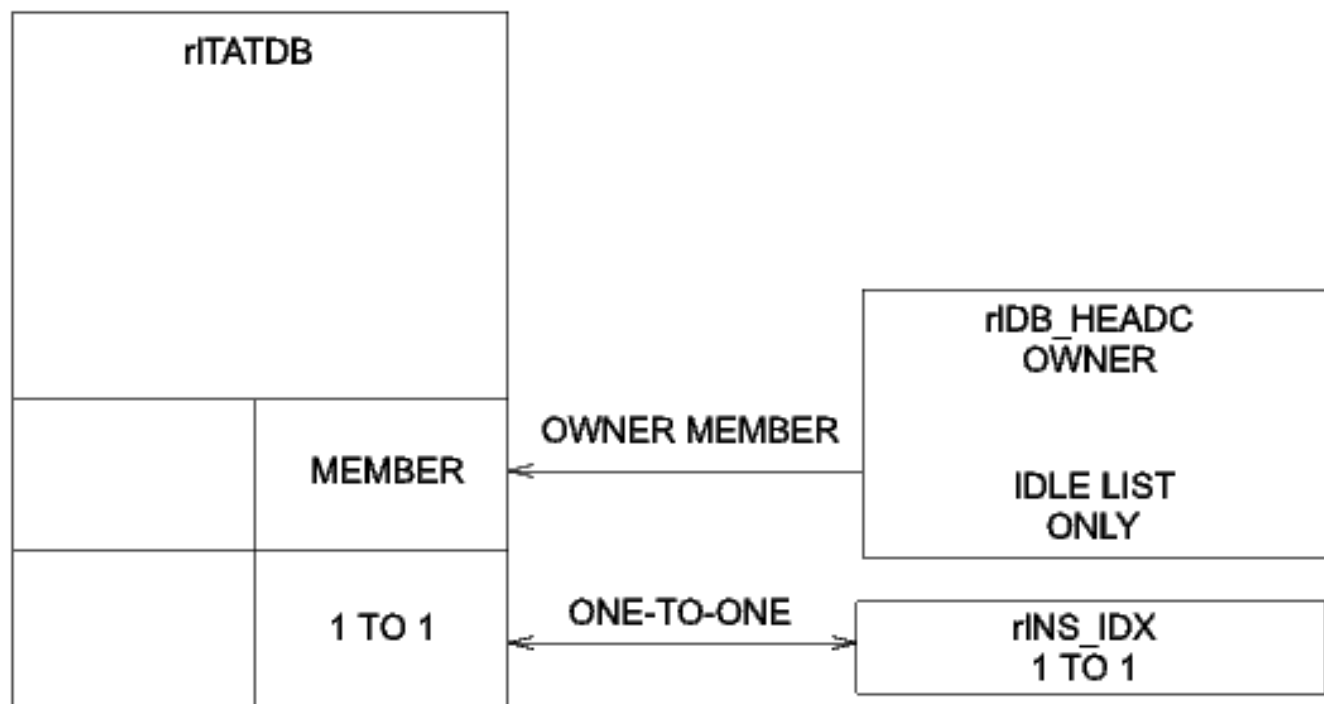


Figure 1 (TATDB-1) Data Structures of the TATDB Audit

3. AUDIT CHECKS

The TATDB audit uses the DRINASEK audit driver. For each rITATDB tuple found in the relation, the audit makes the following checks:

IDLE_LINK = The owning linkage of the rITATDB tuple must be zero or attached to the idle headcell in RLDB_HEADC.

ZERO_LINK = If the rITATDB tuple is idle, other linkages should be NULL.

If the rITATDB tuple is busy:

DEADPID = The tuple should be linked to an rINS_IDX tuple.

4. ERROR HANDLING

The audit produces the following error codes and those generated by the DRINASEK driver.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rITATDB tuple is not linked to its headcell nor to a rINS_IDX tuple. It must be linked to one or the other.
Possible Error Effect:	The rITATDB tuple is a lost resource, unusable by call processing.
Corrective Action Taken:	The rITATDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rITATDB tuple where the error was found.
<i>Bad Data</i> =	The link to an rINS_IDX tuple (rITATDB.nsidx_mem.st_link).
<i>Good Data</i> =	The zeroed link to rIDB_HEADC (rITATDB.nsidx_mem.st_ownk).
<i>Logical Index</i> =	The key to the rITATDB tuple (rITATDB.tatdb).
<i>Dump</i> =	The rITATDB tuple where the error was found.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rITATDB tuple is not linked correctly to either rIDB_HEADC or rINS_IDX.
Possible Error Effect:	The rITATDB tuple is a lost resource, unusable by call processing.
Corrective Action Taken:	The rITATDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the rITATDB tuple where the error was found.
<i>Bad Data</i> =	The link to rIDB_HEADC (rITATDB.nsidx_mem.st_ownk).
<i>Good Data</i> =	The value the link should be (DBTATIDLE).
<i>Logical Index</i> =	The key to the rITATDB tuple (rITATDB.tatdb).
<i>Dump</i> =	The rITATDB tuple where the error was found.

Table 3 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rITATDB tuple is still linked to an rINS_IDX tuple. The requirement is that this linkage is null when the rITATDB tuple is idle.
Possible Error Effect:	Resources are on the idle list. However, since they are still linked together, when

	an rINS_IDX tuples is obtained, the rITATDB tuple will already be associated incorrectly. This may cause call failures.
Corrective Action Taken:	The rITATDB tuple is recovered to an idle state, and related audits will be scheduled to recover any related data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the rITATDB tuple where the error was found.
<i>Bad Data =</i>	The bad linkage to an rINS_IDX tuple.
<i>Good Data =</i>	The link to rIDB_HEADC (rITATDB.nsidx mem.st_ownk - DBSFNS_IDX).
<i>Logical Index =</i>	The key to the rITATDB tuple (rITATDB.tatdb).
<i>Dump =</i>	The rITATDB tuple where the error was found.

TCB

Software Release: 5E10 and later

1. TIMER CONTROL BLOCK (TCB) AUDIT

The 5ESS[®] switching system requires various timing capabilities:

- Some processes in the administrative module (AM), switching module (SM) or communications module processor (CMP) must note or report exactly when certain events occur. Such processes must therefore be able to access a time-of-day clock and a calendar.
- Several subsystems require interval timing. For example, the peripheral control and the feature control subsystems require timing intervals of about 128 milliseconds.
- In many situations, processes wait for events from other processes only for a fixed duration. These processes require a time-out interval. If the process receives the awaited event before the time-out, the time-out request is canceled. If the awaited event is not received before the time-out, operating system for distributed switching (OSDS) sends a timer to the process, which then takes appropriate action.
- Certain activities must occur on a cyclic basis. Examples are automatic diagnostic testing of circuits and noncritical input/output jobs.

To implement the various timing requirements, OSDS uses special data structures called timer control blocks (TCBs), or timers. Each timer control block represents a timed event. A fixed number of timer control blocks are provided by OSDS per processor. The TCB audit runs in the OSDS kernel process (OKP) and switch maintenance kernel process (SMKP) in the AM, the SM and the CMP.

2. DATA STRUCTURES

Figure 1 illustrates the layout of a timer control block and Figure 2 shows the fields checked by this audit.

In addition to the TCBs themselves (OSbtcb[]), the audit verifies and uses the following global data:

- The array of process control blocks (OSbpcb[]) any of which may own a doubly linked list of TCBs. The ends of the list are indexed by the fields PCB.tcb0 and PCB.tcbn. Each process is also allocated one dedicated timer whose index is the same as the PCB's. If the dedicated timer is in use, the index is kept in PCB.timeout.
- The global indexes OSfftcb and OSlftcb which indicate the first and last members of the doubly linked TCB idle list.
- The global array (OStheap[]) which is used to implement a heap algorithm for ordering timers by the time they are supposed to fire (TCB.go_off). The TCB points to its position in the heap using TCB.thi.
- The global index OSfftheap which indicates the first available entry in the heap.

Figure 3 shows the configuration of the idle list. Figure 4 is a graphical representation of an ordered binary tree stack. Figure 5 is a graphical representation of a timing heap and its linkages.

OStcb

tstate
go_off
thi
llink
rlink
owner

Figure 1 (TCB-1) Timer Control Block

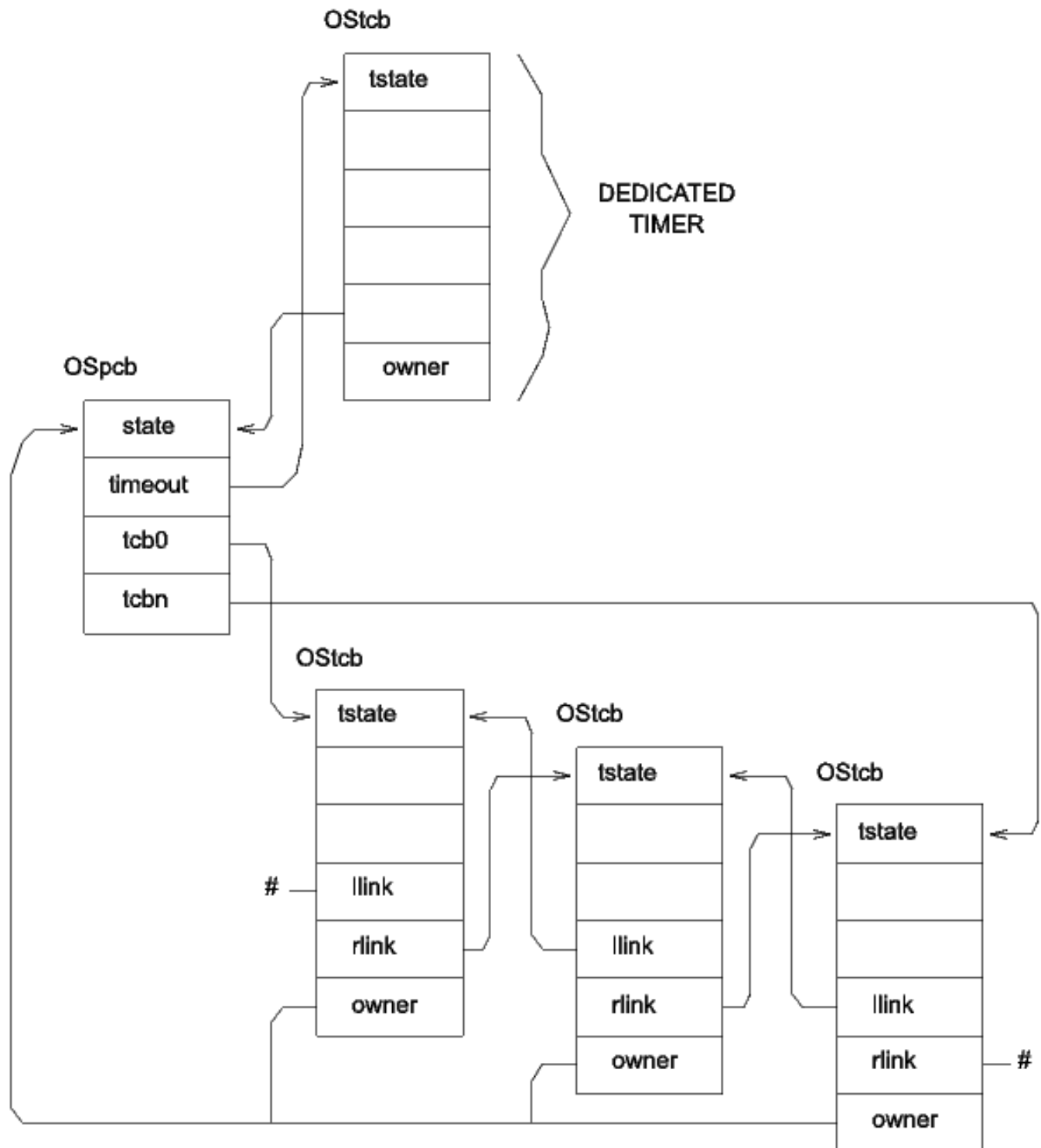


Figure 2 (TCB-2) Configuration of Timer and Process Control Blocks

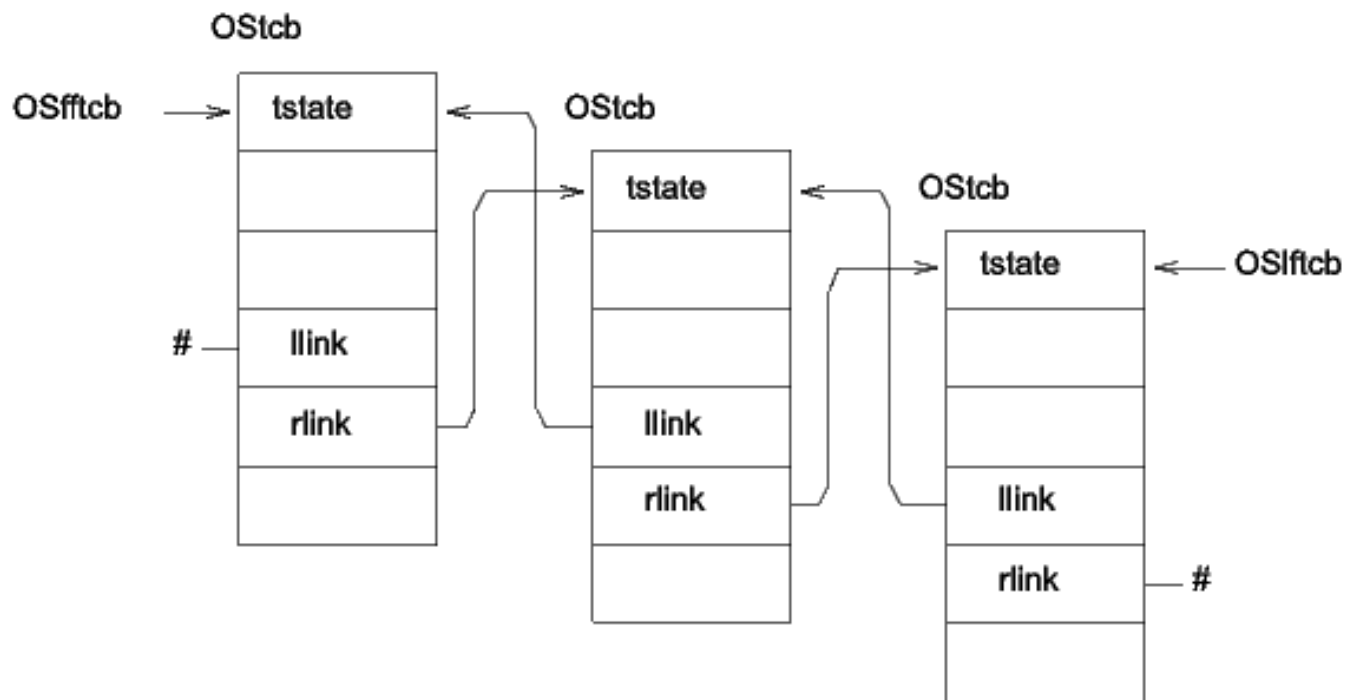


Figure 3 (TCB-3) Idle List Configuration

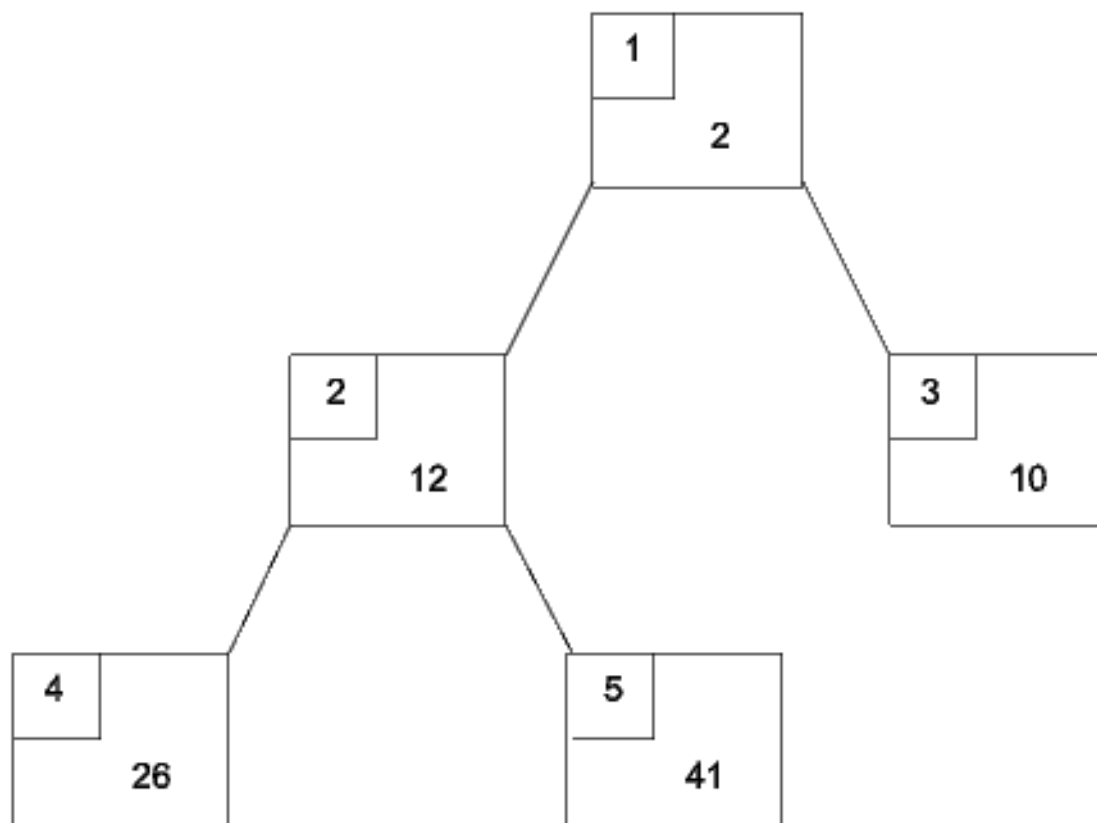


Figure 4 (TCB-4) Ordered Binary Tree Stack

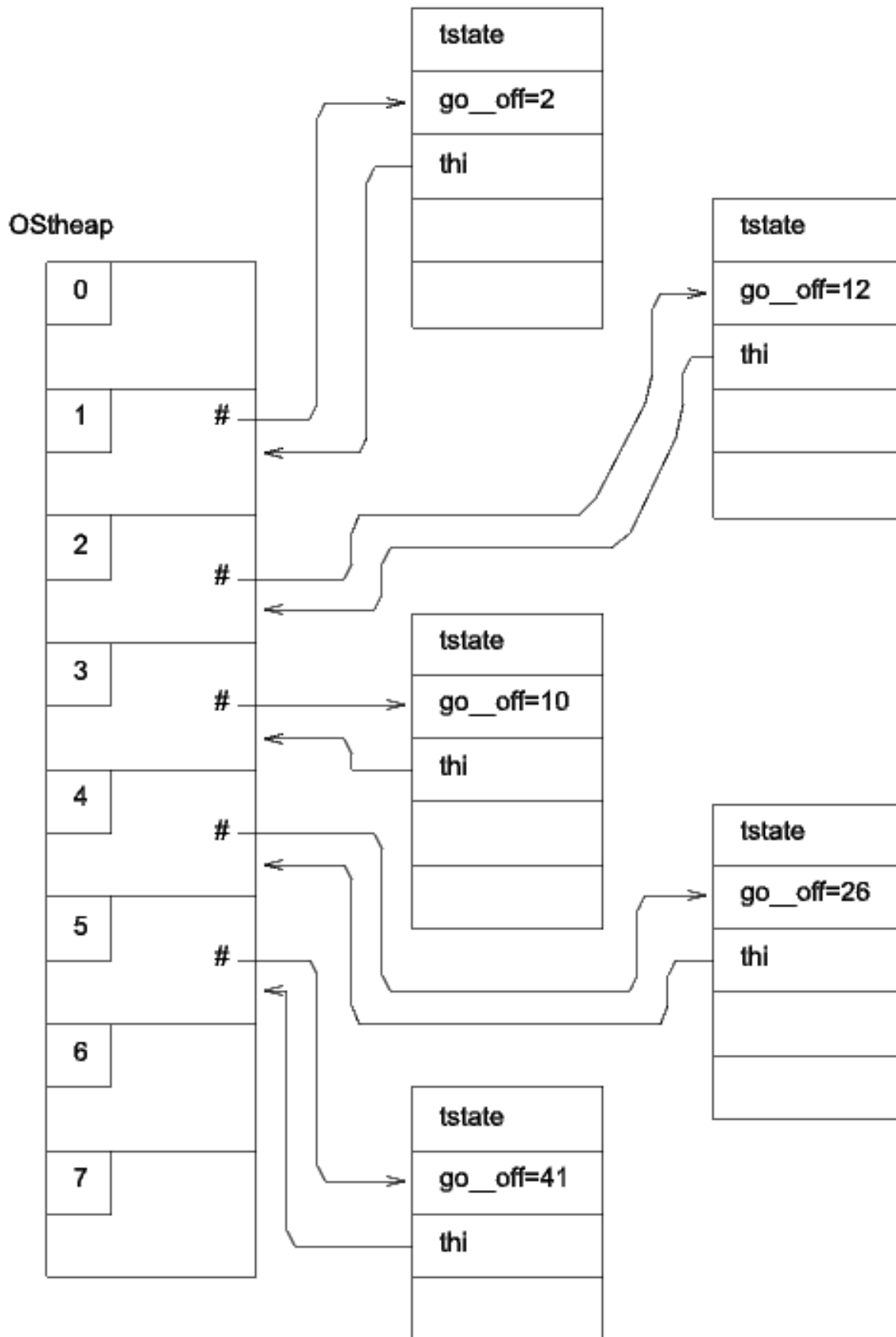


Figure 5 (TCB-5) Timing Heap and Linkages**3. AUDIT CHECKS**

The TCB audit does not use any audit driver. All error codes associated with this audit are described. The following list gives a basic indication of the order of the checks. For details regarding each error code see the comment blocks given.

If the audit is being (re)entered:

ZAP = Check for system interrupt handling.

If the audit is attempting to access the TCB:

MISSING = Check for inaccessible TCBs.

If the audit is verifying the idle list:

LNK2ALST = Range check OSIf tcb.

ALNK2QUE = Semantic check of OSIf tcb.

BLNKINQUE = Semantic check of OSff tcb.

LNKIALST = Range check idle TCB's llink field.

PT2PT = Point-to/point-back check between TCBs.

BADOWNER = Check idle TCB's owner field.

STATE = Check idle TCB's tstate field.

If the audit is verifying the use of dedicated timers:

PROC_DED = Check use of dedicated timer.

If the audit is verifying a TCB list attached to a PCB:

LNK2ALST = Range check PCB.tcbn.

ALNK2QUE = Semantic check PCB.tcbn.

BLNKINQUE = Semantic check PCB.tcb0.

LNKIALST = Range check TCB.llink.

PT2PT = Point-to/point-back check between TCBs.

BADOWNER = Check TCB's owner field.

TIMERLNK = Check TCB's thi ("to heap index").

If the audit is verifying the heap:

HEAP_FREE = Range check OSfttheap.
 HEAP_NULL = Semantic check OSfttheap.
 HEAP_TO = Point-to/point-back check from the heap to a timer.
 BAD_TYPE = Range check of TCB.tstate.
 HEAP_GOFF = Semantic check of TCB.go_off and TCB.tstate.
 PARENT = Range check heap element's parent.
 TIMEORDER = Check order of heap element and its parent.

If the audit is looking for lost TCBs:

KEY = Check TCB.tcbkey.
 INVOWNER = Range check potentially lost TCB's owner field.
 INV = Check for lost TCBs.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ALNK2QUE

Error Code:	ALNK2QUE
Error Description:	An index to the last element in a list of TCBs (OSiftcb if the free list is being checked, otherwise PCB.tcbn for some non-idle PCB) indicates a TCB which either has an incorrect owner (not OSNULL for the free list, different from the PCB number in all other cases) or has a non-null (not OSNULL) rlink field.
Possible Error Effect:	The list may not be accessed correctly. For the free list this could mean lost TCBs (leading to overload situations) or non-idle TCBs being taken for new processes (affecting the process to which they rightfully belong). For TCB lists attached to PCBs this problem could cause the owning process to lose timers or falsely believe it has timers which actually belong to the idle list or other processes. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the TCB audit in directed mode. It is also likely that TCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of TCBs and the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by first verifying OSfttcb (index to the first free TCB) and, if it is ok, verifying the list as far as possible from the top down. If the list is attached to a PCB the associated process will be purged.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	The owner of the list we're trying to access (either OSNULL or a PCB index).
<i>Good Data</i> =	Contents of the TCB's owner field.
<i>Logical Index</i> =	The TCB's index (= OSif tcb in the free list case, = PCB.tcbn otherwise).
<i>Dump 1</i> =	The TCB.
<i>Dump 2</i> =	The owning PCB or nothing if the free list was being checked.

Table 2 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	A TCB's owner field shows a different owner than the owner of the list it was found in. Note that the TCB's rlink field has already been verified (PT2PT) and its llink has been checked to be in range (BLINKINQUE/LNKIALST).
Possible Error Effect:	If the owning list belongs to a PCB then the associated process could erroneously be activated using a timer. Regardless of where the TCB was found, it is possible that it actually belongs on a list belonging to some process in which case that process will be missing a timer. If the TCB was found on the free list than INV errors are also likely to show up.
Corrective Action Taken:	If this is the free list of TCBs just relink the list around the bad timer and let an INV error fire to put it back on the idle list if that's where it really belongs. If the list is attached to a PCB the associated process will be purged. Although the offending TCB may actually belong somewhere else and the remaining TCBs on this list are ok, this list is still corrupted and cannot be trusted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	Contents of the TCB's owner field.
<i>Good Data</i> =	The owner of the list the TCB was found in (either OSNULL or a PCB index).
<i>Logical Index</i> =	The TCB's index (= OSif tcb in the free list case, = PCB.tcbn otherwise).
<i>Dump</i> =	The TCB.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	A TCB found in the heap has an invalid state. That is, TCB.tstate is not OSQTIMER, CTIMER, ATIMER, CATIMER or OSQTIMEOUT.
Possible Error Effect:	The TCB may not belong in the heap. The timer could fire erroneously causing incorrect actions by the process which owns it, asserts or a single process purge.
Corrective Action Taken:	If the audit is not already running in directed mode it will escalate. In any case, OStheap[] will have its bad entry cleared and then be reheaped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	The TCB's state (TCB.tstate).
<i>Good Data</i> =	Contents of the TCB's owner field.
<i>Logical Index</i> =	The TCB's index.
<i>Dump</i> =	1The TCB.

Table 4 Error Code: BLNKINQUE

Error Code:	BLNKINQUE
Error Description:	While traversing a list (bottom up) a TCB was found with its llink field equal to OSNULL but was not the first TCB in the list according to the list owner. Specifically, if the TCB was found on the free list, it was not indicated by

	OSfftcB (index to the first free TCB).
	If the TCB was found on a list belonging to a PCB, it was not indicated by the PCB's tcb0 field (index to the first TCB belonging to the process).
Possible Error Effect:	The list is either broken or the list owner's index to the first element is corrupt. This could have any of the effects of an ALNK2QUE, LNKIALST or PT2PT error.
Corrective Action Taken:	If this is the free list of TCBs and the audit is running in segmented mode, escalate to directed mode and rebuild the list. If the audit is already running in directed mode then rebuild the list right away. The list will be rebuilt by first verifying OSfftcB (index to the first free TCB) and, if it is ok, verifying the list as far as possible from the top down. If the list is attached to a PCB the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
	<i>Error Address</i> = Address of the TCB.
	<i>Bad Data</i> = Index of the TCB the list's owner says is first on the list.
	<i>Good Data</i> = The owner of the list the TCB was found in (either OSNULL or a PCB index).
	<i>Logical Index</i> = The TCB's index.
	<i>Dump 1</i> = The TCB.
	<i>Dump 2</i> = The owning PCB or nothing if the free list was being checked.

Table 5 Error Code: HEAP_FREE

Error Code:	HEAP_FREE
Error Description:	The index (OSfftheap) to the first available entry in the heap (OStheap[]) is out of range.
Possible Error Effect:	Without defensive checks, this error would either make it impossible for OSDS to add new entries to the heap (that is, schedule timers) or would cause them to over/under index OStheap[] when trying to do so. However, in practice, this error will cause asserts from OS which will schedule the TCB audit in directed mode.
Corrective Action Taken:	If the audit is not already running in directed mode it will escalate. In any case, OStheap[] will be reheaped.
Manual Action Required:	None.
Dump Description:	
	<i>Error Address</i> = NA.
	<i>Bad Data</i> = Index to the first available heap entry (OSfftheap).
	<i>Good Data</i> = OSTCBNUM+1 (Maximum legal value for OSfftheap).
	<i>Logical Index</i> = NA.
	<i>Dump</i> = None.

Table 6 Error Code: HEAP_GOFF

Error Code:	HEAP_GOFF
Error Description:	A TCB in the heap has a go-off time (TCB.go_off) which is either too far in the future (> 2 days) or too far in the past (> 1 hour). Note that its state (TCB.tstate) is valid and not OSQTIMEOUT (see TCB BADTYPE error).
Possible Error Effect:	The timer may never get fired or will fire at the wrong time. This may have unpredictable effects on its owning process.
Corrective Action Taken:	If the audit is not already running in directed mode it will escalate. In any case, OStheap[] will have its bad entry cleared and then be reheaped.
Manual Action Required:	None.
Dump Description:	
	<i>Error Address</i> = Address of the TCB.
	<i>Bad Data</i> = The TCB's index in OStheap[] (TCB.thi).
	<i>Good Data</i> = The TCB's state (TCB.tstate).
	<i>Logical Index</i> = The TCB's index.
	<i>Dump</i> = The TCB.

Table 7 Error Code: HEAP_NULL

Error Code:	HEAP_NULL
Error Description:	An element of the heap (OSheap[]) was expected to be null (equal to OSNULL) but wasn't.
Possible Error Effect:	There are two possibilities here. Either OSfftheap has been corrupted but is still in range (see TCB HEAP_FREE error) or an entry in OSheap[] has not been cleared when it should have been. In the first case, timers which appear in the heap after OSfftheap will never be fired. This may have unpredictable effects on the owning process(es). In the second case, the TCB indicated by the bad heap entry may have never fired or will fire unintentionally at some future time.
Corrective Action Taken:	If the audit is not already running in directed mode it will escalate. In any case, OSheap[] will have its bad entry cleared and then be reheaped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	Index into OSheap[].
<i>Good Data</i> =	OSNULL (the expected value of the entry).
<i>Logical Index</i> =	Contents of the entry (the TCB index).
<i>Dump</i> =	The TCB.

Table 8 Error Code: HEAP_TO

Error Code:	HEAP_TO
Error Description:	An entry in the heap (OSheap[]) is either out of range (not a valid TCB index) or indicates a TCB which does not point back (TCB.thi is not the index of this heap entry).
Possible Error Effect:	Without defensive checks, this error could cause a timer to fire incorrectly (the timer indicated by the heap entry) and/or cause another timer to fail to fire (the timer which should have been in the heap). However, in practice, this error is likely to cause OS to assert and schedule the TCB audit in directed mode.
Corrective Action Taken:	If the audit is not already running in directed mode it will escalate. In any case, OSheap[] will have its bad entry cleared and then be reheaped.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB (if any) indicated by the heap entry.
<i>Bad Data</i> =	If the OSheap[] entry was in range, this will be the associated TCB's thi field. Otherwise it will be 0.
<i>Good Data</i> =	Index into OSheap[].
<i>Logical Index</i> =	Contents of the OSheap[] entry.
<i>Dump</i> =	The associated TCB (if any).

Table 9 Error Code: INV

Error Code:	INV
Error Description:	A TCB has been found which is not linked to any PCB or the free list.
Possible Error Effect:	The TCB is lost and inaccessible to the system. If enough TCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	The TCB is cleared and linked to the idle list. In particular, its owner field is set to OSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	0 (Indicates that the audit did not find the TCB on any list).
<i>Good Data</i> =	1 (Indicates that the audit should have found the TCB on some list).
<i>Logical Index</i> =	The TCB's index.
<i>Dump</i> =	The TCB.

Table 10 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	A TCB has been found which the audit did not see on any list. Before taking an INV error the audit will verify that the TCB is truly lost. For example, if the audit were running in segmented mode, the TCB may have changed lists while the audit took a timebreak. In this case, the audit would not have seen the TCB before now. The audit uses the TCB's owner field to determine which list to check. This error says that the owner field is out of range (not OSNULL and not a PCB index).
Possible Error Effect:	The TCB is effectively lost and inaccessible to the system. If enough TCBs became lost, overload could occur and OS performance would be severely affected.
Corrective Action Taken:	If possible, the TCB is unlinked from the list it's in (the linkages of the TCBs it points to are reset). In any case, the TCB is cleared and linked to the idle list. In particular, its owner field is set to OSNULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the TCB.
<i>Bad Data =</i>	Contents of the TCB's owner field.
<i>Good Data =</i>	OSNULL.
<i>Logical Index =</i>	The TCB's index.
<i>Dump =</i>	The TCB.

Table 11 Error Code: KEY

Error Code:	KEY
Error Description:	A TCB's key (TCB.tcbkey) did not match the key of the TCB tuple.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	TCB.tcbkey is set to the tuple key value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Key from the TCB (TCB.tcbkey).
<i>Good Data =</i>	The key of the TCB tuple.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	The TCB.

Table 12 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	Either the tail of a TCB list was null when the head was not or the tail of a TCB list is out of range. The list owner (<i>Good Data</i>) is either a PCB or the free list (OSNULL). If it is a PCB, the out of range value is the PCB's "tcbn" field. If the owner is the free list, the out of range value is the last free TCB index (OSlftcb).
Possible Error Effect:	If the owner of the list is a PCB, timers associated with that process may fail to go off. The results of this depend on the process and timer types. If the free list is involved, this could result in lost TCBs which could, eventually, lead to OSDS TCB overloads.
Corrective Action Taken:	If the TCB list belongs to a PCB, that process will be purged. Otherwise, if the audit is running segmented, it will escalate to directed mode. If the audit is already running in directed mode it will attempt to rebuild the idle list from the top (OSftcb) down.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The out of range TCB index.
<i>Good Data =</i>	The list owner (a PCB index or OSNULL for the idle list).

<i>Logical Index</i> =	OSNULL.
<i>Dump</i> =	The PCB (if any) which owns the list.

Table 13 Error Code: LNKIALST

Error Code:	LNKIALST
Error Description:	While traversing a list, a TCB was found with an out of range llink field (not OSNULL and not a TCB index).
Possible Error Effect:	The list is broken. For the free list this could mean lost TCBs (leading to overload situations). For TCB lists attached to PCBs this problem could cause the owning process to lose timers. In practice, because of defensive checks in OS, this condition is much more likely to result in asserts from OS which schedule the TCB audit in directed mode. It is also likely that TCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of TCBs and the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by first verifying OSftcb (index to the first free TCB) and, if it is ok, verifying the list as far as possible from the top down and then linking with the segment (bottom up) the audit has already checked. If the list is attached to a PCB the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	The TCB's llink field.
<i>Good Data</i> =	OSTCBNUM (llink should have been OSNULL or less than this value).
<i>Logical Index</i> =	The TCB index.
<i>Dump</i> =	The TCB.

Table 14 Error Code: MISSING

Error Code:	MISSING
Error Description:	The meaning of this error depends on the value of <i>Error Address</i> : AUPCBI: The audit was unable to access a PCB tuple even though the key was valid. AUTCBI: The audit was unable to access a TCB tuple even though the key was valid.
Possible Error Effect:	Effectively, the tuple is lost to the system and can not be used.
Corrective Action Taken:	The audit will attempt to insert the missing tuple into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Tuple type indicator.
<i>Bad Data</i> =	Value returned from database access attempt.
<i>Good Data</i> =	The current maximum valid tuple key + 1.
<i>Logical Index</i> =	Index of the tuple.
<i>Dump</i> =	NA.

Table 15 Error Code: PARENT

Error Code:	PARENT
Error Description:	This error can only occur when the audit is running in segmented mode. During a timebreak, the "parent" of a heap entry has been corrupted (that is, the entry no longer contains a valid TCB index).
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will be restarted in directed mode. Note that it will then almost certainly find errors and take recovery.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the TCB associated with the heap entry whose parent has been corrupted.
<i>Bad Data</i> =	Contents of the parent entry in OSheap[].
<i>Good Data</i> =	OSTCBNUM (Maximum legal value for the parent entry).
<i>Logical Index</i> =	Index of the TCB associated with the heap entry whose parent has been corrupted.
<i>Dump</i> =	The TCB associated with the heap entry whose parent has been corrupted.

Table 16 Error Code: PROC_DED

Error Code:	PROC_DED
Error Description:	A dedicated timer is being used invalidly. That is, either one of the two situations has occurred: - A PCB is idle (PCB.state = OSEMPY) or is not using its dedicated timer (PCB.timeout OSNULL) but the dedicated timer is not idle (TCB.owner != OSNULL or TCB.tstate = OSEMPY). - A PCB is using its dedicated timer (PCB.state != OSEMPY and PCB.timeout != OSNULL) but either PCB.timeout is corrupt (!= the PCB's index) or the timer's owner field is incorrect (TCB.owner != PCB index) or the timer's state is incorrect (TCB.tstate != OSQTIMEOUT).
Possible Error Effect:	In the first case, it is possible that some other process is using the TCB. If the process to which this timer is dedicated decides to make use of it, the process which is using it invalidly will lose it. In the second case, the owning PCB may take the timer away from some other process which is (invalidly) using it. Or, the owning PCB may be unable to make use of the timer or will lose the timer itself. The loss of a timer by any process may cause unpredictable results for that process.
Corrective Action Taken:	The timer in question will be idled (but not returned to the free list since it is dedicated to a particular PCB). If the owning PCB is in use, its associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the dedicated TCB.
<i>Bad Data</i> =	The owning PCB's timeout field (PCB.timeout).
<i>Good Data</i> =	The owning PCB's state (PCB.state).
<i>Logical Index</i> =	Index of the TCB (this is also the PCB's index).
<i>Dump 1</i> =	The TCB.
<i>Dump 2</i> =	The PCB.

Table 17 Error Code: PT2PT

Error Code:	PT2PT
Error Description:	While traversing a list, a TCB was found whose llink field indicated a TCB whose rlink field was not equal to the first TCB's index.
Possible Error Effect:	The list is broken. For the free list this could mean lost TCBs (leading to overload situations) or non-idle TCBs being taken for new processes (affecting the process to which they rightfully belong). For TCB lists attached to PCBs this problem could cause the owning process to lose timers or falsely believe it has timers which actually belong to the idle list or other processes. In practice, because of defensive checks in OS, this condition is much more

	likely to result in asserts from OS which schedule the TCB audit in directed mode. It is also likely that TCB "INV" errors will be seen along with this one.
Corrective Action Taken:	If this is the free list of TCBs and the audit is running in segmented mode, it will escalate to directed mode and rebuild the list. If the audit is already running in directed mode, it will rebuild the list right away. The list will be rebuilt by first verifying OSftcb (index to the first free TCB) and, if it is ok, verifying the list as far as possible from the top down. If the list is attached to a PCB the associated process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	The TCB's link field.
<i>Good Data</i> =	The link field of the TCB indexed by <i>Bad Data</i> .
<i>Logical Index</i> =	The TCB's index.
<i>Dump</i> =	The TCB.

Table 18 Error Code: STATE

Error Code:	STATE
Error Description:	A TCB was found on the idle list with a state other than TEMPT (TCB.tstate != TEMPT).
Possible Error Effect:	Varies according to conditions but a TCB INV error is very likely to accompany this error.
Corrective Action Taken:	The audit will relink the idle list around the bad TCB. If the TCB truly belongs on the idle list, an INV error will find it, clear it and relink it there.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	The TCB's state (TCB.tstate).
<i>Good Data</i> =	The correct state (TEMPT).
<i>Logical Index</i> =	Index of the TCB.
<i>Dump</i> =	The TCB.

Table 19 Error Code: TIMEORDER

Error Code:	TIMEORDER
Error Description:	A TCB was found in the heap (OSheap[]) with a go-off time (TCB.go_off) earlier than its "parent" TCB in the heap.
Possible Error Effect:	The heap is out of order. This means that timers will not go off in their expected sequence. This may have unpredictable effects on the owning processes.
Corrective Action Taken:	If the audit is running in segmented mode, it will be restarted in directed mode. If it is already running in directed mode, it will reheap OSheap[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB which appeared out of order.
<i>Bad Data</i> =	Index of the parent TCB.
<i>Good Data</i> =	Index to the TCB's entry in OSheap[] (TCB.thi).
<i>Logical Index</i> =	Index of the TCB.
<i>Dump 1</i> =	The TCB.
<i>Dump 2</i> =	The parent TCB.

Table 20 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	A non-idle TCB's heap index is incorrect. That is, TCB.thi indexes an entry in the heap (OSheap[]) which does not contain the index of the TCB.
Possible Error Effect:	The timer will not fire as expected or at all. This may have unpredictable effects on the owning process.
Corrective Action Taken:	The owning process will be purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB.
<i>Bad Data</i> =	The TCB's index into the heap (TCB.thi).

<i>Good Data</i> =	The contents of the heap entry (OSheap[TCB.thi]).
<i>Logical Index</i> =	Index of the TCB.
<i>Dump</i> =	The TCB.

Table 21 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, an out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will idle the last TCB it was looking at (if any), abort and be restarted in directed mode. Otherwise it will just be restarted in directed mode.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TCB (if any).
<i>Bad Data</i> =	NA
<i>Good Data</i> =	NA
<i>Logical Index</i> =	Index of the TCB (if any).
<i>Dump</i> =	The TCB (if any).

TCPBLK

Software Release: 5E13 and later

1. TCP DATA BLOCK (TCPBLK) AUDIT

The TCPBLK audit verifies the TCP data block relation (RLTCPBLK). RLTCPBLK is a dynamic relation that contains channel specific data used by OAP LSP.

2. DATA STRUCTURES

For details on the fields contained in the RLTCPBLK relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	Using rITCPBLK's local port (rITCPBLK.local_port), rITCPBLK's remote address (rITCPBLK.rmt_addr) and rITCPBLK's remote port (rITCPBLK.rmt_port) as keys to the RLTCPCONN relation, a rITCPCONN tuple was read that did not contain the rITCPBLK key in it (rITCPCONN.local_name).
Possible Error Effect:	The TCP application will not be able to find the connection information it needs to receive data to the OAP.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPCONN's key to the RLTCPBLK relation (rITCPCONN.local_name).
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump =</i>	rITCPBLK tuple.

Table 2 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The rITCPBLK tuple is on the busy list, however the rITCPBLK's state (rITCPBLK.state) is not set to TPBUSY.
Possible Error Effect:	The TCP application may process the connection's packets incorrectly causing the data to be lost and data transferred using TCP to not be correctly delivered to other applications using TCP such as the OAP.
Corrective Action Taken:	The audit sets the state to TPBUSY.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPBLK's state (rITCPBLK.state).
<i>Good Data =</i>	TPBUSY.
<i>Logical Index =</i>	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump =</i>	rITCPBLK tuple.

Table 3 Error Code: BUSY3MAP

Error Code:	BUSY3MAP
Error Description:	The rITCPBLK's delay queue (rITCPBLK.delay_queue.st_headk) is linked to rITCPPKT tuples or transmit queue (rITCPBLK.xmit_queue.st_headk) is not linked to rITCPPKT tuples when the rITCPBLK's owner is the retransmit list (rITCPBLK.bi_link.st_ownk == DBTCPBLXMIT). A TCP connection in the retransmit state cannot have data awaiting transmission.
Possible Error Effect:	The TCP application either has the rITCPBLK tuple on the wrong list or has the RLTCPPKT resource linked incorrectly to the rITCPBLK tuple. Either may lead to lost resources and exhaustion of the RLTCPPKT resource.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPPKT's physical key from the transmit queue (rITCPBLK.xmit_queue.st_headk).
<i>Good Data =</i>	rITCPPKT's physical key from the delay queue (rITCPBLK.delay_queue.st_headk).
<i>Logical Index =</i>	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump =</i>	rITCPBLK tuple.

Table 4 Error Code: BUSY4MAP

Error Code:	BUSY4MAP
Error Description:	The rITCPBLK's delay queue (rITCPBLK.delay_queue.st_headk) is not linked to rITCPPKT tuples or transmit queue (rITCPBLK.xmit_queue.st_headk) is linked to rITCPPKT tuples when the rITCPBLK's owner is the delay list (rITCPBLK.bi_link.st_ownk == DBTCPBLDLY). A TCP connection in the delay state has been delayed and cannot have data to transmit.
Possible Error Effect:	The TCP application either has the rITCPBLK tuple on the wrong list or has the RLTCPPKT resource linked incorrectly to the rITCPBLK tuple. Either may lead to lost resources and exhaustion of the RLTCPPKT resource.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPPKT's physical key from the transmit queue (rITCPBLK.xmit_queue.st_headk).
<i>Good Data =</i>	rITCPPKT's physical key from the delay queue (rITCPBLK.delay_queue.st_headk).
<i>Logical Index =</i>	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump =</i>	rITCPBLK tuple.

Table 5 Error Code: BUSY5MAP

Error Code:	BUSY5MAP
Error Description:	The rITCPBLK's delay queue (rITCPBLK.delay_queue.st_headk) is not linked to rITCPPKT tuples or transmit queue (rITCPBLK.xmit_queue.st_headk) is not linked to rITCPPKT tuples when the rITCPBLK's owner is the delay list (rITCPBLK.bi_link.st_ownk == DBTCPBLRTDL). A TCP connection in the retransmit delay state has been delayed and is retransmitting data.
Possible Error Effect:	The TCP application either has the rITCPBLK tuple on the wrong list or has the RLTCPPKT resource linked incorrectly to the rITCPBLK tuple. Either may lead to lost resources and exhaustion of the RLTCPPKT resource.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPPKT's physical key from the transmit queue (rITCPBLK.xmit_queue.st_headk).

<i>Good Data</i> =	rITCPPKT's physical key from the delay queue (rITCPBLK.delay_queue.st_headk).
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 6 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rITCPBLK's terminal process (rITCPBLK.pid) is not alive. This process must be alive at all times, except when rITCPBLK's establish state (rITCPBLK.estab_state) is TPTIME_WAIT.
Possible Error Effect:	The TCP application has resources that are allocated but do not have a process using them. They are considered lost and may lead to resource exhaustion.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPBLK tuple.
<i>Bad Data</i> =	rITCPBLK's process number (rITCPBLK.pid.procno).
<i>Good Data</i> =	rITCPBLK's uniqueness number (rITCPBLK.pid.uniq).
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 7 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The rITCPBLK tuple is linked to the idle list, however its linkages are not null. An idle resource cannot have anything linked to it.
Possible Error Effect:	The linked resources may be lost leading to resource exhaustion.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPBLK tuple.
<i>Bad Data</i> =	Number of non-null linkages.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 8 Error Code: IDL_STATE

Error Code:	IDL_STATE
Error Description:	The rITCPBLK tuple is on the idle list, however the rITCPBLK's state (rITCPBLK.state) is not set to TPIDLE.
Possible Error Effect:	The TCP application may reuse the rITCPBLK tuple causing any existing connection to be lost.
Corrective Action Taken:	The audit sets the state to TPIDLE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPBLK tuple.
<i>Bad Data</i> =	rITCPBLK's state (rITCPBLK.state).
<i>Good Data</i> =	TPIDLE.
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 9 Error Code: INT_STATE

Error Code:	INT_STATE
Error Description:	The rITCPBLK tuple is on the retransmit list, however the rITCPBLK's state (rITCPBLK.state) is not set to TPRETRANS.
Possible Error Effect:	The TCP application may process the connection's packets incorrectly causing the data to be lost and data transferred using TCP to not be correctly delivered to other applications using TCP such as the OAP.
Corrective Action Taken:	The audit sets the state to TPRETRANS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPBLK tuple.

<i>Bad Data</i> =	rITCPBLK's state (rITCPBLK.state).
<i>Good Data</i> =	TPRETRANS.
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 10 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The rITCPBLK tuple is on the delay list, however the rITCPBLK's state (rITCPBLK.state) is not set to TPDELAY.
Possible Error Effect:	The TCP application may process the connection's packets incorrectly causing the data to be lost and data transferred using TCP to not be correctly delivered to other applications using TCP such as the OAP.
Corrective Action Taken:	The audit sets the state to TPDELAY.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPBLK tuple.
<i>Bad Data</i> =	rITCPBLK's state (rITCPBLK.state).
<i>Good Data</i> =	TPDELAY.
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 11 Error Code: NOMATCH

Error Code:	NOMATCH
Error Description:	Using rITCPBLK's local port (rITCPBLK.local_port), rITCPBLK's remote address (rITCPBLK.rmt_addr) and rITCPBLK's remote port (rITCPBLK.rmt_port) as keys to the RLTCPCONN relation, a rITCPCONN tuple could not be read.
Possible Error Effect:	The TCP application will not be able to find the connection information it needs to receive data to the OAP.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of rITCPBLK's local port (rITCPBLK.local_port).
<i>Bad Data</i> =	The address of rITCPBLK's remote address (rITCPBLK.rmt_addr).
<i>Good Data</i> =	The address of rITCPBLK's remote port (rITCPBLK.rmt_port).
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 12 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The rITCPBLK's establish state (rITCPBLK.estab_state) has been set to TPTIME_WAIT for more than 2 minutes. The resource is considered lost, because it is still waiting.
Possible Error Effect:	The TCP application has resources that are allocated but are not being used. They are considered lost and may lead to resource exhaustion.
Corrective Action Taken:	The audit idles the tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPBLK tuple.
<i>Bad Data</i> =	rITCPBLK's quarantine state (rITCPBLK.quarstate).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump</i> =	rITCPBLK tuple.

Table 13 Error Code: QUE_BUSY

Error Code:	QUE_BUSY
Error Description:	The rITCPBLK's delay queue (rITCPBLK.delay_queue.st_headk) or transmit queue (rITCPBLK.xmit_queue.st_headk) is linked to rITCPPKT tuples when the rITCPBLK's owner is the busy list (rITCPBLK.bi_link.st_ownk == DBTCPBLBUSY). An busy TCP connection cannot be delayed and has transmitted its data.

Possible Error Effect:	The TCP application either has the rITCPBLK tuple on the wrong list or has the RLTCPPKT resource linked incorrectly to the rITCPBLK tuple. Either may lead to lost resources and exhaustion of the RLTCPPKT resource.
Corrective Action Taken:	The audit idles the tuple and may reset the TCP connection.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPPKT's physical key from the transmit queue (rITCPBLK.xmit_queue.st_headk).
<i>Good Data =</i>	rITCPPKT's physical key from the delay queue (rITCPBLK.delay_queue.st_headk).
<i>Logical Index =</i>	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump =</i>	rITCPBLK tuple.

Table 14 Error Code: STATE

Error Code:	STATE
Error Description:	The rITCPBLK tuple is on the delay retransmit list, however the rITCPBLK's state (rITCPBLK.state) is not set to TPDEL_RETRANS.
Possible Error Effect:	The TCP application may process the connection's packets incorrectly causing the data to be lost and data transferred using TCP to not be correctly delivered to other applications using TCP such as the OAP.
Corrective Action Taken:	The audit sets the state to TPDEL_RETRANS.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPBLK tuple.
<i>Bad Data =</i>	rITCPBLK's state (rITCPBLK.state).
<i>Good Data =</i>	TPDEL_RETRANS.
<i>Logical Index =</i>	rITCPBLK's key (rITCPBLK.con_id).
<i>Dump =</i>	rITCPBLK tuple.

TCPCB-A

Software Release: 5E12 only

1. ETHERNET TRANSMISSION CONTROL PROTOCOL CONTROL BLOCK (TCPCB) AUDIT

Transmission control protocol/internet protocol (TCP/IP) is used to transmit and receive data through an ethernet network. The TCPCBs are used to represent per-connection information such as sequence numbers and protocol state. The TCPCBs are either idle and are linked to the idle list or linked and are linked to an internet protocol control block (INPCB). The audit verifies the linkages among CMtcb[], CMinpcb[] and CMidletcb and data within every entry in CMtcb[].

2. DATA STRUCTURES

The following data are used by the TCPCB audit:

CMtcb[] = TCP control block.

CMinpcb[] = Internet protocol control block.

CMidletcb = TCP control block idle list.

3. AUDIT CHECKS

The TCPCB audit makes the following checks:

BADRANGE_INDEX	= Check an index into CMtcb[].
BADSTATE	= Range check the state of the CMtcb[] entry.
BUSYIDLE	= Check a CMtcb[] entry that is allocated but not linked.
IDLE_LINK	= Check an in-use list entry.
IDLE_OWNER	= Check the idle list.
INV_STATE	= Verify values of state.
INV_TIMER	= Verify value in one of t_timer[] for established t_state.
IS_NOTUSED	= Check an unused list entry.
NONNULL	= Verify value in one of t_timer[] closed t_state.
NON_NULL	= Check an idle list entry.
OUTOFRANGE	= Validity check CMtcb[].t_state.
ZAP	= Check for system interrupt handling.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	This CMtpcb[] entry should be linked to an entry in CMinpcb[]. One of the following has occurred: - CMtpcb[index].t_qindx is not CMNIL_NDX, - CMtpcb[index].t_inpcb is out of range or - CMinpcb[CMtpcb[index].t_inpcb].inp_ppcb does not equal the index into CMtpcb[].
Possible Error Effect:	The MELNK is removed and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMtpcb[index].t_inpcb.
<i>Bad Data =</i>	CMtpcb[index].t_qindx.
<i>Good Data =</i>	CMtpcb[CMtpcb[index].t_inpcb].inp_ppcb.
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 2 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The state of an entry in CMtpcb[] is out of range.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMtpcb[index].t_cbstate.
<i>Good Data =</i>	CMINCBMAX
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 3 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The CMtpcb[] is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	CMtpcb[index].t_cbstate..
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 4 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An TCPCB in-use idle stack entry (index $\leq$ idle stack <i>toppt</i>) is corrupt. One or more of the following is true ("idle_tcbcb" refers to the contents of the idle stack entry <i>tpcb[index]</i>): idle_tcbcb $\geq$ max

	The TCPCB indexed the contents of idle stack entry <i>tcpcb[idle_tcpcb]</i> and does not point back to the idle list (<i>TCPCB.t_qindx != idle_tcpcb</i>).
Possible Error Effect:	TCPCB may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry (index = idle stack <i>toppt</i>), the audit clears the idle stack entry (<i>tcpcb[index] = CMNIL_NDX</i>) and decrements the count of idle TCPCBs (<i>toppt</i> is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets <i>tcpcb[index] = sopcb[toppt]</i>, relinks (<i>TCPCB.t_qindx = index</i>), clears the old entry (<i>so_pcb[CMidletcpb.toppt] = CMNIL_NDX</i>), and decrements the number of idle TCPCBs (decrements <i>toppt</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Indicates TCPCB idle list (<i>CMidletcpb</i>).
<i>Bad Data =</i>	Value from idle stack entry (<i>TCPCB index</i>).
<i>Good Data =</i>	value of "max".
<i>Logical Index =</i>	Index into idle stack.
<i>Dump =</i>	NA.

Table 5 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A <i>CMtcpcb[]</i> idle stack is corrupt. One or more of the following is true: - The idle stack <i>toppt</i> value is out-of-range (<i>> max</i> or <i>< min</i>). - The idle stack entry <i>tcpcb[toppt] = CMNIL_NDX</i>. - The idle stack <i>toppt</i> is <i>< max</i> and the idle stack entry <i>tcpcb[toppt+1] != CMNIL_NDX</i>.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit clears the <i>CMtcpcb[]</i> idle stack (<i>toppt = 0</i>) and rebuilds the idle list (links all entries whose <i>t_cbstate</i> is <i>CMINCBFREE</i> to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	<i>CMidletcpb.tcpb[CMidletcpb.toppt]</i>
<i>Bad Data =</i>	<i>CMidletcpb.tcpb[CMidletcpb.toppt+1]</i> .
<i>Good Data =</i>	<i>CM_NUMTCPCB</i> .
<i>Logical Index =</i>	<i>CMidletcpb.toppt</i>
<i>Dump =</i>	NA.

Table 6 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The state of a <i>CMtcpcb[]</i> is one not recognized by the audit.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (<i>CMidletcpb</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	<i>CMtcpcb[index].t_cbstate</i> .
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into <i>CMtcpcb[]</i> .
<i>Dump 1 =</i>	First half of contents of <i>CMtcpcb[index]</i> .
<i>Dump 2 =</i>	Second half of contents of <i>CMtcpcb[index]</i> .

Table 7 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The value of <i>t_state</i> in a CMtcb[] entry is CMTCPD_ESTABLISHED but the value of one of the <i>t_timer[]</i> is 0 and should not be 0.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMtcb[index].t_cbstate.
<i>Bad Data</i> =	CMTCPD_KEEP (index into t_timer[]).
<i>Good Data</i> =	CMtcb[index].t_state.
<i>Logical Index</i> =	Index into CMtcb[].
<i>Dump 1</i> =	First half of contents of CMtcb[index].
<i>Dump 2</i> =	Second half of contents of CMtcb[index].

Table 8 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused TCPCB idle stack entry is not null (index < <i>toppt</i> and <i>tcpcb[index] != CMNIL_NDX</i>).
Possible Error Effect:	TCPCBs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the TCPCB idle stack entry (<i>tcpcb[index] = CMNIL_NDX</i>).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Indicate TCPCB idle list (CMidletcb).
<i>Bad Data</i> =	Contents of the idle stack entry (<i>tcpcb[index]</i>).
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into the idle stack.
<i>Dump</i> =	NA.

Table 9 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The value of <i>t_state</i> in a CMtcb[] entry is CMTCPD_CLOSED but the value of one of the <i>t_timer[]</i> is not 0 and should be 0.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Timer index.
<i>Bad Data</i> =	CMtcb.t_timer[Timer index].
<i>Good Data</i> =	CMtcb[index].t_state.
<i>Logical Index</i> =	Index into CMtcb[].
<i>Dump 1</i> =	First half of contents of CMtcb[index].
<i>Dump 2</i> =	Second half of contents of CMtcb[index].

Table 10 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The value of <i>t_cbstate</i> in a CMtcb[] entry is CMINCBFREE but the value of one of its fields is not the null value.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMidletcb.tcpcb[CMtcb[t_qindx]].
<i>Good Data</i> =	CMtcb[index].t_cbstate.
<i>Logical Index</i> =	Index into CMtcb[].
<i>Dump 1</i> =	First half of contents of CMtcb[index].
<i>Dump 2</i> =	Second half of contents of CMtcb[index].

Table 11 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The CMtcbp[index].t_cbstate is set to CMINCBLSNKD. The audit detects that CMtcbp[index].t_state is invalid.
Possible Error Effect:	TCP message transport may not be working correctly.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletcbp).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMtcbp[index].t_cbstate.
<i>Bad Data =</i>	CMtcbp[index].t_state.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMtcbp[].
<i>Dump 1 =</i>	First half of contents of CMtcbp[index].
<i>Dump 2 =</i>	Second half of contents of CMtcbp[index].

Table 12 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle stack entry, if any, that was being audited when the stimulus occurred. • Recovers the CMtcbp[] entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMidletcbp.tcpb[CM_NUMTCPCB - AUcur_audstat.position].
<i>Good Data =</i>	CM_NUMTCPCB - AUcur_audstat.position.
<i>Logical Index =</i>	AUcur_audstat.position
<i>Dump 1 =</i>	First half of contents of CMtcbp[AUcur_audstat.position].
<i>Dump 2 =</i>	Second half of contents of CMtcbp[AUcur_audstat.position].

TCPCB-B

Software Release: 5E13 and later

1. ETHERNET TRANSMISSION CONTROL PROTOCOL CONTROL BLOCK (TCPCB) AUDIT

Transmission control protocol/internet protocol (TCP/IP) is used to transmit and receive data through an ethernet network. The TCPCBs are used to represent per-connection information such as sequence numbers and protocol state. The TCPCBs are either idle and are linked to the idle list or linked and are linked to an internet protocol control block (INPCB). The audit verifies the linkages among CMtcb[], CMinpcb[] and CMidletcb and data within every entry in CMtcb[].

2. DATA STRUCTURES

The following data are used by the TCPCB audit:

CMtcb[] = TCP control block.

CMinpcb[] = Internet protocol control block.

CMidletcb = TCP control block idle list.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BADRANGE_INDEX

Error Code:	BADRANGE_INDEX
Error Description:	This CMtcb[] entry should be linked to an entry in CMinpcb[]. One of the following has occurred: - CMtcb[index].t_qindx is not CMNIL_NDX, - CMtcb[index].t_inpcb is out of range or - CMinpcb[CMtcb[index].t_inpcb].inp_ppcb does not equal the index into CMtcb[].
Possible Error Effect:	The MELNK is removed and real time call detail (RTCD) records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMtcb[index].t_inpcb.
<i>Bad Data =</i>	CMtcb[index].t_qindx.
<i>Good Data =</i>	CMinpcb[CMtcb[index].t_inpcb].inp_ppcb.
<i>Logical Index =</i>	Index into CMtcb[].
<i>Dump 1 =</i>	First half of contents of CMtcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtcb[index].

Table 2 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	The state of an entry in CMtpcb[] is out of range.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMtpcb[index].t_cbstate.
<i>Good Data =</i>	CMINCBMAX
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 3 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The CMtpcb[] is allocated but not linked. Application code allocated the entry but was not able to complete linking it.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	CMtpcb[index].t_cbstate..
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 4 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A TPCB in-use idle stack entry ($\text{index} \leq \text{CMidletpcb.toppt}$) is corrupt. One or more of the following is true: • $\text{CMidletpcb.tpcb}[\text{index}] \geq \text{CM\_NUMTCPCB}$ • $\text{CMidletpcb.tpcb}[\text{index}] \geq 0$ • $\text{CMtpcb}[\text{CMidletpcb.tpcb}[\text{index}]].\text{t\_qindx} \neq \text{index}$ into $\text{CMidletpcb.tpcb}[]$.
Possible Error Effect:	TCPCB may not be allocatable. If so, the switching module (SM) may lose ethernet connectivity.
Corrective Action Taken:	If this is the first in-use idle list entry ($\text{index} = \text{CMidletpcb.toppt}$), the audit clears the idle stack entry ($\text{CMidletpcb}[\text{index}] = \text{CMNIL\_NDX}$) and decrements the count of idle TCPCBs (CMidletpcb.toppt is decremented). Otherwise, the audit puts a good value into the entry. That is, the audit sets $\text{CMidletpcb.tpcb}[\text{index}] = \text{CMidletpcb.tpcb}[\text{CMidletpcb.toppt}]$, relinks the TCPCB ($\text{CMtpcb}[\text{CMidletpcb.tpcb}[\text{CMidletpcb.toppt}]].\text{t\_qindx} = \text{index}$), clears the old entry ($\text{CMidletpcb.tpcb}[\text{CMidletpcb.toppt}] = \text{CMNIL\_NDX}$) and decrements the number of idle TCPCBs (decrements CMidletpcb.toppt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	AUTCTYPE.
<i>Bad Data =</i>	Index of next available idle CMtpcb[].
<i>Good Data =</i>	CM_NUMTCPCB.
<i>Logical Index =</i>	Index into CMidletpcb.tpcb[].
<i>Dump =</i>	NA.

Table 5 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A CMtcb[] idle stack is corrupt. One or more of the following is true: - The idle stack CMidletpcb.toppt value is out-of-range (> max or < min). - The idle stack entry CMidletpcb.tpcb[CMidletpcb.toppt] = CMNIL_NDX. - The idle stack CMidletpcb.toppt is < max and the idle stack entry CMidletpcb.tpcb[CMidletpcb.toppt+1] != CMNIL_NDX.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit sets CMidletpcb.toppt = 0 and rebuilds the idle list (links all entries whose CMtcb[].t_cbstate is CMINCBFREE to the idle list).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMidletpcb.tpcb[CMidletpcb.toppt]
<i>Bad Data =</i>	CMidletpcb.tpcb[CMidletpcb.toppt+1].
<i>Good Data =</i>	CM_NUMTCPCB.
<i>Logical Index =</i>	CMidletpcb.toppt
<i>Dump =</i>	NA.

Table 6 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The state of a CMtcb[] is one not recognized by the audit.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMtcb[index].t_cbstate.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMtcb[].
<i>Dump 1 =</i>	First half of contents of CMtcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtcb[index].

Table 7 Error Code: INV_TIMER

Error Code:	INV_TIMER
Error Description:	The value of CMtcb[index].t_state is CMTCP_ESTABLISHED but the value of one of its CMtcb[index].t_timer[] is 0 and should not be 0.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMtcb[index].t_cbstate.
<i>Bad Data =</i>	CMTCP_KEEP (index into t_timer[]).
<i>Good Data =</i>	CMtcb[index].t_state.
<i>Logical Index =</i>	Index into CMtcb[].
<i>Dump 1 =</i>	First half of contents of CMtcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtcb[index].

Table 8 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused TCPCB idle stack entry (index > CMidletpcb.toppt) is not null (CMidletpcb.tpcb[index] != CMNIL_NDX and index < CMidletpcb.toppt)
Possible Error Effect:	TCPCBs may not be allocatable. If so, the SM may lose ethernet connectivity.
Corrective Action Taken:	The audit clears the TCPCB idle stack entry (CMidletpcb.tpcb[index] =

	CMNIL_NDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	AUTCTYPE.
<i>Bad Data =</i>	Index of next available idle CMtpcb[].
<i>Good Data =</i>	CMNIL_NDX.
<i>Logical Index =</i>	Index into CMidletpcb.tcpb[].
<i>Dump =</i>	NA.

Table 9 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	The value of CMtpcb[index].t_state is CMTCP_CLOSED but the value of one of its CMtpcb[index].t_timer[] is not 0 and should be 0.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Timer index.
<i>Bad Data =</i>	CMtpcb.t_timer[Timer index].
<i>Good Data =</i>	CMtpcb[index].t_state.
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 10 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The value of CMtpcb[index].t_cbstate is CMINCBFREE but the value of one of its other fields is not the null value.
Possible Error Effect:	The MELNK is removed and RTCD records may be lost.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CMidletpcb.tcpb[CMtpcb[index].t_qindx].
<i>Good Data =</i>	CMtpcb[index].t_cbstate.
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 11 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The CMtpcb[index].t_cbstate is set to CMINCBSLNKD. The audit detects that CMtpcb[index].t_state is invalid.
Possible Error Effect:	TCP message transport may not be working correctly.
Corrective Action Taken:	The audit has all the fields set to default values and puts the entry on the idle stack (CMidletpcb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	CMtpcb[index].t_cbstate.
<i>Bad Data =</i>	CMtpcb[index].t_state.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Index into CMtpcb[].
<i>Dump 1 =</i>	First half of contents of CMtpcb[index].
<i>Dump 2 =</i>	Second half of contents of CMtpcb[index].

Table 12 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.

Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears the idle list entry, if any, that was being audited when the stimulus occurred. • Recovers the CMtcpcb[] entry, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the entry from any other data structures, sets all fields to default values and puts the entry on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMidletcpb.tcpb[CM_NUMTCPCB - AUcur_audstat.position].
<i>Good Data</i> =	CM_NUMTCPCB - AUcur_audstat.position.
<i>Logical Index</i> =	AUcur_audstat.position
<i>Dump 1</i> =	First half of contents of CMtcpcb[AUcur_audstat.position].
<i>Dump 2</i> =	Second half of contents of CMtcpcb[AUcur_audstat.position].

TCPCONN

Software Release: 5E13 and later

1. TCP CONNECTION (TCPCONN) AUDIT

The TCPCONN audit verifies data in the RLTCPCONN relation. The relation RLTCPCONN contains data on tcp connection.

2. DATA STRUCTURES

For details on the fields contained in the RLTCPCONN relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The rITCPCONN's local_name is used as a key to read the corresponding rITCPBLK tuple. However, the rITCPBLK's key's for the rITCPCONN tuple (rITCPBLK.local_port, rITCPBLK.rmt_addr or rITCPBLK.rmt_port) did not equal the rITCPCONN's keys(rITCPCONN.local_port, rITCPCONN.rmt_addr or rITCPCONN.rmt_port). Alternatively, the rITCPBLK's state (rITCPBLK.state) could be idle, disallowing the existence of this rITCPCONN tuple.
Possible Error Effect:	The TCP application will not be able to receive data blocks from the OAP. If left uncorrected, the connection will eventually be reset.
Corrective Action Taken:	The rITCPCONN tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPCONN tuple.
<i>Bad Data =</i>	rITCPBLK's remote port (rITCPBLK.rmt_port).
<i>Good Data =</i>	rITCPBLK's remote address (rITCPBLK.rmt_addr).
<i>Logical Index =</i>	rITCPCONN's key (rITCPCONN.local_port).
<i>Dump 1 =</i>	rITCPCONN tuple.
<i>Dump 2 =</i>	rITCPBLK tuple.

TCPPKT

Software Release: 5E13 and later

1. TCP PACKET (TCPKPT) AUDIT

The TCPKPT audit verifies data in the RLTCPPKT relation. The relation RLTCPPKT contains data to be sent/received over a TCP/IP connection.

2. DATA STRUCTURES

For details on the fields contained in the RLTCPPKT relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	Neither of the owning rITCPBLK's linkage to the ends of rITCPPKT's queue reference the rITCPPKT tuple, even though the rITCPPKT's linkage forward or backwards is null, indicating that it must be at either end of the rITCPPKT queue. Use the rITCPPKT's state (rITCPPKT.pkt_state) to determine which queue's linkages are in error.
Possible Error Effect:	The transmission control protocol (TCP) application may lose packets in the queue. Data will not be correctly delivered to applications such as the operator services position system (OSPS) administrative processor (OAP).
Corrective Action Taken:	The audit idles the rITCPPKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPPKT tuple.
<i>Bad Data =</i>	The owning rITCPBLK's linkage to the front of the rITCPPKT's queue.
<i>Good Data =</i>	The owning rITCPBLK's linkage to the back of the rITCPPKT's queue.
<i>Logical Index =</i>	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump =</i>	rITCPPKT tuple.

Table 2 Error Code: DEADEND

Error Code:	DEADEND
Error Description:	The owning rITCPBLK's ending sequence number (rITCPBLK.rcv_next) does not equal the sum of last rITCPPKT tuple's sequence number (rITCPPKT.begin_seq) and size (rITCPPKT.datasz) on the rITCPBLK's receive queue (rITCPBLK.rcv_queue).
Possible Error Effect:	The TCP application may lose packets or incorrectly order packets because the sequence numbers of the packets are incorrect. Data will not be correctly delivered to applications such as the OAP.
Corrective Action Taken:	The audit idles the rITCPPKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPPKT tuple.
<i>Bad Data =</i>	The owning rITCPBLK's key (rITCPBLK.con_id).

<i>Good Data</i> =	The owning rITCPBLK's receive next number (rITCPBLK.rcv_next).
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump</i> =	rITCPPKT tuple.

Table 3 Error Code: LISTEND_BAD

Error Code:	LISTEND_BAD
Error Description:	The owning rITCPBLK's ending sequence number (rITCPBLK.send_next) does not equal the sum of last rITCPPKT tuple's sequence number (rITCPPKT.begin_seq) and size (rITCPPKT.datasz) on the rITCPBLK's transmit queue (rITCPBLK.xmit_queue).
Possible Error Effect:	The TCP application may lose packets or send packets in the incorrect order because the sequence numbers of the packets are incorrect. Data will not be correctly delivered to applications such as the OAP.
Corrective Action Taken:	The audit idles the rITCPPKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPPKT tuple.
<i>Bad Data</i> =	The owning rITCPBLK's key (rITCPBLK.con_id).
<i>Good Data</i> =	The owning rITCPBLK's transmit next number (rITCPBLK.send_next).
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump</i> =	rITCPPKT tuple.

Table 4 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rITCPPKT tuple is on the idle list, however the rITCPPKT's state (rITCPPKT.pkt_state) is not set to TPIDLE.
Possible Error Effect:	The TCP application will not use this packet, possibly leading to resource exhaustion.
Corrective Action Taken:	The audit change the pkt_state to TPPKTIDLE.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPPKT tuple.
<i>Bad Data</i> =	rITCPPKT tuple's state (rITCPPKT.pkt_state).
<i>Good Data</i> =	TPPKTIDLE.
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump</i> =	rITCPPKT tuple.

Table 5 Error Code: QUE

Error Code:	QUE
Error Description:	The rITCPPKT tuple's state (rITCPPKT.pkt_state) does not equal the next rITCPPKT's state (rITCPPKT.pkt_state). A packet may be on the wrong queue.
Possible Error Effect:	The TCP application may reallocate or process the packet incorrectly causing the data to be lost or data to not be correctly delivered to applications such as the OAP.
Corrective Action Taken:	The audit idles the rITCPPKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPPKT tuple.
<i>Bad Data</i> =	The next rITCPPKT tuple's state (rITCPPKT.pkt_state).
<i>Good Data</i> =	rITCPPKT tuple's state (rITCPPKT.pkt_state).
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump</i> =	rITCPPKT tuple.

Table 6 Error Code: STATE

Error Code:	STATE
Error Description:	The rITCPPKT's state (rITCPPKT.pkt_state) is not active. The tuple may be idle and not linked to the idle list.
Possible Error Effect:	The TCP application will not be able to reuse this packet, possibly leading to resource exhaustion.
Corrective Action Taken:	The audit idles the rITCPPKT tuple.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPPKT tuple.
<i>Bad Data</i> =	The rITCPPKT's state (rITCPPKT.pkt_state).
<i>Good Data</i> =	Address of the owning TCPBLK tuple.
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump</i> =	rITCPPKT tuple.

Table 7 Error Code: TIMEORDER

Error Code:	TIMEORDER
Error Description:	The next rITCPPKT's sequence number (rITCPPKT.begin_seq) in the queue does not equal the sum of this rITCPPKT tuple's sequence number (rITCPPKT.begin_seq) and size (rITCPPKT.datasz). The rITCPPKT tuple is on the rITCPBLK's receive queue (rITCPBLK.rcv_queue) or transmit queue (rITCPBLK.xmit_queue).
Possible Error Effect:	The TCP application may lose packets or send packets in the incorrect order because the sequence numbers of the packets are incorrect. Data will not be correctly delivered to applications such as the OAP.
Corrective Action Taken:	The audit idles the rITCPPKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPPKT tuple.
<i>Bad Data</i> =	The next rITCPPKT's key (rITCPPKT.pktkey).
<i>Good Data</i> =	The next rITCPPKT's beginning sequence number (rITCPPKT.begin_seq).
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump</i> =	rITCPPKT tuple.

Table 8 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	The rITCPPKT's time stamp (rITCPPKT.time) indicates that the tuple has been in the rITCPBLK's receive queue (rITCPBLK.rcv_queue) or fragment queue (rITCPBLK.frag_queue) for more than two hours.
Possible Error Effect:	The TCP application will not be able to reuse this packet, possibly leading to resource exhaustion.
Corrective Action Taken:	The audit idles the rITCPPKT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITCPPKT tuple.
<i>Bad Data</i> =	Current time (OSGETTIME()).
<i>Good Data</i> =	Packet state (rITCPPKT.pkt_state).
<i>Logical Index</i> =	rITCPPKT's key (rITCPPKT.pktkey).
<i>Dump 1</i> =	The rITCPPKT tuple.
<i>Dump 2</i> =	The rITCPBLK tuple.

TCPRCV

Software Release: 5E13 and later

1. TCP RECEIVE DATA BLOCK (TCPRCV) AUDIT

The TCPRCV audit verifies the TCP receive data block relation (RLTCPRCV). This dynamic relation, RLTCPRCV, contains information about outstanding receive request.

2. DATA STRUCTURES

For details on the fields contained in the RLTCPRCV relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The transfer communication protocol (TCP) application is processing a request for a dead process.
Possible Error Effect:	The TCP application may exhaust the rITCPRCV resources, because dead processes still have this resource allocated.
Corrective Action Taken:	The audit idles the tuple and wakes any processes waiting for data being transferred by way of TCP. Because other processes may also be waiting to receive from the same connection, the audit attempts to repair the linkages in the receive request queue.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITCPRCV tuple.
<i>Bad Data =</i>	rITCPRCV's process number (rITCPRCV.pid.procno).
<i>Good Data =</i>	rITCPRCV's uniqueness number (rITCPRCV.pid.uniq).
<i>Logical Index =</i>	rITCPRCV's key(rITCPRCV.rcvid).
<i>Dump =</i>	rITCPRCV tuple.

TD

Software Release: 5E10 and later

1. TRANSMIT DESCRIPTOR (TD) AUDIT

Frames are used to transmit data through the quad linked packet switched (QLPS) network. TDs (transmit descriptors, elements of CMtduse[]) are used to represent frames which are to be transmitted. TDs are linked to frame descriptors and transmit lists.

2. DATA STRUCTURES

The following data is used by the TD audit:

CMbufidle = Buffer Idle List.
 CMspscdb[] = SPYDER-T Channel Data Block.
 CMsptidle[] = SPYDER-T TD Idle Stack.
 CMsptduse[] = SPYDER-T Transmit Descriptor.

3. AUDIT CHECKS

The following checks are performed by the TD audit:

BADUSE = Check for an invalid usage type in a TD.
 IDLE_LINK = Check for an invalid entry in the stack of idle TDs.
 IDLE_OWNER = Check IDLE TD for linkage and default values.
 IS_NOTUSED = Check for default field values in unused entry in TD idle stack.
 OWNER = Check for invalid idle stack head.
 OWNLINK = Check TD linkage with an FD.
 ZAP = A reset was taken during the audit.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	A transmit descriptor (TD), an entry in CMsptduse[], contains an invalid usage type (CMSPTDUSE.usetype $\geq$ CMMAXTDUSE).
Possible Error Effect:	The out-of-range value may cause asserts. Also, the TD may not be handled correctly, resulting in lost QLPS network traffic.
Corrective Action Taken:	The audit unlinks the TD from any associated FD, sets all fields to their default values, and links the TD to the TD idle stack.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	CMSPTDUSE.usetype.
<i>Good Data</i> =	CMMAXTDUSE.
<i>Logical Index</i> =	TD's index (index into CMsptduse[]).
<i>Dump</i> =	TD (entry in CMsptduse[]).

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An entry in the stack of idle TDs is invalid. That is, an entry in CMsptdidle.tdindx[] has an index which is less than or equal to CMsptdidle.toppt, but one or more of the following is true: - The value of the entry is out-of-range (CMsptdidle.tdindx[index] $\geq$ CM_NUMTD or < 0). - The associated TD is not linked to the list (CMsptduse[CMsptdidle.tdindx[index]].qindex $\neq$ index).
Possible Error Effect:	The idle stack of TDs is corrupted. This may cause asserts and failures to allocate TDs when requested. Allocation failures may result in lost or delayed QLPS network traffic.
Corrective Action Taken:	If the index into CMsptdidle.tdindx[] is CMsptdidle.toppt, the audit clears the entry (CMsptdidle.tdindx[CMsptdidle.toppt] = CMNIL_NDX) and decrements the count of idle TDs (decrements CMsptdidle.toppt). Otherwise, the audit copies the value of CMsptdidle.tdindx[CMsptdidle.toppt] into CMsptdidle.tdindx[index], sets the associated TD's <i>qindx</i> value to "index" and decrements the count of idle TDs (decrements CMsptdidle.toppt).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Value in CMsptdidle.tdindx[index].
<i>Good Data</i> =	CM_NUMTD.
<i>Logical Index</i> =	Index into CMsptdidle.tdindx[].
<i>Dump</i> =	NA.

Table 3 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	A TD is idle (CMSPTDUSE.usetype = CMTDFREES) but one or more of the following is true: - The TD is not correctly linked to the idle stack (CMSPTDUSE.qindx $>$ CMsptdidle.toppt or CMSPTDUSE.qindx ≤ 0 or the TD's index $\neq$ CMsptdidle.tdindx[CMSPTDUSE.qindx]). - The TD does not contain the expected default values for all fields: - CMSPTDUSE.fd_owner $\neq$ CMNIL_NDX - CMSPTDUSE.channel $\neq$ CMSPCHNULL
Possible Error Effect:	The TD may be lost to the system. If enough TDs were lost, no frames could be transmitted.
Corrective Action Taken:	The audit unlinks the TD from any associated FD, sets all fields to their default values, and links the TD to the TD idle stack.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	CMSPTDUSE.qindx.
<i>Bad Data</i> =	CMSPTDUSE.fd_owner.
<i>Good Data</i> =	CMSPTDUSE.channel.
<i>Logical Index</i> =	TD index (index into CMSptduse[]).
<i>Dump</i> =	The TD (entry in CMSptduse[]).

Table 4 Error Code: IS_NOTUSED

Error Code:	IS_NOTUSED
Error Description:	An unused entry in the TD idle stack (that is, an entry whose index is greater than CMSptdidle.toppt) is not set to the default value (CMSptdidle.tdindx[index] != CMNIL_NDX).
Possible Error Effect:	The head of the TD idle stack (CMSptdidle.toppt) may be corrupted or the entry in the stack itself is corrupted. In either case, TD allocation may fail, causing QLPS network traffic to be lost or delayed.
Corrective Action Taken:	The audit sets the entry to the default value (CMSptdidle.tdindx[index] = CMNIL_NDX). If an idle TD is lost because of this, it is recovered by other checks in this audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Value in CMSptdidle.tdindx[index].
<i>Good Data</i> =	CMNIL_NDX.
<i>Logical Index</i> =	Index into CMSptdidle.tdindx[].
<i>Dump</i> =	NA.

Table 5 Error Code: OWNER

Error Code:	OWNER
Error Description:	The TD idle stack head is invalid. That is, one or more of the following is true: • CMSptdidle.toppt > CM_NUMTDS • CMSptdidle.tdindx[CMSptdidle.toppt] = CMNIL_NDX • CMSptdidle.tdindx[CMSptdidle.toppt+1] != CMNIL_NDX
Possible Error Effect:	TDs may not be allocatable. This may cause the loss or delay of QLPS traffic.
Corrective Action Taken:	The audit rebuilds the TD idle stack by clearing it (CMSptdidle.toppt = 0) and then idling all TDs in the idle state (CMSPTDUSE.usetype = CMTDFREES).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMSptdidle.tdindx[CMSptdidle.toppt].
<i>Bad Data</i> =	CMSptdidle.tdindx[CMSptdidle.toppt+1].
<i>Good Data</i> =	CM_NUMTDS.
<i>Logical Index</i> =	CMSptdidle.toppt.
<i>Dump</i> =	NA.

Table 6 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The specific nature of this error depends on the state of the TD that is given in the error address field of the error report. If CMSPTDUSE.usetype is CMSPTDEOLCLNK or CMTDEOLQPIPE, a TD is in an "end of list" state, but is either linked to an FD (CMSPTDUSE.fd_owner != CMNIL_NDX), or is not linked to any channel data block (CMSPTDUSE.channel ≥ CM_SPYMAX or CMspcdb[CMSPTDUSE.channel].xmt_eoltd != the TD's index). If CMSPTDUSE.usetype is any other value, a TD either has an out-of-range

	FD index (CMSPTDUSE.fd_owner $\geq$ CMNUM_BUF), or is not linked to the FD (CMfduse[CMSPTDUSE.fd_idx].td_idx != the TD's index).
Possible Error Effect:	Any frame associated with this TD is dropped. The TD itself may become unavailable for use. If enough TDs became unavailable, the QLPS network would become degraded.
Corrective Action Taken:	The audit unlinks the TD from any associated FD, sets all fields to their default values, and links the TD to the TD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	CMSPTDUSE.usetype.
<i>Bad Data</i> =	CMSPTDUSE.fd_owner.
<i>Good Data</i> =	CMSPTDUSE.channel.
<i>Logical Index</i> =	TD's index (index into CMsptduse[]).
<i>Dump</i> =	TD (entry in CMsptduse[]).

Table 7 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out-of-range read, or invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit: • Clears (set = CMNIL_NDX) the entry (if any) in CMsptdidle.tdindx[] which was being audited when the stimulus occurred. • Recovers the TD, if any, which was being audited when the stimulus occurred. Specifically, the audit unlinks the TD from any associated FD, sets all fields to their default values, and links the TD to the TD idle stack.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	SMmhrunning (the current base-level loop job).
<i>Bad Data</i> =	Current entry in CMsptdidle.tdindx[].
<i>Good Data</i> =	Current index into CMsptdidle.tdindx[].
<i>Logical Index</i> =	Current TD index (index into CMsptduse[]).
<i>Dump</i> =	TD (entry in CMsptduse[]).

TDB

Software Release: 5E10 and later

1. TASK DATA BLOCK (TDB) AUDIT

Peripheral control uses task data blocks (TDBs) to control tasks coming into the integrated services line unit (ISLU) and remote ISLU (RISLU). Thirty-two task data blocks are allocated per ISLU or RISLU. The TDB audit ensures the integrity of the data structure associated with the task data blocks.

The TDB audit is part of the peripheral control audits subgroup. It runs in the OSDS-M environment in the segmented modes. Refer to the **AUDIT CHECKS** section for more information about the purpose and usage of the TDB audit.

2. DATA STRUCTURES

The TDB audit checks the task ownership table (PCtdbtbl[][]) and the task usage array (PCtdbusage[]). The task usage array contains the busy/idle status of each of the task data blocks in an ISLU or RISLU. Other data structures the audit accesses include the channel control block common area relation (RLCCBCOM), the channel control block foreground region array (PCccbfg[]), and the ring generator task data block (PCrgtdb[][]). These data structures are shown in Figure 1.

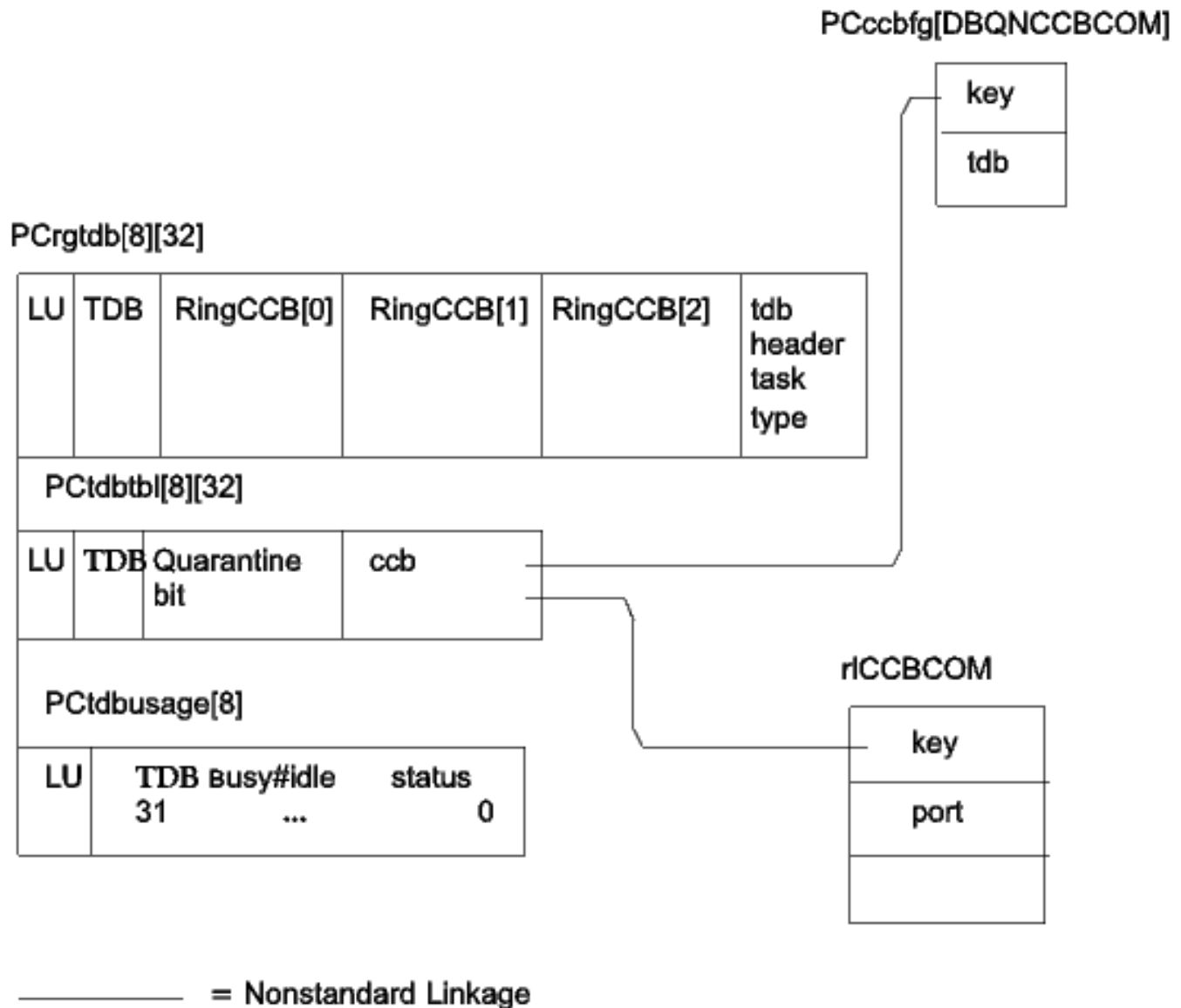


Figure 1 (TDB-1) Data Structures

3. AUDIT CHECKS

The audit checks the task ownership table and the task usage array as follows.

- The TDB audit first checks the value of the quarantine bit in the task ownership table, PCtdbtbl[][]. If the quarantine bit has been set by an audit (AUAUSET), system integrity (AUSISET), or a client program (AUDEFCHK), recovery action is initiated.
- The audit reads the ccbnum field of the task ownership table. If the field is idle, the corresponding bit in the task usage array should indicate that the task data block is idle. If the field is busy, or PCrgtdb[][], the TDB number is verified.
- If the ccbnum field is busy, the TDB number is verified as follows. The audit accesses the corresponding rICCBCom tuple using the ccbnum field in the task ownership table. It checks the owner of the rICCBCom tuple to ensure that the tuple is linked to a process or an unacknowledged task head cell in rPCBLA. If the

riCCBCOM tuple is associated with a process, the audit ensures that the process is alive.

- (d) The audit makes a point-to-point-back check between the PCtdbtbl[][] and the peripheral control channel control block foreground region to ensure that task data blocks marked busy are busy.
- (e) If the ccbnum field is set to PC RG CCB, the corresponding PCrgtdb[][] array entry is accessed using the line unit (LU) and tdb number (TDB) from PCtdbtbl[][]. If the tasktype field in the PCrgtdb[][] array does not indicate that ringing is in progress, recovery action is taken.

4. ERROR HANDLING

If the task usage array is in error, its busy/idle bits are updated according to the state of the task data block. If the PCtdbtbl[][] array is in error, its fields are put to an idle state.

The CCBCOM and RGTDB audit is scheduled as a related audit.

The audit produces the following error code(s).

Table 1 Error Code: BUSYIDLE

Error Code:	BUSYIDLE
Error Description:	The PCtdbtbl[][] indicates that this task data block is busy but the PCtdbusage[] array indicates that the task data block is idle.
Possible Error Effect:	Another process may erroneously be assigned a busy task data block which may cause task failures.
Corrective Action Taken:	Recover the PCtdbusage[] arrays to busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data =</i>	Idle state (0).
<i>Good Data =</i>	Busy state (1).
<i>Logical Index =</i>	(lu << 8) tdb.
<i>Dump =</i>	PCtdbtbl[][] entry.

Table 2 Error Code: CBCOM_LINK

Error Code:	CBCOM_LINK
Error Description:	The point-to-point-back check has failed from the PCtdbtbl[][] and the pcCCBFG region.
Possible Error Effect:	This error will cause application programs to fire asserts.
Corrective Action Taken:	Recover the PCtdbtbl[][] and PCtdbusage[] arrays to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data =</i>	ccbnum field in the PCtdbtbl[][].
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	(lu << 8) tdb.
<i>Dump 1 =</i>	riCCBCOM tuple.
<i>Dump 2 =</i>	pcCCBFG region.

Table 3 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The PCtdbtbl[][] indicates that this task data block is idle but the PCtdbusage[] array indicates that the task data block is busy.
Possible Error Effect:	Loss of task data blocks to the system.
Corrective Action Taken:	Recover the PCtdbusage[] arrays to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data =</i>	Busy state (1).
<i>Good Data =</i>	Idle state (0).

<i>Logical Index</i> =	(lu << 8) tdb.
<i>Dump</i> =	PCtdbtbl[][].

Table 4 Error Code: IDL_STATE

Error Code:	IDL_STATE
Error Description:	The corresponding PCrgtdb ringccb is null.
Possible Error Effect:	The ringing data has been garbaged.
Corrective Action Taken:	This TDB is purged.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the TDB tuple.
<i>Bad Data</i> =	ccbnum of PCtdbtbl array.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	TDB tuple.

Table 5 Error Code: QUAR

Error Code:	QUAR
Error Description:	The TDB audit has found the PCtdbtbl[][] entry to be quarantined.
Possible Error Effect:	Loss of task data block.
Corrective Action Taken:	Recover the PCtdbtbl[][] and PCtdbusage[] arrays to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data</i> =	quar field in PCtdbtbl[][].
<i>Good Data</i> =	Idle value (AUNOTSET).
<i>Logical Index</i> =	(lu << 8) tdb.
<i>Dump</i> =	PCtdbtbl[][].

Table 6 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	The TDB audit has detected that the quarantine bit in the PCtdbtbl[][] has been set to an invalid state.
Possible Error Effect:	Loss of task data blocks.
Corrective Action Taken:	Recover the PCtdbtbl[][] and PCtdbusage[] arrays to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	(lu << 8) tdb.
<i>Dump</i> =	PCtdbtbl[][].

Table 7 Error Code: STUCK

Error Code:	STUCK
Error Description:	The TDB audit has detected that a task data block has been owned by a process for longer than 5 minutes.
Possible Error Effect:	Loss of task data blocks which may affect call processing.
Corrective Action Taken:	Recover the PCtdbtbl[][] and PCtdbusage[] arrays to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	(lu << 8) tdb.
<i>Dump</i> =	PCtdbtbl[][] entry.

Table 8 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audits were invoked after taking a system reset.
Possible Error Effect:	Audits may have caused the interrupt so action must be taken not to roll in the

	interrupts.
Corrective Action Taken:	Recover the PCtdbtbl[] and PCtdbusage[] arrays to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the PCtdbtbl[lu][tdb] array.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	(lu << 8) tdb.
<i>Dump</i> =	PCtdbtbl[][] entry.

TDRWL

Software Release: 5E12 and later

1. TIME DEPENDANT ROUTING WORK LIST (TDRWL) AUDIT

Switch-based time dependent routing (TDR) allows time-of-day/day-of-week routing pattern changes for incoming calls, and supports up to 16 time-of-day changes, allowing a range of start/end times that can be provisioned. The RLTDW_WL relation serves as the work list for time originated recent changes (TORCs) to route indexes. The RLTDW_WL relation is audited by the TDRWL audit, which runs in all SMs and the CMP.

2. DATA STRUCTURES

For details on the fields contained in the RLTDW_WL relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BDPID

Error Code:	BDPID
Error Description:	An invalid process was found stored in the rITDR_WL.tdrpid field. Either the process was not a TDR terminal process (rITDR_WL.tdrpid.progi != GLTPTDR), or the age of the process was older than 30 minutes, or the process was not a live process.
Possible Error Effect:	Possible asserts during TDR operations. Possible exhaustion of TDR resources.
Corrective Action Taken:	The rITDR_WL tuple is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The age of the process in milliseconds. If this is a negative number, then the process found was not a live process.
<i>Bad Data</i> =	The process number (rITDR_WL.tdrpid.procno).
<i>Good Data</i> =	The program ID for the process.
<i>Logical Index</i> =	The logical key of the rITDR_WL tuple (rITDR_WL.irte_idx).
<i>Dump</i> =	The contents of the rITDR_WL tuple found in error.

TEDB-A**Software Release:** 5E10 - 5E11**1. TERMINAL EQUIPMENT DATA BLOCK (TEDB) AUDIT**

The TEDB relation is a dynamic indexed relation which contains the service and feature subscribed by the user. There are four types of TEDB tuples: circuit-switch, on demand-packet, permanent-packet-D, and default service TEDB. TEDB is initialized from static relation and linked to D-channel port block (DPB) during full initialization. When switch establishes layer 2 with the customer premises equipment (CPE), the terminal endpoint identifier (TEI) is sent from the protocol handler (PH) to the switching module (SM) which creates the TEI linkage area (TEILA) tuple. When end point initialization is performed, the TEILA CPE is linked to the services it subscribes (TEDB).

2. DATA STRUCTURES

The data structure checked by TEDB audit is the terminal equipment data block (RLTEDB). TEDB is initialized from the static relation service profile identification (RLSPID) and equipment status (RLDSLEQUIP) DSL equipment status and linked to DPB during full initialization. Each SPID tuple represents the service profile subscribed by a user; Each TEDB tuple contains the specific feature subscribed such as circuit, on-demand-packet, permanent-packet D-channel and default service. Figure 1 illustrates the data structure and linkage area used by TEDB audit.

It operates in the operating system for distributed switching-switching module (OSDS-SM) environment and runs in the initialization and segmented modes. The TEDB audit invokes DRINDEX driver to step through the tuple and validate the key and linkage. The linkage structure is shown in Figure 2.

3. AUDIT CHECKS

The following semantic checks are also performed by this audit:

- (1) Check TEDB against static relation RLSPID or RLDSLEQUIP, if TEDB is not derived from static relation, delete the TEDB tuple.
- (2) Perform semantic linkage between TEDB and TEILA. When end point initialization is performed, TEILA (CPE) is linked to the services to which it subscribes (TEDB). Normally, one TEDB is linked one TEILA but default service TEDB can be shared by up to 8 TEILA (CPEs).
- (3) Verify the attributes in TEDB such as csv_hunt, party, lport and B-channel data from the static relation logical port to physical port (RLLPT_PT).
- (4) Verify the attributes such as cpetype, brcs_flag in TEDB from the static relation RLIS_TERM (ISDN line terminal feature data).
- (5) Verify that TEDB can be accessed for call processing if no process owns D-channel port or automatic customer station rearrangement (ACSR) is not in progress.
- (6) Verify whether the service subscribed requires fully initialized terminal (FIT) or not from static relation service profile identifier (RLSPID).
- (7) If user subscribes multi-line hunt group (MLHG) feature and CPE is not in service, then call to this user should be blocked.
- (8) For attendant console, verify that TEDB is linked to attendant busy lamp feature (ATTM_BLF) through semantic linkage.
- (9) In addition to owner-member linkage between TEDB and DALB, there are semantic linkages that exist for

these two relations such as feat_dalb, sca_dalb, soft_dalb and disp_dalb. For example, the TEDB audit verifies that the DALB tuple used for selected call appearance (sca_dalb) is correctly linked to TEDB.

- (10) Verify that attributes [logical PH number and high level data link control (HDLC) member number] are consistent with these indicated by the RLDSLEQIP static relation.
- (11) Verify that the current selected call appearance and the soft held call appearance are consistent with each other.
- (12) Verify that the active bit (for attendants to determine if a CPE is idle or active) is consistent with the TEDB current selected call appearance, the busy call appearance status in RLLTMSTAT, and the attendant console status relation (RLATTM_BLF).

4. ERROR HANDLING

If audit detects error and this error can be corrected from the static relation, attributes in TEDB are updated per static relation. Otherwise, the TEDB audit quarantines DPB and forces DSL to be recovered. Generally speaking, the linkage errors result in digital subscriber line (DSL) being recovered.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ALINK

Error Code:	ALINK
Error Description:	A TEDB active field is set up to indicate that this TEDB does not have all call appearances busy while the corresponding ATTM_BLF does not indicate the same.
Possible Error Effect:	Attendants utilizing this TEDB and associated ATTM_BLF may not be able to set up call.
Corrective Action Taken:	The associated ATTM_BLF data will be corrected to agree with the TEDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address - this is the value of the associated ATTM_BLF nhscs field.
<i>Logical Index =</i>	Value of htr field of the associated ATTM_BLF.
<i>Bad Data =</i>	Value of alsca field of the associated ATTM_BLF.
<i>Good Data =</i>	TEDB active field.
<i>Dump 1 =</i>	RLTEDB.
<i>Dump 2 =</i>	RLATTM_BLF.

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	TEDB has no BRCS feature but brcs_flag in TEDB is set to DBYES.
Possible Error Effect:	CPE may not be able to communicate with switch.
Corrective Action Taken:	The brcs_flag in TEDB is reset to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	User should be able to use CSV/CSD MLHG feature but ok_term bit in TEDB is

	set to DBNO which will block MLHG termination.
Possible Error Effect:	The available member will be blocked for termination.
Corrective Action Taken:	ok_term bit in TEDB is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	ok_term in TEDB.
<i>Good Data</i> =	DBYES(1).
<i>Dump</i> =	RLTEDB.

Table 4 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	User should be blocked for CSV/CSD MLHG termination because CPE is not bound to the subscribed feature but ok_term bit in TEDB is set to DBYES.
Possible Error Effect:	CSV/CSD MLHG may terminate a call to OOS CPE and possibly cause rehunt of another member.
Corrective Action Taken:	ok_term bit in TEDB is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	ok_term in TEDB.
<i>Good Data</i> =	DBNO(0).
<i>Dump</i> =	RLTEDB.

Table 5 Error Code: BRCSFLAG

Error Code:	BRCSFLAG
Error Description:	TEDB has BRCS feature but brcs_flag in TEDB is set to DBNO.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	brcs_flag is set to DBYES in TEDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	1 (indicate brcs_flag should be set to DBYES).
<i>Dump</i> =	RLTEDB.

Table 6 Error Code: BU_STAT

Error Code:	BU_STAT
Error Description:	The TEDB active bit is set to indicate active when there is no selected call appearance DALB and all the call appearances available are not busy.
Possible Error Effect:	Wrong information may be indicated to attendant console.
Corrective Action Taken:	TEDB active field is reset to indicate not busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not an address; this is the value of internal variable used to indicate if all call appearances are busy or at least one is idle. 0 = idle. 1 = busy.
<i>Bad Data</i> =	TEDB active field (0 = idle, 1 = busy).
<i>Good Data</i> =	Value returned from macro reading busy/idle status of last call appearance status accessed by audit. The status is read from the RLLTMSTAT relation.
<i>Logical Index</i> =	TEDB key.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	LTMSTAT.

Table 7 Error Code: CALLFLAG_BAD

Error Code:	CALLFLAG_BAD
Error Description:	blkcall field in TEDB should be set to DBNO but it is set to DBYES. It will block the feature subscribed by the user.
Possible Error Effect:	CPE cannot utilize the subscribed feature.
Corrective Action Taken:	blkcall field is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	blkcall field in TEDB.
<i>Good Data =</i>	DBNO.
<i>Dump =</i>	RLTEDB.

Table 8 Error Code: CPETYPE

Error Code:	CPETYPE
Error Description:	The type of CPE equipment the TEDB thinks it owns is not consistent with the RLISTERM relation.
Possible Error Effect:	Switch may not correctly communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CPE equipment value in TEDB.
<i>Good Data =</i>	CPE equipment value in ISTERM.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	RLTEDB.

Table 9 Error Code: DFLG

Error Code:	DFLG
Error Description:	The default group in the TEDB is not set to correct group.
Possible Error Effect:	CPE cannot utilize the subscribed default feature or may get the wrong service.
Corrective Action Taken:	The default group in TEDB is reset to the right group per static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	Default group in TEDB.
<i>Good Data =</i>	The right default group.
<i>Dump =</i>	RLTEDB.

Table 10 Error Code: DSLEQUIP

Error Code:	DSLEQUIP
Error Description:	DSLEQUIP tuple does not exist, but TEDB is correctly linked to DPB.
Possible Error Effect:	CPE cannot communicate with switch and application code will assert.
Corrective Action Taken:	NA.
Manual Action Required:	The missing DSLEQUIP tuple needs to be inserted.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Logical Index =</i>	TEDB key.
<i>Good Data =</i>	NA.
<i>Bad Data =</i>	The key of DSLEQUIP.
<i>Dump =</i>	RLTEDB.

Table 11 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The executive ringer cutoff (ERCO) special features flag (rTEDB.ercos_on) is not marked correctly. If ERCO is assigned to a terminal and is active as indicated by:

	rIFC_LINE.erco_asgn = DBYES and rIFC_LINE.erco_act = DBYES where rIFC_LINE.port.member = (rITEDB.dpbmem.st_ownk - DBSFDPB), rIFC_LINE.port.module = GI_pcrd, and rIFC_LINE.party = rITEDB.party The erco_on field in RLTEDB should be marked DBYES. Otherwise, it should be marked DBNO.
Possible Error Effect:	Power ringing will be allowed to occur when it is not supposed to, or it will be denied when it should be allowed.
Corrective Action Taken:	The erco_on attribute in the rITEDB tuple will be corrected to properly reflect whether or not the ERCO feature is assigned, as indicated in the rIFC_LINE static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The rITEDB tuple address.
<i>Bad Data</i> =	The incorrect state the erco_on attribute was found in, as shown in the following table.
<i>Good Data</i> =	The correct value of the erco_on attribute, as shown in the following table.
<i>Logical Index</i> =	The rITEDB tuple's key (tedb).
<i>Dump 1</i> =	The contents of the rITEDB dynamic tuple.
<i>Dump 2</i> =	The contents of the rIFC_LINE static tuple.

erco_asgn	erco_act	erco_on (valid)	erco_on (invalid)
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1

Table 12 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	There is an inconsistency between the TEDB port information and the static information regarding the port in the RLDSLEQUIP relation.
Possible Error Effect:	Switch may not correctly communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	dsl attribute of the RLDSLEQUIP relation, arrived at through lport attribute of TEDB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEDB.

Table 13 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rITEDB dynamic tuple without a corresponding static tuple to represent the valid service-of-profile(SOP) subscribed by the user. The TEDB key is represented by the diagram below: <pre> ----- xxx xxxxxxxxxxxxxx ----- 15 12 0 Bit 0--12 (lport) Bit 13--15 (lpt_type) </pre> The TEDB tuple does not have one of corresponding static tuples where:

	(1) rLTEDB.tedb = rLTPT_PT.lport and rLTPT_PT.lpt_type (2) rLTPT_PT.lport = rLTPT_SPID.lport rLTPT_PT.lpt_type = rLTPT_SPID.lpt_type (3) rLTPT_SPID.epdn = rLTSPID.epdn rLTPT_SPID.natype = rLTSPID.natype rLTPT_SPID.npid = rLTSPID.npid (4) rLTSPID.ckt_lport = lport in rLTEDB key (with lpt_type = LPT_ISDN) rLTSPID.odp_lport = lport in rLTEDB key (with lpt_type = LPT_PKT) rLTSPID.ppd_lport = lport in rLTEDB key (with lpt_type = LPT_PKT)
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEDB.

Table 14 Error Code: FIT

Error Code:	FIT
Error Description:	The subscribed feature requires the fully initialized terminal (FIT) but TEDB fip_req field is not set correctly.
Possible Error Effect:	CPE cannot utilize the subscribed feature.
Corrective Action Taken:	fip_req field in TEDB is set correctly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	fip_req field in TEDB.
<i>Good Data</i> =	Valid values are:
	1 = FIT is required for this feature.
	0 = FIT is not required for this feature.
<i>Dump</i> =	RLTEDB.

Table 15 Error Code: FLAG_CALL_BAD

Error Code:	FLAG_CALL_BAD
Error Description:	If user does not subscribe CSV/CSD MLHG or user subscribes ISAT MLHG, ok_term bit in TEDB should be set to DBNO. But it is set to DBYES.
Possible Error Effect:	Call processing may try to terminate to this non CSV/CSD member.
Corrective Action Taken:	ok_term bit in TEDB is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	ok_term in TEDB.

<i>Good Data</i> =	DBNO(0).
<i>Dump</i> =	RLTEDB.

Table 16 Error Code: GPORT

Error Code:	GPORT
Error Description:	The user is active on a point-to-point BRI (rIDSLEQUIP.briclass = BRI_PP) or restrictive multipoint BRI (rIDSLEQUIP.briclass = BRI_MP), but the ok_term bit in the rITEDB tuple is incorrectly marked DBNO. For non-default Service rITEDB tuples associated with restrictive multipoint, or point-to-point circuit switched calls (rITEDB.sc = DMSC_CKT and rITEDB.dflt_grp = 0 and rITEDB.lport = rIDSLEQUIP.bind_port), the ok_term bit should be marked DBYES if there is an active user (rITEDB.csv_hunt = DBYES or rITEDB.csd_hunt = DBYES).
Possible Error Effect:	The active user will be blocked for call terminations.
Corrective Action Taken:	The ok_term bit in the rITEDB tuple is marked DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rITEDB tuple address.
<i>Logical Index</i> =	rITEDB tuple's key.
<i>Bad Data</i> =	The ok_term bit as found in RLTEDB -- DBNO (0).
<i>Good Data</i> =	The value the ok_term bit was corrected to -- DBYES(1).
<i>Dump</i> =	The contents of the rITEDB tuple.

Table 17 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The TEDB sca_dalb field is null while the soft_dalb is not null. This indicates that a party has been put on hold while there is no other call associated with the TEDB.
Possible Error Effect:	Party may be left on hold indefinitely with no indication to the console.
Corrective Action Taken:	TEDB is zapped and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rITEDB tuple.
<i>Bad Data</i> =	sca_dalb field of rITEDB tuple.
<i>Good Data</i> =	soft_dalb field of rITEDB tuple.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	rITEDB tuple.

Table 18 Error Code: LPHBAD

Error Code:	LPHBAD
Error Description:	The logical PH value in the TEDB is not consistent with the value in the DSLEQUIP static data.
Possible Error Effect:	Switch may not communicate with customer premise equipment.
Corrective Action Taken:	The attribute in TEDB is updated correctly.
Manual Action Required:	None.
<i>Logical Index</i> =	TEDB key.
<i>Good Data</i> =	NA
<i>Bad Data</i> =	lph attribute found in TEDB.
<i>Good Data</i> =	dsl_group attribute found in the RLDSLEQUIP relation.
<i>Dump</i> =	RLTEB.

Table 19 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	An inconsistency was found in the port name in the TEDB tuple. An attempt to convert the port name from logical port to physical port was unsuccessful.
Possible Error Effect:	Calls may be set-up incorrectly, and attempts to make calls on call appearances related to the CPE associated with this TEDB may not be successful.
Corrective Action Taken:	TEDB is zapped out and placed on the idle list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Bad Data</i> =	TEDB lport attribute.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEDB.

Table 20 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	A semantic data inconsistency was found between TEDB and RLLPT_PT.
Possible Error Effect:	CPE may not be able to utilize the features subscribed such as B-channels and MLHG.
Corrective Action Taken:	The semantic data in the TEDB is corrected to be consistent with RLLPT_PT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	TEDB lport attribute.
<i>Dump</i> =	RLTEB.

Table 21 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	The attributes in TEDB are not set correctly per static relation RLLPT_PT.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	The attributes are set correctly per static relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Dump</i> =	RLTEDB.

Table 22 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The PH member value in the TEDB is not consistent with the value in the DSLEQUIP static data.
Possible Error Effect:	Switch may not communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	mem attribute found in TEDB.
<i>Good Data</i> =	Member attribute found in the RLDSLEQUIP relation.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEB.

Table 23 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The TEDB audit cannot access the RLLTMSTAT dynamic relation with the ltm_idx field from a valid rIPOINT_GROUP static data tuple.
Possible Error Effect:	The TEDB audit cannot complete all of its checks.
Corrective Action Taken:	Schedule the related LTMSTAT audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rITEDB tuple.
<i>Bad Data</i> =	ltm_idx field of tuple returned from database primitive.
<i>Good Data</i> =	ltm_idx field from associated static rIPOINT_GROUP tuple.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	rITEDB tuple.

Table 24 Error Code: MSG_PTR

Error Code:	MSG_PTR
Error Description:	TEDB should not forward every message to the designated ACSR process but fwd_msgs flag is set to.
Possible Error Effect:	CPE will forward the message to wrong process and lose communication.
Corrective Action Taken:	fwd_msgs field is reset to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	fwd_msgs in TEDB.
<i>Good Data =</i>	DBNO(0).
<i>Dump =</i>	RLTEDB.

Table 25 Error Code: NOACCBLK

Error Code:	NOACCBLK
Error Description:	The blkcall attribute in this rLTEDB tuple was found incorrectly marked DBNO. If the user is being intercepted (rIFC_LINE.intercept = DBYES) or this is a non-default service RLTEB (no default service group is associated with it such that rLTEDB.dflt_grp = 0), and the user service order profile (USOP) indicates that an automatic customer station rearrangement (ACSR) request is currently in progress (AUspid_buf.blk_call = DBYES), then the blkcall bit in RLTEB should be DBYES, indicating that calls to this port should be blocked.
Possible Error Effect:	The user will be allowed access to unsubscribed services.
Corrective Action Taken:	The blkcall field in RLTEB is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rLTEDB tuple address.
<i>Bad Data =</i>	The blkcall field found in RLTEB (DBNO).
<i>Good Data =</i>	The correct value of the blkcall attribute in RLTEB (DBYES).
<i>Logical Index =</i>	The rLTEDB key (tedb).
<i>Dump =</i>	The contents of the rLTEDB tuple.

Table 26 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The TEDB active bit is set to indicate not active when there is a selected call appearance DALB or all the call appearances available are busy.
Possible Error Effect:	Wrong information may be indicated to attendant console.
Corrective Action Taken:	TEDB active field is reset to indicate busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the value of internal variable used to indicate if all call appearances are busy or at least one is idle. 0 = idle. 1 = busy.
<i>Bad Data =</i>	TEDB active field (0 = idle, 1 = busy).
<i>Good Data =</i>	Value returned from macro reading busy/idle status of last call appearance status accessed by audit. The status is read from the rLTMSTAT relation.
<i>Logical Index =</i>	TEDB key.
<i>Dump 1 =</i>	RLTEB.
<i>Dump 2 =</i>	LTMSTAT.

Table 27 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The user is inactive on a point-to-point BRI or inactive on a restrictive multipoint BRI (rIDSLEQUIP.briclass = BRI_MP or rIDSLEQUIP.briclass = BRI_PP, and

	rITEDB.lport != rIDSEQUIP.bind_port) but the ok_term bit in the rITEDB tuple is incorrectly marked DBYES. For non-default service rITEDB tuples associated with restrictive multipoint, or point-to-point circuit switched calls (rITEDB.dflt_grp = 0 and rITEDB.sc = DMSC_CKT) the ok_term bit should be marked DBNO if there is no active user.
Possible Error Effect:	An inactive user will be allowed call terminations.
Corrective Action Taken:	The ok_term bit in RLTEDB is set to DBNO.
Manual Action Required:	None
Dump Description:	
<i>Error Address =</i>	The rITEDB tuple's address.
<i>Logical Index =</i>	The rITEDB tuple's key (tedb).
<i>Bad Data =</i>	The ok_term attribute in rITEDB (DBYES).
<i>Good Data =</i>	The value the ok_term attribute was corrected to (DBNO).
<i>Dump =</i>	The contents of the rITEDB tuple.

Table 28 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The attributes in TEDB are not set correctly per static relation RLIS_TERM.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	The attributes are set correctly per static relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 29 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	RLSPID or RLLPT_PT tuple is missing in the ODD.
Possible Error Effect:	CPE cannot utilize the subscribed features.
Corrective Action Taken:	None.
Manual Action Required:	Need to insert the missing SPID or LPT_PT tuple. (If missing SPID tuple, get lport from GOOD DATA and access relation rLPT_SPID which contains the key of SPID tuple) (The key of RLLPT_PT can be taken out of lport in GOOD DATA).
Dump Description:	
<i>Error Address =</i>	NA.
<i>Logical Index =</i>	TEDB Key.
<i>Bad Data =</i>	The relation ID or RLSPID or RLLPT_PT.
<i>Good Data =</i>	The logical port name.
<i>Dump =</i>	RLTEDB.

Table 30 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The user is active on a pure multipoint BRI (rIDSLEQUIP.bri_class = BRI_PP) but the ok_term bit in RLTEDB is set to DBNO. For pure multipoint service, PPD (permanent packet D) rITEDB tuples (rITEDB.sc = DMSC_PPD), the ok_term attribute should be DBYES when a user is active (rITEDB.csv_hunt = DBYES or rITEDB.csd_hunt = DBYES).
Possible Error Effect:	The active user will be blocked for call terminations.
Corrective Action Taken:	The ok_term bit in rITEDB is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rITEDB tuple's address.
<i>Bad Data =</i>	The ok_term attribute found in RLTEDB (DBNO).
<i>Good Data =</i>	The value that the ok_term attribute was corrected to (DBYES).
<i>Logical Index =</i>	The rITEDB tuple's key(tedb).
<i>Dump =</i>	The contents of the rITEDB tuple.

Table 31 Error Code: PORT

Error Code:	PORT
Error Description:	The TEDB has encountered an ODD inconsistency when trying to access the static relation RLPORT_GROUP. The inconsistency is due to using the global variable GI_pcrd (module number) and port field from the ODD relation RLLPT_PT to access the RLPORT_GROUP relation.
Possible Error Effect:	Conversions of logical to physical port may be incorrect.
Corrective Action Taken:	TEDB tuple being checked is zapped and placed on idle list.
Manual Action Required:	Check the rIPOINT_GROUP tuple with port from RLLPT_PT relation (listed in Good Data) and the module number for this SM.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to rITEDB tuple.
<i>Bad Data =</i>	rIPOINT_GROUP.port.member field, part of key to rIPOINT_GROUP.
<i>Good Data =</i>	Associated rIPOINT_PT.port field.
<i>Logical Index =</i>	TEDB key.
<i>Dump 1 =</i>	rITEDB tuple.
<i>Dump 2 =</i>	rIPOINT_PT tuple.

Table 32 Error Code: RDSC

Error Code:	RDSC
Error Description:	The service order profile (SOP) capability is not allowed by the system.
Possible Error Effect:	CPE will lose the feature which it subscribes.
Corrective Action Taken:	The TEDB tuple will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	SOP capability in TEDB.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 33 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The attributes in TEDB such as cpetype are not set to default value if no IS_TERM static tuple exists.
Possible Error Effect:	CPE may not be able to communicate with switch.
Corrective Action Taken:	The attributes are correctly set to default value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 34 Error Code: SOPCAP

Error Code:	SOPCAP
Error Description:	The SOP capability in the TEDB is not set to correct feature.
Possible Error Effect:	CPE cannot utilize the subscribed feature.
Corrective Action Taken:	The SOP capability is reset to the right feature per static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	SOP capability in TEDB.
<i>Good Data =</i>	The right SOP (DMCKT_SOP, DMPKT_SOP, DMODP_COP).
<i>Dump =</i>	RLTEDB.

Table 35 Error Code: STABLE_LINK

Error Code:	STABLE_LINK

Error Description:	The semantic linkage between TEDB and ATTM BLF does not point back to each other.
Possible Error Effect:	Attendant console is OOS because of semantic linkage error.
Corrective Action Taken:	DSL will be reinitialized and attendant console will be initialized to be in-service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	tedb KEY in ATTM BLF.
<i>Good Data</i> =	tedb KEY.
<i>Dump 1</i> =	RLTEDB
<i>Dump 2</i> =	RLATTM BLF.

Table 36 Error Code: TEDB

Error Code:	TEDB
Error Description:	The DALB owned by this TEDB does not think it belongs to the TEDB.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DALB linkage field to TEDB.
<i>Good Data</i> =	TEDB linkage field to TEDB.
<i>Logical Index</i> =	PORT number of D-PORT.
<i>Dump</i> =	RLTEDB.

Table 37 Error Code: TEDBDISP

Error Code:	TEDBDISP
Error Description:	A TEDB believes there is a DALB linked to it which contains the display terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	Display information on the customer premise equipment associated with this tuple may not be correct. Communication to the CPE may be prohibited.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	DALB member linkage.
<i>Good Data</i> =	TEDB owner linkage.
<i>Dump</i> =	RLTEDB.

Table 38 Error Code: TEDBFEAT

Error Code:	TEDBFEAT
Error Description:	A TEDB believes there is a DALB linked to it which contains the feature terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	Switch may not correctly set up certain feature calls on the related customer premise equipment.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DALB member linkage.
<i>Good Data</i> =	TEDB owner linkage.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEDB, RLDALB.

Table 39 Error Code: TEDBSCA

Error Code:	TEDBSCA
Error Description:	A TEDB believes there is a DALB linked to it which contains the current selected call appearance terminal process ID. However, the DALB is not linked back to the TEDB.

Possible Error Effect:	The selected call appearance capability on the customer premise equipment associated with this TEDB may not work. Communication to the CPE may be affected.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DALB member linkage.
<i>Good Data =</i>	TEDB owner linkage.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	RLTEDB, RLDALB.

Table 40 Error Code: TEDBSOFT

Error Code:	TEDBSOFT
Error Description:	A TEDB believes there is a DALB linked to it which contains the soft held terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	The soft hold capability on the customer premise equipment associated with this TEDB may not work. Calls on hold may be lost, and communication to the CPE may be affected.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	DALB member linkage.
<i>Good Data =</i>	TEDB owner linkage.
<i>Logical Index =</i>	TEDB Key.
<i>Dump =</i>	RLTEDB RLDALB.

Table 41 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	TEDB is not correctly linked to TEILA through semantic linkage.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	The TEDB tuple will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	tedb stored at TEILA.
<i>Good Data =</i>	tedb key should be stored at TEILA.
<i>Dump 1 =</i>	RLTEDB.
<i>Dump 2 =</i>	RLTEILA.

Table 42 Error Code: TYPE

Error Code:	TYPE
Error Description:	The TEDB has encountered an ODD inconsistency when trying to access the static relation RLIS_LINE. The inconsistency is due to using the global variable GI_pcrd (module number) and ltm_idx field from the associated rIPORT_GROUP ODD tuple to access the RLIS_LINE relation.
Possible Error Effect:	The TEDB audit cannot complete all of its checks.
Corrective Action Taken:	TEDB tuple being checked is zapped and placed on idle list.
Manual Action Required:	Check the rIPORT_GROUP tuple and the rLIS_LINE tuple with information given.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to rITEDB tuple.
<i>Bad Data =</i>	rIPORT_GROUP.ltm_idx field (used to access rLIS_LINE).
<i>Good Data =</i>	Return from database access primitive.
<i>Logical Index =</i>	TEDB key.
<i>Dump 1 =</i>	rITEDB tuple.
<i>Dump 2 =</i>	Associated rIPORT_GROUP tuple.

Table 43 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A TEDB which is linked to the idle list (DBHEADC) has non-null value in its other linkage fields.
Possible Error Effect:	There may be problems in communicating to a CPE if the TEDB is linked to a DPB.
Corrective Action Taken:	Linkage fields in the TEDB tuple are set to null, except for member linkage to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	TEDB owner head linkage field.
<i>Good Data =</i>	TEDB owner tail linkage field.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	RLTEDB.

TEDB-B

Software Release: 5E12 and later

1. TERMINAL EQUIPMENT DATA BLOCK (TEDB) AUDIT

The TEDB relation is a dynamic indexed relation which contains the service and feature subscribed by the user. There are four types of TEDB tuples: circuit-switch, on demand-packet, permanent-packet-D, and default service TEDB. TEDB is initialized from static relation and linked to D-channel port block (DPB) during full initialization. When switch establishes layer 2 with the customer premises equipment (CPE), the terminal endpoint identifier (TEI) is sent from the protocol handler (PH) to the switching module (SM) which creates the TEILA tuple. When end point initialization is performed, the TEILA CPE is linked to the services it subscribes (TEDB).

2. DATA STRUCTURES

The data structure checked by TEDB audit is the terminal equipment data block (RLTEDB). TEDB is initialized from the static relation service profile identification (RLSPID) and equipment status (RLDSLEQUIP) DSL equipment status and linked to DPB during full initialization. Each SPID tuple represents the service profile subscribed by a user; Each TEDB tuple contains the specific feature subscribed such as circuit, on-demand-packet, permanent-packet D-channel and default service. Figure 1 illustrates the data structure and linkage area used by TEDB audit. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment and runs in the initialization and segmented modes. The TEDB audit invokes DRINDEX driver to step through the tuple and validate the key and linkage. The linkage structure is shown in Figure 2 .

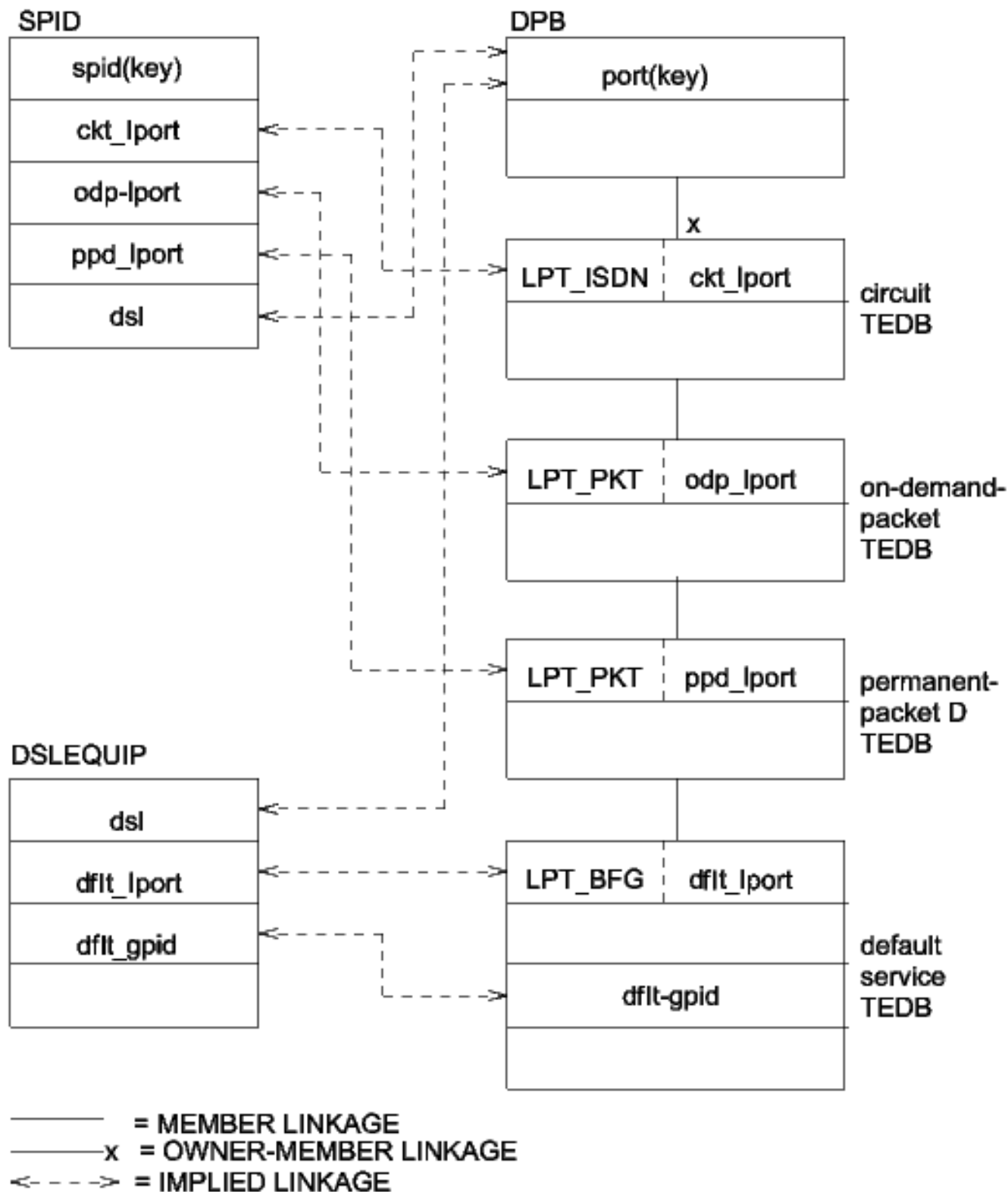


Figure 1 (TEDB-1) TEDB Initialization Diagram

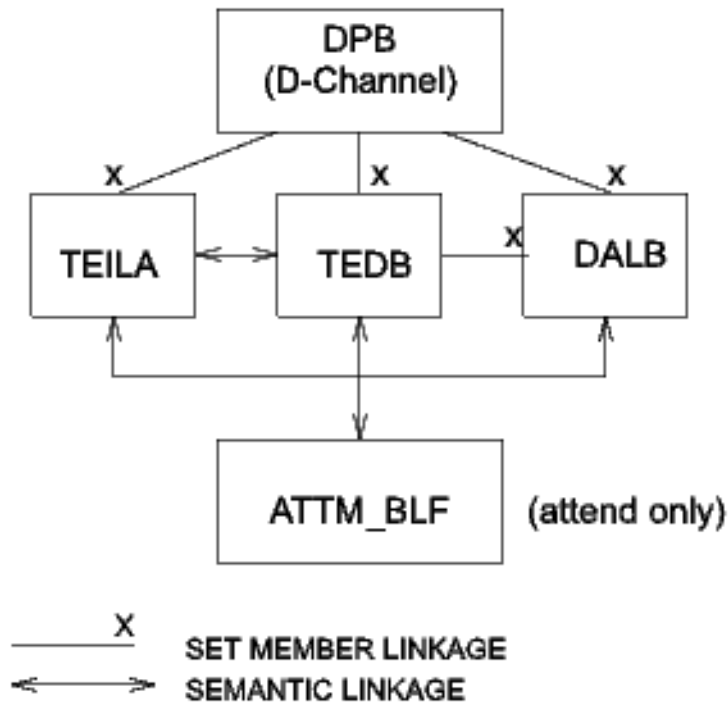


Figure 2 (TEDB-2) The Linkage Structure of TEDB

3. AUDIT CHECKS

The following semantic checks are also performed by this audit:

- Check TEDB against static relation RLSPID or RLDSLEQUIP, if TEDB is not derived from static relation, delete the TEDB tuple.
- Perform semantic linkage between TEDB and TEI linkage area (TEILA). When end point initialization is performed, TEILA (CPE) is linked to the services to which it subscribes (TEDB). Normally, one TEDB is linked one TEILA but default service TEDB can be shared by up to 8 TEILA (CPEs).
- Verify the attributes in TEDB such as csv_hunt, party, lport and B-channel data from the static relation logical port to physical port (RLLPT_PT).
- Verify the attributes such as cptype, brcs_flag in TEDB from the static relation RLIS_TERM (ISDN line terminal feature data).
- Verify that TEDB can be accessed for call processing if no process owns D-channel port or automatic customer station rearrangement (ACSR) is not in progress.
- Verify whether the service subscribed requires fully initialized terminal (FIT) or not from static relation service profile identifier (RLSPID).
- If user subscribes multi-line hunt group (MLHG) feature and CPE is not in service, then call to this user should be blocked.
- For attendant console, verify that TEDB is linked to attendant busy lamp feature (ATTM_BLF) through semantic linkage.
- In addition to owner-member linkage between TEDB and DALB, there are semantic linkages that exist for

these two relations such as feat_dalb, sca_dalb, soft_dalb and disp_dalb. For example, the TEDB audit verifies that the DALB tuple used for selected call appearance (sca_dalb) is correctly linked to TEDB.

- (j) Verify that attributes [logical PH number and high level data link control (HDLC) member number] are consistent with these indicated by DSLEQIP static relation.
- (k) Verify that the current selected call appearance and the soft held call appearance are consistent with each other.
- (l) Verify that the active bit (for attendants to determine if a CPE is idle or active) is consistent with the TEDB current selected call appearance, the busy call appearance status in RLLTMSTAT, and the attendant console status relation (RLATTM_BLF).

4. ERROR HANDLING

If audit detects error and this error can be corrected from the static relation, attributes in TEDB are updated per static relation. Otherwise, the TEDB audit quarantines DPB and forces DSL to be recovered. Generally speaking, the linkage errors result in digital subscriber line (DSL) being recovered.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ALINK

Error Code:	ALINK
Error Description:	A TEDB active field is set up to indicate that this TEDB does not have all call appearances busy while the corresponding ATTM_BLF does not indicate the same.
Possible Error Effect:	Attendants utilizing this TEDB and associated ATTM_BLF may not be able to set up call.
Corrective Action Taken:	The associated ATTM_BLF data will be corrected to agree with the TEDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NOT an address - this is the value of the associated ATTM_BLF nhscs field.
<i>Logical Index =</i>	Value of htr field of the associated ATTM_BLF.
<i>Bad Data =</i>	Value of alsca field of the associated ATTM_BLF.
<i>Good Data =</i>	TEDB active field.
<i>Dump 1 =</i>	RLTEDB.
<i>Dump 2 =</i>	RLATTM_BLF.

Table 2 Error Code: BADFLAG

Error Code:	BADFLAG
Error Description:	TEDB has no BRCS feature but brcs_flag in TEDB is set to DBYES.
Possible Error Effect:	CPE may not be able to communicate with switch.
Corrective Action Taken:	The brcs_flag in TEDB is reset to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 3 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	User should be able to use CSV/CSD MLHG feature but ok_term bit in TEDB is

	set to DBNO which will block MLHG termination.
Possible Error Effect:	The available member will be blocked for termination.
Corrective Action Taken:	ok_term bit in TEDB is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	ok_term in TEDB.
<i>Good Data</i> =	DBYES(1).
<i>Dump</i> =	RLTEDB.

Table 4 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	User should be blocked for CSV/CSD MLHG termination because CPE is not bound to the subscribed feature but ok_term bit in TEDB is set to DBYES.
Possible Error Effect:	CSV/CSD MLHG may terminate a call to OOS CPE and possibly cause rehunt of another member.
Corrective Action Taken:	ok_term bit in TEDB is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	ok_term in TEDB.
<i>Good Data</i> =	DBNO (0).
<i>Dump</i> =	RLTEDB.

Table 5 Error Code: BRCSFLAG

Error Code:	BRCSFLAG
Error Description:	TEDB has BRCS feature but brcs_flag in TEDB is set to DBNO.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	brcs_flag is set to DBYES in TEDB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	1 (indicate brcs_flag should be set to DBYES).
<i>Dump</i> =	RLTEDB.

Table 6 Error Code: BU_STAT

Error Code:	BU_STAT
Error Description:	The TEDB active bit is set to indicate active when there is no selected call appearance DALB and all the call appearances available are not busy.
Possible Error Effect:	Wrong information may be indicated to attendant console.
Corrective Action Taken:	TEDB active field is reset to indicate not busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Not an address; this is the value of internal variable used to indicate if all call appearances are busy or at least one is idle. 0 = idle. 1 = busy.
<i>Bad Data</i> =	TEDB active field (0 = idle, 1 = busy).
<i>Good Data</i> =	Value returned from macro reading busy/idle status of last call appearance status accessed by audit. The status is read from the RLLTMSTAT relation.
<i>Logical Index</i> =	TEDB key.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	LTMSTAT.

Table 7 Error Code: CALLFLAG_BAD

Error Code:	CALLFLAG_BAD
Error Description:	blkcall field in TEDB should be set to DBNO but it is set to DBYES. It will block the feature subscribed by the user.
Possible Error Effect:	CPE cannot utilize the subscribed feature.
Corrective Action Taken:	blkcall field is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	blkcall field in TEDB.
<i>Good Data =</i>	DBNO.
<i>Dump =</i>	RLTEDB.

Table 8 Error Code: CPETYPE

Error Code:	CPETYPE
Error Description:	The type of CPE equipment the TEDB thinks it owns is not consistent with the ISTERM relation.
Possible Error Effect:	Switch may not correctly communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	CPE equipment value in TEDB.
<i>Good Data =</i>	CPE equipment value in ISTERM.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	RLTEDB.

Table 9 Error Code: DFLG

Error Code:	DFLG
Error Description:	The default group in the TEDB is not set to correct group.
Possible Error Effect:	CPE cannot utilize the subscribed default feature or may get the wrong service.
Corrective Action Taken:	The default group in TEDB is reset to the right group per static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	Default group in TEDB.
<i>Good Data =</i>	The right default group.
<i>Dump =</i>	RLTEDB.

Table 10 Error Code: DSLEQUIP

Error Code:	DSLEQUIP
Error Description:	DSLEQUIP tuple does not exist, but TEDB is correctly linked to DPB.
Possible Error Effect:	CPE cannot communicate with switch and application code will assert.
Corrective Action Taken:	NA.
Manual Action Required:	The missing DSLEQUIP tuple needs to be inserted.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Logical Index =</i>	TEDB key.
<i>Good Data =</i>	NA.
<i>Bad Data =</i>	The key of DSLEQUIP.
<i>Dump =</i>	RLTEDB.

Table 11 Error Code: DYN_STAT

Error Code:	DYN_STAT
Error Description:	The executive ringer cutoff (ERCO) special features flag (rTEDB.ercu_on) is not marked correctly. If ERCO is assigned to a terminal and is active as indicated by:

	rIFC_LINE.erco_asgn = DBYES and rIFC_LINE.erco_act = DBYES where rIFC_LINE.port.member = (rITEDB.dpbmem.st_ownk - DBSFDPB), rIFC_LINE.port.module = GI_pcrd, and rIFC_LINE.party = rITEDB.party The erco_on field in RLTEDB should be marked DBYES. Otherwise, it should be marked DBNO.
Possible Error Effect:	Power ringing will be allowed to occur when it is not supposed to, or it will be denied when it should be allowed.
Corrective Action Taken:	The erco_on attribute in the rITEDB tuple will be corrected to properly reflect whether or not the ERCO feature is assigned, as indicated in the rIFC_LINE static tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rITEDB tuple address.
<i>Bad Data =</i>	The incorrect state the erco_on attribute was found in, as shown in the following table.
<i>Good Data =</i>	The correct value of the erco_on attribute, as shown in the following table.
<i>Logical Index =</i>	The rITEDB tuple's key (tedb).
<i>Dump 1 =</i>	The contents of the rITEDB dynamic tuple.
<i>Dump 2 =</i>	The contents of the rIFC_LINE static tuple.

erco_asgn	erco_act	erco_on (valid)	erco_on (invalid)
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1

Table 12 Error Code: EQUIP_DSL

Error Code:	EQUIP_DSL
Error Description:	There is an inconsistency between the TEDB port information and the static information regarding the port in the DSLEQUIP relation.
Possible Error Effect:	Switch may not correctly communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	dsl attribute of DSLEQUIP relation, arrived at through lport attribute of TEDB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	RLTEDB.

Table 13 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The audit has detected an rITEDB dynamic tuple without a corresponding static tuple to represent the valid service-of-profile (SOP) subscribed by the user. The TEDB key is represented by the diagram below: <pre> ----- xxx xxxxxxxxxxxxxx ----- 15 12 0 Bit 0--12 (lport) Bit 13--15 (lpt_type) </pre> The TEDB tuple does not have one of corresponding static tuples where:

	(1) rLTEDB.tedb = rLTPT_PT.lport and rLTPT_PT.lpt_type (2) rLTPT_PT.lport = rLTPT_SPID.lport rLTPT_PT.lpt_type = rLTPT_SPID.lpt_type (3) rLTPT_SPID.epdn = rLTSPID.epdn rLTPT_SPID.natype = rLTSPID.natype rLTPT_SPID.npid = rLTSPID.npid (4) rLTSPID.ckt_lport = lport in rLTEDB key (with lpt_type = LPT_ISDN) rLTSPID.odp_lport = lport in rLTEDB key (with lpt_type = LPT_PKT) rLTSPID.ppd_lport = lport in rLTEDB key (with lpt_type = LPT_PKT)
Possible Error Effect:	If call processing or terminal maintenance attempts to access the unassigned resources represented by this tuple, operations may be disrupted.
Corrective Action Taken:	Tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEDB.

Table 14 Error Code: FIT

Error Code:	FIT
Error Description:	The subscribed feature requires the fully initialized terminal (FIT) but TEDB fip_req field is not set correctly.
Possible Error Effect:	CPE cannot utilize the subscribed feature.
Corrective Action Taken:	fip_req field in TEDB is set correctly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	fip_req field in TEDB.
<i>Good Data</i> =	Valid values are: 1 = FIT is required for this feature. 0 = FIT is not required for this feature.
<i>Dump</i> =	RLTEDB.

Table 15 Error Code: FLAG_CALL_BAD

Error Code:	FLAG_CALL_BAD
Error Description:	If user does not subscribe CSV/CSD MLHG or user subscribes ISAT MLHG, ok_term bit in TEDB should be set to DBNO. But it is set to DBYES.
Possible Error Effect:	Call processing may try to terminate to this non CSV/CSD member.
Corrective Action Taken:	ok_term bit in TEDB is set to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	ok_term in TEDB.
<i>Good Data</i> =	DBNO(0).
<i>Dump</i> =	RLTEDB.

Table 16 Error Code: GPORT

Error Code:	GPORT

Error Description:	The user is active on a point-to-point BRI (rIDSLEQUIP.briclass = BRI_PP) or restrictive multipoint BRI (rIDSLEQUIP.briclass = BRI_MP), but the ok_term bit in the rITEDB tuple is incorrectly marked DBNO. For non-default service rITEDB tuples associated with restrictive multipoint, or point-to-point circuit switched calls (rITEDB.sc = DMSC_CKT and rITEDB.dft_grp = 0 and rITEDB.lport = rIDSEQUIP.bind_port), the ok_term bit should be marked DBYES if there is an active user (rITEDB.csv_hunt = DBYES or rITEDB.csd_hunt = DBYES).
Possible Error Effect:	The active user will be blocked for call terminations.
Corrective Action Taken:	The ok_term bit in the rITEDB tuple is marked DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rITEDB tuple address.
<i>Logical Index =</i>	rITEDB tuple's key.
<i>Bad Data =</i>	The ok_term bit as found in RLTEDB -- DBNO (0).
<i>Good Data =</i>	The value the ok_term bit was corrected to -- DBYES (1).
<i>Dump =</i>	The contents of the rITEDB tuple.

Table 17 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The TEDB sca_dalb field is null while the soft_dalb is not null. This indicates that a party has been put on hold while there is no other call associated with the TEDB.
Possible Error Effect:	Party may be left on hold indefinitely with no indication to the console.
Corrective Action Taken:	TEDB is zapped and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to rITEDB tuple.
<i>Bad Data =</i>	sca_dalb field of rITEDB tuple.
<i>Good Data =</i>	soft_dalb field of rITEDB tuple.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	rITEDB tuple.

Table 18 Error Code: LPHBAD

Error Code:	LPHBAD
Error Description:	The logical PH value in the TEDB is not consistent with the value in the DSLEQUIP static data.
Possible Error Effect:	Switch may not communicate with customer premise equipment.
Corrective Action Taken:	The attribute in TEDB is updated correctly.
Manual Action Required:	None.
Dump Description:	
<i>Logical Index =</i>	TEDB key.
<i>Good Data =</i>	NA
<i>Bad Data =</i>	lph attribute found in TEDB.
<i>Good Data =</i>	dsl_group attribute found in DSLEQUIP relation.
<i>Dump =</i>	RLTEB.

Table 19 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	An inconsistency was found in the port name in the TEDB tuple. An attempt to convert the port name from logical port to physical port was unsuccessful.
Possible Error Effect:	Calls may be set-up incorrectly, and attempts to make calls on call appearances related to the CPE associated with this TEDB may not be successful.
Corrective Action Taken:	TEDB is zapped out and placed on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	TEDB lport attribute.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	RLTEDB.

Table 20 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	A semantic data inconsistency was found between TEDB and RLLPT_PT.
Possible Error Effect:	CPE may not be able to utilize the features subscribed such as B-channels and MLHG.
Corrective Action Taken:	The semantic data in the TEDB is corrected to be consistent with RLLPT_PT.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	TEDB lport attribute.
<i>Dump =</i>	RLTEB.

Table 21 Error Code: LPT_PT

Error Code:	LPT_PT
Error Description:	The attributes in TEDB are not set correctly per static relation RLLPT_PT.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	The attributes are set correctly per static relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 22 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The PH member value in the TEDB is not consistent with the value in the DSLEQUIP static data.
Possible Error Effect:	Switch may not communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Logical Index =</i>	TEDB Key.
<i>Bad Data =</i>	mem attribute found in TEDB.
<i>Good Data =</i>	Member attribute found in DSLEQUIP relation.
<i>Dump =</i>	RLTEB.

Table 23 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The TEDB audit cannot access the RLLTMSTAT dynamic relation with the ltm_idx field from a valid rIORT_GROUP static data tuple.
Possible Error Effect:	The TEDB audit cannot complete all of its checks.
Corrective Action Taken:	Schedule the related LTMSTAT audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of local pointer to rITEDB tuple.
<i>Bad Data =</i>	ltm_idx field of tuple returned from database primitive.
<i>Good Data =</i>	ltm_idx field from associated static rIORT_GROUP tuple.
<i>Logical Index =</i>	TEDB key.
<i>Dump =</i>	rITEDB tuple.

Table 24 Error Code: MSG_PTR

Error Code:	MSG_PTR
Error Description:	TEDB should not forward every message to the designated ACSR process but fwd_msgs flag is set to.

Possible Error Effect:	CPE will forward the message to wrong process and lose communication.
Corrective Action Taken:	fwd_msgs field is reset to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	fwd_msgs in TEDB.
<i>Good Data =</i>	DBNO (0).
<i>Dump =</i>	RLTEDB.

Table 25 Error Code: NOACCBLK

Error Code:	NOACCBLK
Error Description:	The blkcall attribute in this rITEDB tuple was found incorrectly marked DBNO. If the user is being intercepted (rIFC_LINE.intercept = DBYES) or this is a non-default service RLTEDB (no default service group is associated with it such that rITEDB.dflt_grp = 0), and the user service order profile (USOP) indicates that an automatic customer station rearrangement (ACSR) request is currently in progress (AUspsid_buf.blk_call = DBYES), then the blkcall bit in RLTEDB should be DBYES, indicating that calls to this port should be blocked.
Possible Error Effect:	The user will be allowed access to unsubscribed services.
Corrective Action Taken:	The blkcall field in RLTEDB is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rITEDB tuple address.
<i>Bad Data =</i>	The blkcall field found in RLTEDB (DBNO).
<i>Good Data =</i>	The correct value of the blkcall attribute in RLTEDB (DBYES).
<i>Logical Index =</i>	The rITEDB key (tedb).
<i>Dump =</i>	The contents of the rITEDB tuple.

Table 26 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	The TEDB active bit is set to indicate not active when there is a selected call appearance DALB or all the call appearances available are busy.
Possible Error Effect:	Wrong information may be indicated to attendant console.
Corrective Action Taken:	TEDB active field is reset to indicate busy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Not an address; this is the value of internal variable used to indicate if all call appearances are busy or at least one is idle. 0 = idle. 1 = busy.
<i>Bad Data =</i>	TEDB active field (0 = idle, 1 = busy).
<i>Good Data =</i>	Value returned from macro reading busy/idle status of last call appearance status accessed by audit. The status is read from the RLTMSTAT relation.
<i>Logical Index =</i>	TEDB key.
<i>Dump 1 =</i>	RLTEDB.
<i>Dump 2 =</i>	LTMSTAT.

Table 27 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The user is inactive on a point-to-point BRI or inactive on a restrictive multipoint BRI (rIDSLEQUIP.briclass = BRI_MP or rIDSLEQUIP.briclass = BRI_PP, and rITEDB.lport != rIDSEQUIP.bind_port) but the ok_term bit in the rITEDB tuple is incorrectly marked DBYES. For non-default service rITEDB tuples associated with restrictive multipoint, or point-to-point circuit switched calls (rITEDB.dflt_grp = 0 and rITEDB.sc = DMSC_CKT) the ok_term bit should be marked DBNO if there is no active user.

Possible Error Effect:	An inactive user will be allowed call terminations.
Corrective Action Taken:	The ok_term bit in RLTEDB is set to DBNO.
Manual Action Required:	None
Dump Description:	
<i>Error Address =</i>	The rITEDB tuple's address.
<i>Logical Index =</i>	The rITEDB tuple's key (tedb).
<i>Bad Data =</i>	The ok_term attribute in rITEDB (DBYES).
<i>Good Data =</i>	The value the ok_term attribute was corrected to (DBNO).
<i>Dump =</i>	The contents of the rITEDB tuple.

Table 28 Error Code: ODD_DATA

Error Code:	ODD_DATA
Error Description:	The attributes in TEDB are not set correctly per static relation RLIS_TERM.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	The attributes are set correctly per static relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	TEDB tuple address.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Dump =</i>	RLTEDB.

Table 29 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	RLSPID or RLLPT_PT tuple is missing in the ODD.
Possible Error Effect:	CPE cannot utilize the subscribed features.
Corrective Action Taken:	None.
Manual Action Required:	Need to insert the missing SPID or LPT_PT tuple. (If missing SPID tuple, get lport from GOOD DATA and access relation rLPT_SPID which contains the key of SPID tuple) (The key of RLLPT_PT can be taken out of lport in GOOD DATA).
Dump Description:	
<i>Error Address =</i>	NA.
<i>Logical Index =</i>	TEDB key.
<i>Bad Data =</i>	The relation ID or RLSPID or RLLPT_PT.
<i>Good Data =</i>	The logical port name.
<i>Dump =</i>	RLTEDB.

Table 30 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The user is active on a pure multipoint BRI (rIDSLEQUIP.bri_class = BRI_PP) but the ok_term bit in RLTEDB is set to DBNO. For pure multipoint service, PPD (Permanent Packet D) rITEDB tuples (rITEDB.sc = DMSC_PPD), the ok_term attribute should be DBYES when a user is active (rITEDB.csv_hunt = DBYES or rITEDB.csd_hunt = DBYES).
Possible Error Effect:	The active user will be blocked for call terminations.
Corrective Action Taken:	The ok_term bit in rITEDB is set to DBYES.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rITEDB tuple's address.
<i>Bad Data =</i>	The ok_term attribute found in RLTEDB (DBNO).
<i>Good Data =</i>	The value that the ok_term attribute was corrected to (DBYES).
<i>Logical Index =</i>	The rITEDB tuple's key(tedb).
<i>Dump =</i>	The contents of the rITEDB tuple.

Table 31 Error Code: PORT

Error Code:	PORT
Error Description:	The TEDB has encountered an ODD inconsistency when trying to access the static relation RLPORT_GROUP. The inconsistency is due to using the global variable GI_pcrd (module number) and port field from the ODD relation

	RLLPT_PT to access the RLPORT_GROUP relation.
Possible Error Effect:	Conversions of logical to physical port may be incorrect.
Corrective Action Taken:	TEDB tuple being checked is zapped and placed on idle list.
Manual Action Required:	Check the rLPORT_GROUP tuple with port from RLLPT_PT relation (listed in <i>Good Data</i>) and the module number for this SM.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rLTEDB tuple.
<i>Bad Data</i> =	rLPORT_GROUP.port.member field, part of key to rLPORT_GROUP.
<i>Good Data</i> =	Associated rLPT_PT.port field.
<i>Logical Index</i> =	TEDB key.
<i>Dump 1</i> =	rLTEDB tuple.
<i>Dump 2</i> =	rLPT_PT tuple.

Table 32 Error Code: RDSC

Error Code:	RDSC
Error Description:	The service order profile (SOP) capability is not allowed by the system.
Possible Error Effect:	CPE will lose the feature which it subscribes.
Corrective Action Taken:	The TEDB tuple will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	SOP capability in TEDB.
<i>Good Data</i> =	NA.
<i>Dump</i> =	RLTEDB.

Table 33 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	The attributes in TEDB such as cpetype are not set to default value if no IS_TERM static tuple exists.
Possible Error Effect:	CPE may not be able to communicate with switch.
Corrective Action Taken:	The attributes are correctly set to default value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Dump</i> =	RLTEDB.

Table 34 Error Code: SOPCAP

Error Code:	SOPCAP
Error Description:	The SOP capability in the TEDB is not set to correct feature.
Possible Error Effect:	CPE cannot utilize the subscribed feature.
Corrective Action Taken:	The SOP capability is reset to the right feature per static data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	SOP capability in TEDB.
<i>Good Data</i> =	The right SOP (DMCKT_SOP, DMPKT_SOP, DMODP_COP).
<i>Dump</i> =	RLTEDB.

Table 35 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	The semantic linkage between TEDB and ATTMBLF does not point back to each other.
Possible Error Effect:	Attendant console is OOS because of semantic linkage error.
Corrective Action Taken:	DSL will be reinitialized and attendant console will be initialized to be in-service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.

<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	tedb KEY in ATTMBLF.
<i>Good Data</i> =	tedb KEY.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	RLATTM_BLF.

Table 36 Error Code: STATUS_BAD

Error Code:	STATUS_BAD
Error Description:	Either the audit was unable to retrieve the correct telephone monitoring status (expected value of rITEDB.tsm_stat) or the telephone monitoring status stored in the rITEDB tuple was incorrect.
Possible Error Effect:	Telephone monitoring may not work correctly for an end user.
Corrective Action Taken:	If the audit was able to get the correct status, it will correct the value in the rITEDB tuple (rITEDB.tsm_stat). Otherwise the audit will reinitialize the DSL and the attendant console to be in-service.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	A boolean indicating whether or not the audit was able to retrieve the correct telephone monitoring status for the rITEDB being audited.
<i>Bad Data</i> =	The telephone monitoring status value from the rITEDB tuple (rITEDB.tsm_stat).
<i>Good Data</i> =	The correct telephone monitoring status if available.
<i>Logical Index</i> =	The logical key of the rITEDB tuple (rITEDB.tedb).
<i>Dump</i> =	The rITEDB tuple.

Table 37 Error Code: TEDB

Error Code:	TEDB
Error Description:	The DALB owned by this TEDB does not think it belongs to the TEDB.
Possible Error Effect:	Switch cannot communicate with customer premise equipment.
Corrective Action Taken:	TEDB is zapped and placed on idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DALB linkage field to TEDB.
<i>Good Data</i> =	TEDB linkage field to TEDB.
<i>Logical Index</i> =	Port number of D-port.
<i>Dump</i> =	RLTEDB.

Table 38 Error Code: TEDBDISP

Error Code:	TEDBDISP
Error Description:	A TEDB believes there is a DALB linked to it which contains the display terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	Display information on the customer premise equipment associated with this tuple may not be correct. Communication to the CPE may be prohibited.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	DALB member linkage.
<i>Good Data</i> =	TEDB owner linkage.
<i>Dump</i> =	RLTEDB.

Table 39 Error Code: TEDBFEAT

Error Code:	TEDBFEAT
Error Description:	A TEDB believes there is a DALB linked to it which contains the feature terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	Switch may not correctly set up certain feature calls on the related customer premise equipment.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	NA.
<i>Logical Index</i> =	TEDB Key.
<i>Bad Data</i> =	DALB member linkage.
<i>Good Data</i> =	TEDB owner linkage.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	RLDALB.

Table 40 Error Code: TEDBSCA

Error Code:	TEDBSCA
Error Description:	A TEDB believes there is a DALB linked to it which contains the current selected call appearance terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	The selected call appearance capability on the customer premise equipment associated with this TEDB may not work. Communication to the CPE may be affected.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Logical Index</i> =	TEDB Key.
<i>Bad Data</i> =	DALB member linkage.
<i>Good Data</i> =	TEDB owner linkage.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	RLDALB.

Table 41 Error Code: TEDBSOFT

Error Code:	TEDBSOFT
Error Description:	A TEDB believes there is a DALB linked to it which contains the soft held terminal process ID. However, the DALB is not linked back to the TEDB.
Possible Error Effect:	The soft hold capability on the customer premise equipment associated with this TEDB may not work. Calls on hold may be lost, and communication to the CPE may be affected.
Corrective Action Taken:	TEDB will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Logical Index</i> =	TEDB Key.
<i>Bad Data</i> =	DALB member linkage.
<i>Good Data</i> =	TEDB owner linkage.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	RLDALB.

Table 42 Error Code: TEILALINK

Error Code:	TEILALINK
Error Description:	TEDB is not correctly linked to TEILA through semantic linkage.
Possible Error Effect:	CPE will lose the feature which it subscribes and call processing may assert when message exchanges between CPE and switch.
Corrective Action Taken:	The TEDB tuple will be reinitialized and DSL will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	TEDB tuple address.
<i>Logical Index</i> =	TEDB key.
<i>Bad Data</i> =	tedb stored at TEILA.
<i>Good Data</i> =	tedb key should be stored at TEILA.
<i>Dump 1</i> =	RLTEDB.
<i>Dump 2</i> =	RLTEILA.

Table 43 Error Code: TYPE

Error Code:	TYPE
Error Description:	The TEDB has encountered an ODD inconsistency when trying to access the static relation RLIS_LINE. The inconsistency is due to using the global variable

	Gl_pcrd (module number) and ltm_idx field from the associated rIPORT_GROUP ODD tuple to access the RLIS_LINE relation.
Possible Error Effect:	The TEDB audit cannot complete all of its checks.
Corrective Action Taken:	TEDB tuple being checked is zapped and placed on idle list.
Manual Action Required:	Check the rIPORT_GROUP tuple and the RLIS_LINE tuple with information given.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to rITEDB tuple.
<i>Bad Data</i> =	rIPORT_GROUP.ltm_idx field (used to access RLIS_LINE).
<i>Good Data</i> =	Return from database access primitive.
<i>Logical Index</i> =	TEDB key.
<i>Dump 1</i> =	rITEDB tuple.
<i>Dump 2</i> =	associated rIPORT_GROUP tuple.

Table 44 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A TEDB which is linked to the idle list (DBHEADC) has non-null value in its other linkage fields.
Possible Error Effect:	There may be problems in communicating to a CPE if the TEDB is linked to a DPB.
Corrective Action Taken:	Linkage fields in the TEDB tuple are set to null, except for member linkage to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	TEDB owner head linkage field.
<i>Good Data</i> =	TEDB owner tail linkage field.
<i>Logical Index</i> =	TEDB key.
<i>Dump</i> =	RLTEDB.

TEILA-A

Software Release: 5E10 - 5E11

1. TERMINAL ENDPOINT IDENTIFIER (TEI) LINKAGE AREA (TEILA) AUDIT

The audit runs on the operating system for distributed switching-switching module (OSDS-SM) environment under the segmented and non-segmented modes.

2. DATA STRUCTURES

The RLTEILA is an indexed sequential relation. The rLTEILA tuple contains the signaling link (tei, multi-frame establishment) information for every TEI assigned, and contains the binding data such as semantic linkage to the terminal equipment data block (TEDB) and customer premises equipment (CPE) types. After full initialization, all the available TEILA tuples are initialized and linked to the database head cell (DBHEADC) to serve as the idle pool. When a TEI is assigned, one TEILA tuple is retrieved from the idle list and linked to the D-channel port block (DPB). Binding, which associates each TEI with the services subscribed, TEDB, occurs when the CPE is performing end point initialization. Data structures and their linkages are shown in Figure 1 and Figure 2.

3. AUDIT CHECKS

The TEILA audit checks the linkage for the idle TEILA tuple, which is linked to the DBHEADC, and for in-use TEILA which is linked to the DPB. After passing quarantine, key and linkage tests, the TEILA audit verifies the semantic linkage with the TEDB and checks the binding for this TEILA. Different binding rules are used for different types of the basic rate interface (BRI). Generally speaking, the TEILA audit uses a static relation such as the service profile identifier (SPID) and the digital subscriber line equipment (DSLEQUIP), to check whether the customer premises equipment (CPE) (represented by the TEILA audit) is properly bound (linked) to the service subscribed (represented by the TEDB audit).

4. ERROR HANDLING

Any binding error, or semantic and relational definition language (RDL) linkage error results in the recovery of the digital subscriber line (DSL) using the to-be-idle list recovery interface with terminal maintenance (TM).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The TEILA tuple is supposed to be in the idle list but the linkage has an error.
Possible Error Effect:	The TEILA tuple cannot be used for call processing.
Corrective Action Taken:	The TEILA tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLTEILA.
<i>Bad Data =</i>	Linkage fields to DBHEADC.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rLTEILA.
<i>Dump =</i>	rLTEILA tuple.

Table 2 Error Code: BRADPROTOCOL

Error Code:	BRADPROTOCOL
Error Description:	The CPE protocol field, <i>cpeprot</i> , in the TEILA tuple indicates failed full initializing protocol (FFIP). However, this CPE is on a multipoint DSL where FFIP is not allowed.

Possible Error Effect:	The TEILA tuple cannot be used for call processing.
Corrective Action Taken:	The TEILA tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The value of the enumeration representing FFIP.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rIDSLEQUIP tuple.

Table 3 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The rTEILA tuple contains the wrong non-call appearance request count, <i>nca_count</i> , used in the automatic terminal setup (ATS) feature. This count represents the number of register message requests. This count is verified by counting rDALB tuples linked to the rDPB tuple for this port and the number of rAPDU tuples linked to the message inbound queue (MIB) for this port.
Possible Error Effect:	Terminal down load requests may be prevented or lost.
Corrective Action Taken:	The count is corrected based on the number of rDALB and rAPDU tuples found for this port.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The value of the <i>nca_count</i> before correction.
<i>Good Data =</i>	The value of the <i>nca_count</i> after correction. NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rDPB tuple.

Table 4 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The overall binding data for this terminal is inconsistent. The boundtyp is NOT_BOUND but one or more of the semantic linkages to TEDB is not null.
Possible Error Effect:	The terminal will not be able to originate/terminate calls properly. Call processing will fail.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 5 Error Code: BINDWRONG

Error Code:	BINDWRONG
Error Description:	The bind_wrong field in the TEILA is improperly set to DBYES.
Possible Error Effect:	The bind_wrong field is used to indicate when terminal binding is wrong due to loss and reassignment of TEI's while active calls are on the terminal. A possible case is during ACSR moves. Rebinding is needed once active calls go away on the terminal.
Corrective Action Taken:	Reset the bind_wrong field to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	DBYES.
<i>Good Data =</i>	Return value from Rlusrquery().
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 6 Error Code: CAMPSTATE

Error Code:	CAMPSTATE
Error Description:	The blkcall field is DBYES or the campstat field is not DMNO_CAMP and the PORTLA is not linked to a process.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	DBYES.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rPORTLA tuple.

Table 7 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	When verifying the nca_count in the rTEILA for non-call appearances, the audit found corrupt rDALB linkages off the rDPB. The value of good data indicates if the first rDALB has bad linkage or an rDALB after the first one had bad linkage. If good data is 0, we could not get the first rDALB in the list. If good data is 1, we could not obtain the successor to the rDALB that we just processed.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The audit error is reported and the related DALB audit is scheduled to clean up the rDALB bad linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The logical key of the rDALB tuple with bad linkage.
<i>Good Data =</i>	0 or 1.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rDPB tuple if good data = 0.
<i>Dump 3 =</i>	rDALB tuple with bad successor linkage if good data = 1.

Table 8 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The terminal is bound to default service but the semantic linkages to TEDB are not set correctly.
Possible Error Effect:	The terminal will not be able to operate properly. This will affect the invoking of services supported by default services such as ACSR or 911.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 9 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The TEILA tuple is supposed to link to DPB but the linkage is in error.
Possible Error Effect:	Application codes may assert and messages may be erroneously delivered.
Corrective Action Taken:	The audit idles the rTEILA tuple and should result in the recovery of the DSL.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Linkage fields to DPB.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 10 Error Code: DSLEQUIP

Error Code:	DSLEQUIP
Error Description:	The static DSLEQUIP tuple could not be read given the owning DPB of the TEILA.
Possible Error Effect:	The TEILA may have a linkage error or there may be an ODD inconsistency. The terminal will not be able to originate or terminate calls.
Corrective Action Taken:	None. Return to audit the next TEILA.
Manual Action Required:	If this error rolls, check the DSLEQUIP static ODD relation.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 11 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rTEILA is in the idle state but is not linked to the rTEILA idle list.
Possible Error Effect:	The rTEILA is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rTEILA tuple and relinks back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Owner link of rTEILA.
<i>Good Data</i> =	Physical key of idle head cell.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 12 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The semantic linkage between TEILA and TEDB has failed a point-to-point-back check or the TEDB and TEILA do not have the same DPB owner.
Possible Error Effect:	The binding for this terminal is not correct which will cause call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump 1</i> =	rTEILA tuple.
<i>Dump 2</i> =	rTEDB tuple.

Table 13 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The D-channel PORTLA could not be accessed given the DPB owner.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures. There may be database problems with this DSL.
Corrective Action Taken:	The PORTLA audit is scheduled to try to find the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	PORTLA key returned by DA.

<i>Good Data</i> =	DPB owner key of TEILA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 14 Error Code: MP_BINDERR

Error Code:	MP_BINDERR
Error Description:	There is a semantic data binding error for a terminal on a multipoint DSL. The value in <i>Bad Data</i> determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA
<i>Bad Data</i> =	1 = The boundtyp is DMBIP_BOUND and the terminal is bound to default service but the ptyequip in DSLEQUIP reflects that users are provisioned on the DSL. 2 = The boundtyp is DMBIP_BOUND but the cpeprot has been reset to DMNIP_CPE. 3 = The boundtyp is DMDFLT_BOUND. However, the terminal is not bound to default service, and both the circuit and packet default DNs in the corresponding rDSLEQUIP are zero. Therefore, the rTEILA can not be default DN bound. 4 = The boundtyp is DMDFLT_BOUND. However, the rTEILA tuple is bound to the wrong rTEDB tuples based on the lports obtained from the corresponding rDSLEQUIP tuple. 5 = The boundtyp is DMFTEI_BOUND. However, the rTEILA tuple is associated with a non-standard DSL. This type of binding is only allowed on standard DSLs. 6 = The boundtyp is DMNIP_BOUND but the cpeprot is not DMNIP_CPE, DMFIP_CPE, or DMFFIP_CPE. 7 = The boundtyp is DMFIP_BOUND but the cpeprot is not DMFIP_CPE. 8 = The rTEILA is not bound to the default service rTEDB. However, the rTEILA tuple does not indicate that spid_pop is DBYES or that uspid.digcnt is non-zero. 9 = The rTEILA is bound to the default service rTEDB. However, the rTEILA is DMFIP_BOUND with uspid.digcnt = 0 or DMNIP_BOUND with spid_pop = DBNO. These cases are invalid and can affect 911 or ACSR. 10 = The boundtyp is DMFTEI_BOUND. However, the rTEILA is bound to the default service rTEDB. This is not allowed for this bind type. 11 = The dsl in the corresponding rSPID tuple does not match the DPB owner for this rTEILA tuple. 12 = One or more of the lports in SPID is null but the corresponding semantic TEDB linkage is not null.

	13 = One or more of the lports in SPID is non-null but the corresponding semantic TEDB linkage does not match. 14 = The boundtyp is DMDFLT_BOUND. The corresponding rISPID tuple has its default DN indicator, <i>ddn</i>, set to DBNO. This field must be set to DBYES for this bind type. 15 = The boundtyp is either DMFIP_BOUND or DMFTEI_BOUND. The corresponding rISPID tuple has its default DN indicator, <i>ddn</i>, set to DBYES. This field must be set to DBNO for these bind types. 16 = The boundtyp may or may not be DMFTEI_BOUND. There is an inconsistency between the bind type in the rTEILA tuple and the <i>pfitei</i> flag in the corresponding rISPID tuple. This <i>pfitei</i> flag must be set to DBYES for DMFTEI_BOUND and DBNO otherwise. 17 = The boundtyp is DMFTEI_BOUND. The <i>lport</i> associated with this fixed TEI resource is not the same one identified in the corresponding rISPID tuple. The <i>lport</i> should match the one found in the <i>ppd_lport</i> field of the rISPID tuple. 18 = The <i>natype</i>, <i>npid</i>, or <i>epdn</i> fields in rTEILA do not match the corresponding fields in the associated rLPT_SPID tuple. 19 = The boundtyp is out of range.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 15 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit failed to access static data from a given relation based on other static data. This error can occur from five separate error legs. This is outlined in the <u>good/bad data fields</u> .
Possible Error Effect:	The ODD is inconsistent and the terminal will not function properly due to operational and maintenance failures.
Corrective Action Taken:	None.
Manual Action Required:	The error code will indicate to the craft that manual action is required. The ODD inconsistency should be pinpointed and corrected. This error will continue to be printed until fixed.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Relation ID of static data which could not be read (that is, SPID, LPT_SPID).
<i>Good Data</i> =	Valid values are: 0 = Point-to-point DSL. 1 = Multipoint DSL or standard DSL. 2 = Restricted multipoint DSL.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 16 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The cpestat is in an OOS state (not TMC_IS) and this is a custom point-to-point.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Cpestat in TEILA.
<i>Good Data =</i>	TMC_IS.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 17 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The audit was not able to obtain the rIDPB associated with this rTEILA. This would imply a linkage problem with the rIDPB.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The audit error is reported and the related DPB audit is scheduled to correct any linkage problems found in the rIDPB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The bad key of the rIDPB returned from the DAGETDPB call.
<i>Good Data =</i>	The value of the physical linkage field to the rIDPB in the rTEILA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 18 Error Code: PP_BINDERR

Error Code:	PP_BINDERR
Error Description:	The binding information in the TEILA semantic data is not consistent with respect to the boundtyp. This terminal is associated with a point-to-point DSL. The value in "BAD DATA" determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	The terminal will not function properly.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Valid values are: 1 = The boundtyp is DMNOT_BOUND but the cpeprot is not DMNULL_CPE and DMBIP_CPE. 2 = The boundtyp is DMBIP_BOUND but the cpeprot is not DMBIP_CPE. 3 = The boundtyp is DMBIP_BOUND and the lport in SPID does not match the associated TEDB key (semantic linkage). 4 = The boundtyp is DMBIP_BOUND and the c_mf_est is DBYES but none of the circuit TEDB keys (semantic linkage) match the lports in SPID. 5 = The boundtyp is DMBIP_BOUND and the p_mf_est is DBYES but the packet TEDB key (semantic linkage) does not match the lport in SPID. 6 = The <i>epdn</i> , <i>natype</i> , <i>npid</i> , or <i>uspid</i> in rTEILA do not match the same in rSPID or rLPT_SPID, or the spid_pop field is not DBYES.

	7 = The boundtyp is in an invalid state for this BRI class.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 19 Error Code: PP_BINDTYPE

Error Code:	PP_BINDTYPE
Error Description:	The binding information for a point-to-point DSL is incorrect. This data is a combination of static (DSLEQUIP) and dynamic (TEILA) data. The value in <i>Bad Data</i> determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	This terminal will not be able to function properly.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Valid values are: 1 = The bind_lport field in DSLEQUIP is not null but the TEILA is linked to a TEDB for default service. 2 = The c_mf_est or p_mf_est fields are DBYES and the boundtyp is DMNOT_BOUND. 3 = The bind_lport in DSLEQUIP is null and the dflt_lport in DSLEQUIP is not null and the c_mf_est field is DBYES but the boundtyp field is DMNOT_BOUND. 4 = The bind_lport in DSLEQUIP is null and the dflt_lport in DSLEQUIP is null but the bound type is not DMNOT_BOUND.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 20 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The audit could not obtain the count of rAPDU tuples linked off the MIB queue by calling PCcntmib(). This would imply that the linkages in the rAPDU tuples are corrupt.
Possible Error Effect:	This terminal will not be able to function properly.
Corrective Action Taken:	The audit error is reported and the related APDU audit is scheduled to correct any linkage problem with the rAPDU tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Return code from PCcntmib().
<i>Good Data</i> =	The value of the <i>nca_count</i> field in the rTEILA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump 1</i> =	rTEILA tuple.
<i>Dump 2</i> =	rDPB tuple.

Table 21 Error Code: RMP_BINDERR

Error Code:	RMP_BINDERR
Error Description:	There is a semantic data binding error for a terminal on a restricted multipoint DSL. The value in <i>Bad Data</i> determines the nature of the error. All data fields

	referenced are in TEILA unless otherwise stated.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Valid values are: 1 = The boundtyp is DMNOT_BOUND but the dflt_lport in DSLEQUIP is not null and the c_mf_est is DBYES. 2 = The boundtyp is DMBIP_BOUND and one or more of the lports in SPID are null but the corresponding semantic TEDB linkage is not null. 3 = The boundtyp is DMBIP_BOUND and the c_mf_est is DBYES but one of the circuit related lports in SPID does not match the semantic TEDB linkage. 4 = The boundtyp is DMBIP_BOUND and the p_mf_est is DBYES but the packet related lport in SPID does not match the semantic TEDB linkage. 5 = The boundtyp is in an invalid state for a point-to-point DSL.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 22 Error Code: SOPCAP

Error Code:	SOPCAP
Error Description:	The service order profile (SOP) capability field (sc) in the TEILA does not match that in the TEDB.
Possible Error Effect:	The features assigned for this terminal are inconsistent and will cause call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	sc field in TEILA.
<i>Good Data =</i>	sc field in TEDB.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 23 Error Code: SPID

Error Code:	SPID
Error Description:	The cpeprot is DMNIP_CPE but the spid_pop field is DBNO.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Spid_pop in TEILA.
<i>Good Data =</i>	DBYES.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 24 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The TEILA tuple is supposed to link to TEDB but the implied linkage is wrong.

Possible Error Effect:	The application codes may assert and messages may be erroneously delivered.
Corrective Action Taken:	The audit idles the rTEILA tuple and should result in the recovery of the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	TEILA key in the TEDB.
<i>Good Data =</i>	Key of TEILA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rTEDB tuple.

Table 25 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	There is an inconsistency between the bind type of the TEILA tuple and the resource it represents. If the TEILA tuple represents a FTEI, its binding is not consistent with this type of resource. If the TEILA tuple does not represent a FTEI resource, its binding indicates the type reserved for a FTEI resource.
Possible Error Effect:	The terminal on this DSL may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The address of the rTEILA tuple.
<i>Good Data =</i>	This is 1 if this is a FTEI resource and 0 otherwise.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	If <i>Good Data</i> is 1, this is the rIPFIXTEI tuple.

Table 26 Error Code: TID

Error Code:	TID
Error Description:	For all custom DSL types, the tei and tid fields should not both be set to broadcast TEI (PCBRDTEI). For the standard DSL type, the tid field should be in the range of RITID_MIN_STD to RITID_MAX_STD when the CPE is bound. The audit determined that this was not the case.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	PCBRDTEI.
<i>Good Data =</i>	Valid values are: 1 = Standard DSL. 2 = Custom DSL.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 27 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rTEILA is in the idle state but it still links to rTEDB by implied linkage.
Possible Error Effect:	TEDB is linked to a rTEILA on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rTEILA tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.

<i>Bad Data</i> =	The first linkage field to TEDB.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

TEILA-B

Software Release: 5E12 and later

1. TERMINAL ENDPOINT IDENTIFIER TEI LINKAGE AREA (TEILA) AUDIT

It runs on the operating system for distributed switching-switching module (OSDS-SM) environment. The audit runs under the segmented and non-segmented modes.

2. DATA STRUCTURES

The RLTEILA is an indexed sequential relation. The TEILA tuple contains the signaling link (tei, multi-frame establishment) information for every terminal endpoint identifier (TEI) assigned, and contains the binding data such as semantic linkage to the terminal equipment data block (TEDB) and customer premises equipment (CPE) types. After full initialization, all the available TEILA tuples are initialized and linked to the database head cell (DBHEADC) to serve as the idle pool. When a TEI is assigned, one TEILA tuple is retrieved from the idle list and linked to the D-channel port block (DPB). binding, which associates each TEI with the services subscribed, TEDB, occurs when the CPE is performing end point initialization. Data structures and their linkages are shown in Figure 1 and Figure 2

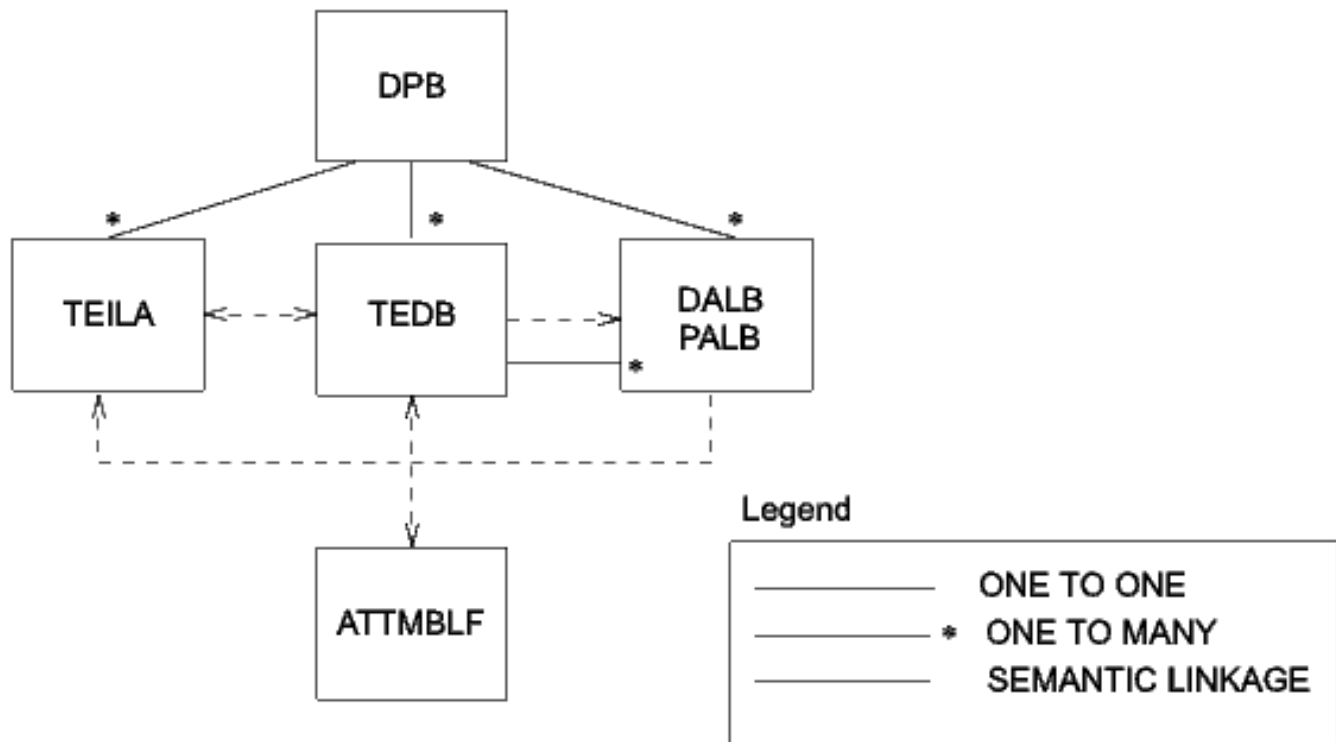


Figure 1 (TEILA-1) Data Structure of TEILA Audit Showing Ordered Set Member Linkages

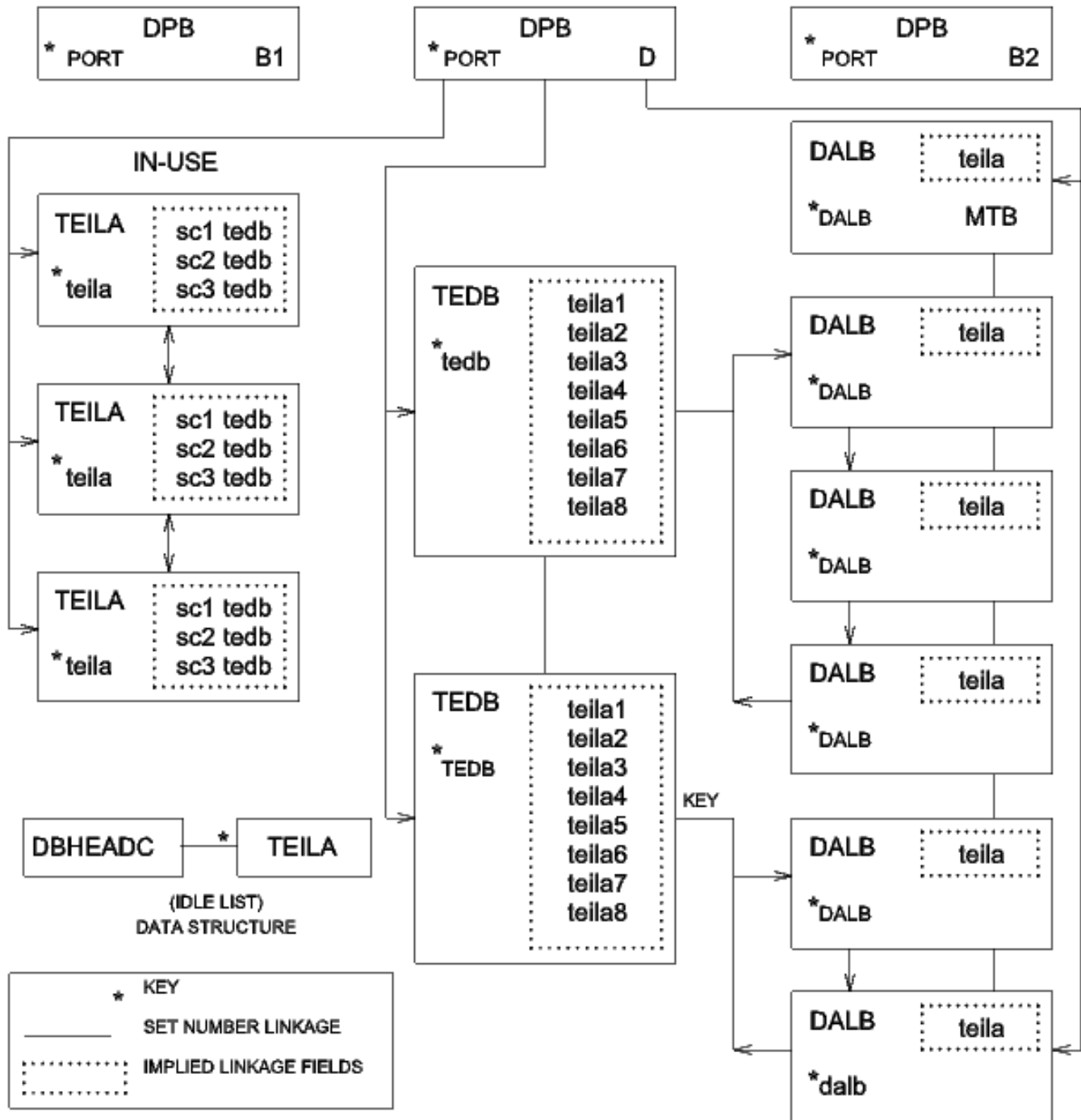


Figure 2 (TEILA-2) Data Structure of TEILA Audit in Detail

3. AUDIT CHECKS

The TEILA audit checks the linkage for the idle TEILA tuple, which is linked to the DBHEADC, and for in-use TEILA which is linked to the DPB. After passing quarantine, key and linkage tests, the TEILA audit verifies the semantic linkage with the TEDB and checks the binding for this TEILA. Different binding rules are used for different types of the basic rate interface (BRI). Generally speaking, the TEILA audit uses a static relation such as the service profile identifier (SPID) and the digital subscriber line equipment (DSLEQUIP), to check whether the customer premises

equipment (CPE) (represented by the TEILA audit) is properly bound (linked) to the service subscribed (represented by the TEDB audit).

4. ERROR HANDLING

Any binding error, or semantic and relational definition language (RDL) linkage error results in the recovery of the digital subscriber line (DSL) using the to-be-idle list recovery interface with terminal maintenance (TM).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: BAD_LINK_DATA

Error Code:	BAD_LINK_DATA
Error Description:	A rTEILA tuple contains the key of an rIGPFTIMER tuple in one of its timer key fields but the rIGPFTIMER tuple is either not linked back (rIGPFTIMER.udata contains a different relation ID or key from the owning tuple) or the timer type (rIGPFTIMER.timer_type) does not have the expected value for the timer key field being used.
Possible Error Effect:	Varies according to conditions including the rTEILA usage and which timer key field is being checked.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data =</i>	The relation ID stored in rIGPFTIMER.udata.
<i>Good Data =</i>	The tuple key stored in rIGPFTIMER.udata.
<i>Logical Index =</i>	The rTEILA tuple logical key (rTEILA.teila).
<i>Dump 1 =</i>	The rTEILA tuple.
<i>Dump 2 =</i>	The rIGPFTIMER tuple.

Table 2 Error Code: BADIDLE2

Error Code:	BADIDLE2
Error Description:	The TEILA tuple is supposed to be in the idle list but the linkage has an error.
Possible Error Effect:	The TEILA tuple cannot be used for call processing.
Corrective Action Taken:	The TEILA tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Linkage fields to DBHEADC.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 3 Error Code: BRADPROTOCOL

Error Code:	BRADPROTOCOL
Error Description:	The customer premise equipment (CPE) protocol field, <i>cpeprot</i> , in the TEILA tuple indicates failed full initializing protocol (FFIP). However, this CPE is on a multipoint digital subscriber line (DSL) where FFIP is not allowed.
Possible Error Effect:	The TEILA tuple cannot be used for call processing.
Corrective Action Taken:	The TEILA tuple is reconstructed and put in the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The value of the enumeration representing FFIP.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.

<i>Dump 2 =</i>	rIDSLEQUIP tuple.

Table 4 Error Code: BDCOUNT

Error Code:	BDCOUNT
Error Description:	The rTEILA tuple contains the wrong non-call appearance request count, <i>nca_count</i> , used in the automatic terminal setup (ATS) feature. This count represents the number of register message requests. This count is verified by counting rDALB tuples linked to the rDPB tuple for this port and the number of rAPDU tuples linked to the message inbound queue (MIB) for this port.
Possible Error Effect:	Terminal down load requests may be prevented or lost.
Corrective Action Taken:	The count is corrected based on the number of rDALB and rAPDU tuples found for this port.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The value of the <i>nca_count</i> before correction.
<i>Good Data =</i>	The value of the <i>nca_count</i> after correction. NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rDPB tuple.

Table 5 Error Code: BINDERR

Error Code:	BINDERR
Error Description:	The overall binding data for this terminal is inconsistent. The boundtyp is NOT_BOUND but one or more of the semantic linkages to TEDB is not null.
Possible Error Effect:	The terminal will not be able to originate/terminate calls properly. Call processing will fail.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 6 Error Code: BINDWRONG

Error Code:	BINDWRONG
Error Description:	The bind_wrong field in the TEILA is improperly set to DBYES.
Possible Error Effect:	The bind_wrong field is used to indicate when terminal binding is wrong due to loss and reassignment of TEI's while active calls are on the terminal. A possible case is during ACSR moves. Rebinding is needed once active calls go away on the terminal.
Corrective Action Taken:	Reset the bind_wrong field to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	DBYES.
<i>Good Data =</i>	Return value from Rlusrquery().
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 7 Error Code: CAMPSTATE

Error Code:	CAMPSTATE
Error Description:	The blkcall field is DBYES or the campstat field is not DMNO_CAMP and the PORTLA is not linked to a process.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by

	restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	DBYES.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rPORTLA tuple.

Table 8 Error Code: DALBLINK

Error Code:	DALBLINK
Error Description:	When verifying the nca_count in the rTEILA for non-call appearances, the audit found corrupt rDALB linkages off the rDPB. The value of <i>Good Data</i> indicates if the first rDALB has bad linkage or an rDALB after the first one had bad linkage. If <i>Good Data</i> is 0, we could not get the first rDALB in the list. If <i>Good Data</i> is 1, we could not obtain the successor to the rDALB that was just processed.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The audit error is reported and the related DALB audit is scheduled to clean up the rDALB bad linkage.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The logical key of the rDALB tuple with bad linkage.
<i>Good Data =</i>	0 or 1.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rDPB tuple if good data = 0.
<i>Dump 3 =</i>	rDALB tuple with bad successor linkage if good data = 1.

Table 9 Error Code: DFLINK

Error Code:	DFLINK
Error Description:	The terminal is bound to default service but the semantic linkages to TEDB are not set correctly.
Possible Error Effect:	The terminal will not be able to operate properly. This will affect the invoking of services supported by default services such as ACSR or 911.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 10 Error Code: DPB_LINK

Error Code:	DPB_LINK
Error Description:	The TEILA tuple is supposed to link to DPB but the linkage is in error.
Possible Error Effect:	Application codes may assert and messages may be erroneously delivered.
Corrective Action Taken:	The audit idles the rTEILA tuple and should result in the recovery of the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Linkage fields to DPB.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 11 Error Code: DSLEQUIP

Error Code:	DSLEQUIP
Error Description:	The static DSLEQUIP tuple could not be read given the owning DPB of the TEILA.
Possible Error Effect:	The TEILA may have a linkage error or there may be an ODD inconsistency. The terminal will not be able to originate or terminate calls.
Corrective Action Taken:	None. Return to audit the next TEILA.
Manual Action Required:	If this error rolls, check the DSLEQUIP static ODD relation.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 12 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	A rTEILA is in the idle state but is not linked to the rTEILA idle list.
Possible Error Effect:	The rTEILA is in a limbo condition, unusable by call processing.
Corrective Action Taken:	The audit idles the rTEILA tuple and relinks back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	Owner link of rTEILA.
<i>Good Data =</i>	Physical key of idle head cell.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 13 Error Code: INVOWNER

Error Code:	INVOWNER
Error Description:	The semantic linkage between TEILA and TEDB has failed a point-to-point-back check or the TEDB and TEILA do not have the same DPB owner.
Possible Error Effect:	The binding for this terminal is not correct which will cause call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump 1 =</i>	rTEILA tuple.
<i>Dump 2 =</i>	rTEDB tuple.

Table 14 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The D-channel PORTLA could not be accessed given the DPB owner.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures. There may be database problems with this DSL.
Corrective Action Taken:	The PORTLA audit is scheduled to try to find the problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	PORTLA key returned by DA.
<i>Good Data =</i>	DPB owner key of TEILA.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

Table 15 Error Code: MP_BINDERR

Error Code:	MP_BINDERR
Error Description:	There is a semantic data binding error for a terminal on a multipoint DSL. The

	value in <i>Bad Data</i> determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA
<i>Bad Data =</i>	Valid values are: 1 = The boundtyp is DMBIP_BOUND and the terminal is bound to default service but the ptyequip in DSLEQUIP reflects that users are provisioned on the DSL. 2 = The boundtyp is DMBIP_BOUND but the cpeprot has been reset to DMNIP_CPE. 3 = The boundtyp is DMDFLT_BOUND. However, the terminal is not bound to default service, and both the circuit and packet default DN's in the corresponding rDSLEQUIP are zero. Therefore, the rTEILA can not be default DN bound. 4 = The boundtyp is DMDFLT_BOUND. However, the rTEILA tuple is bound to the wrong rTEDB tuples based on the lports obtained from the corresponding rDSLEQUIP tuple. 5 = The boundtyp is DMFTEI_BOUND. However, the rTEILA tuple is associated with a non-standard DSL. This type of binding is only allowed on standard DSLs. 6 = The boundtyp is DMNIP_BOUND but the cpeprot is not DMNIP_CPE, DMFIP_CPE, or DMFFIP_CPE. 7 = The boundtyp is DMFIP_BOUND but the cpeprot is not DMFIP_CPE. 8 = The rTEILA is not bound to the default service rTEDB. However, the rTEILA tuple does not indicate that spid_pop is DBYES or that uspid.digcnt is non-zero. 9 = The rTEILA is bound to the default service rTEDB. However, the rTEILA is DMFIP_BOUND with uspid.digcnt = 0 or DMNIP_BOUND with spid_pop = DBNO. These cases are invalid and can affect 911 or ACSR. 10 = The boundtyp is DMFTEI_BOUND. However, the rTEILA is bound to the default service rTEDB. This is not allowed for this bind type. 11 = The dsl in the corresponding rSPID tuple does not match the DPB owner for this rTEILA tuple. 12 = One or more of the lports in SPID is null but the corresponding semantic TEDB linkage is not null. 13 = One or more of the lports in SPID is non-null but the corresponding semantic TEDB linkage does not match. 14 = The boundtyp is DMDFLT_BOUND. The corresponding rSPID tuple has its default DN indicator, <i>ddn</i>, set to DBNO. This field must be set to DBYES

	for this bind type. 15 = The boundtyp is either DMFIP_BOUND or DMFTEI_BOUND. The corresponding rISPID tuple has its default DN indicator, <i>ddn</i>, set to DBYES. This field must be set to DBNO for these bind types. 16 = The boundtyp may or may not be DMFTEI_BOUND. There is an inconsistency between the bind type in the rTEILA tuple and the <i>pfitei</i> flag in the corresponding rISPID tuple. This <i>pfitei</i> flag must be set to DBYES for DMFTEI_BOUND and DBNO otherwise. 17 = The boundtyp is DMFTEI_BOUND. The <i>lport</i> associated with this fixed TEI resource is not the same one identified in the corresponding rISPID tuple. The <i>lport</i> should match the one found in the <i>ppd_lport</i> field of the rISPID tuple. 18 = The <i>natype</i>, <i>npid</i>, or <i>epdn</i> fields in rTEILA do not match the corresponding fields in the associated rLPT_SPID tuple. 19 = The boundtyp is out of range.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 16 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit failed to access static data from a given relation based on other static data. This error can occur from five separate error legs. This is outlined in the <u>good/bad data fields</u> .
Possible Error Effect:	The ODD is inconsistent and the terminal will not function properly due to operational and maintenance failures.
Corrective Action Taken:	None.
Manual Action Required:	The error code will indicate to the craft that manual action is required. The ODD inconsistency should be pinpointed and corrected. This error will continue to be printed until fixed.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Relation ID of static data which could not be read (that is, SPID, LPT_SPID).
<i>Good Data</i> =	Valid values are: 0 = Indicates point-to-point DSL. 1 = Indicates multipoint DSL or standard DSL. 2 = Indicates restricted multipoint DSL.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 17 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	The cpestat is in an OOS state (not TMC_IS) and this is a custom point-to-point.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Cpestat in TEILA.
<i>Good Data</i> =	TMC IS.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 18 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The audit was not able to obtain the rDPB associated with this rTEILA. This would imply a linkage problem with the rDPB.
Possible Error Effect:	The terminal may not perform properly and result in call processing/maintenance failures.
Corrective Action Taken:	The audit error is reported and the related DPB audit is scheduled to correct any linkage problems found in the rDPB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	The bad key of the rDPB returned from the DAGETDPB call.
<i>Good Data</i> =	The value of the physical linkage field to the rDPB in the rTEILA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 19 Error Code: PP_BINDERR

Error Code:	PP_BINDERR
Error Description:	The binding information in the TEILA semantic data is not consistent with respect to the boundtyp. This terminal is associated with a point-to-point DSL. The value in <i>Bad Data</i> determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	The terminal will not function properly.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Valid values are: 1 = The boundtyp is DMNOT_BOUND but the cpeprot is not DMNULL_CPE and DMBIP_CPE. 2 = The boundtyp is DMBIP_BOUND but the cpeprot is not DMBIP_CPE. 3 = The boundtyp is DMBIP_BOUND and the lport in SPID does not match the associated TEDB key (semantic linkage). 4 = The boundtyp is DMBIP_BOUND and the c_mf_est is DBYES but none of the circuit TEDB keys (semantic linkage) match the lports in SPID. 5 = The boundtyp is DMBIP_BOUND and the p_mf_est is DBYES but the packet TEDB key (semantic linkage) does not match the lport in SPID. 6 = The epdn , natype , npid , or uspid in rTEILA do not match the same in rSPID or rLPT_SPID, or the spid_pop field is not DBYES. 7 = The boundtyp is in an invalid state for this BRI class.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 20 Error Code: PP_BINDTYPE

Error Code:	PP_BINDTYPE
Error Description:	The binding information for a point-to-point DSL is incorrect. This data is a combination of static (DSLEQUIP) and dynamic (TEILA) data. The value in <i>Bad Data</i> determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	This terminal will not be able to function properly.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Valid values are: 1 = The bind_lport field in DSLEQUIP is not null but the TEILA is linked to a TEDB for default service. 2 = The c_mf_est or p_mf_est fields are DBYES and the boundtyp is DMNOT_BOUND. 3 = The bind_lport in DSLEQUIP is null and the dflt_lport in DSLEQUIP is not null and the c_mf_est field is DBYES but the boundtyp field is DMNOT_BOUND. 4 = The bind_lport in DSLEQUIP is null and the dflt_lport in DSLEQUIP is null but the bound type is not DMNOT_BOUND.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 21 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The audit could not obtain the count of rAPDU tuples linked off the MIB queue by calling PCcntmib(). This would imply that the linkages in the rAPDU tuples are corrupt.
Possible Error Effect:	This terminal will not be able to function properly.
Corrective Action Taken:	The audit error is reported and the related APDU audit is scheduled to correct any linkage problem with the rAPDU tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Return code from PCcntmib().
<i>Good Data</i> =	The value of the <i>nca_count</i> field in the rTEILA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump 1</i> =	rTEILA tuple.
<i>Dump 2</i> =	rDPB tuple.

Table 22 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve an rGPFTIMER from the key stored in one of the rTEILA tuple's timer key fields.
Possible Error Effect:	Varies according to conditions including which timer field was being accessed.
Corrective Action Taken:	The audit clears the timer key field (sets = TMTNULLIDX).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An internal value used for debugging. It indicates which timer key field was being checked.
<i>Bad Data</i> =	The rGPFTIMER key returned from the read attempt.

<i>Good Data</i> =	The rIGPFTIMER key from the timer key field.
<i>Logical Index</i> =	The rTEILA tuple's logical key (rTEILA.teila).
<i>Dump 1</i> =	The rTEILA tuple.
<i>Dump 2</i> =	The rIGPFTIMER tuple buffer resulting from the read attempt.

Table 23 Error Code: RMP_BINDERR

Error Code:	RMP_BINDERR
Error Description:	There is a semantic data binding error for a terminal on a restricted multipoint DSL. The value in <i>Bad Data</i> determines the nature of the error. All data fields referenced are in TEILA unless otherwise stated.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	Valid values are: 1 = The boundtyp is DMNOT_BOUND but the dflt_lport in DSLEQUIP is not null and the c_mf_est is DBYES. 2 = The boundtyp is DMBIP_BOUND and one or more of the lports in SPID are null but the corresponding semantic TEDB linkage is not null. 3 = The boundtyp is DMBIP_BOUND and the c_mf_est is DBYES but one of the circuit related lports in SPID does not match the semantic TEDB linkage. 4 = The boundtyp is DMBIP_BOUND and the p_mf_est is DBYES but the packet related lport in SPID does not match the semantic TEDB linkage. 5 = The boundtyp is in an invalid state for a point-to-point DSL.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 24 Error Code: SOPCAP

Error Code:	SOPCAP
Error Description:	The SOP capability field (sc) in the TEILA does not match that in the TEDB.
Possible Error Effect:	The features assigned for this terminal are inconsistent and will cause call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	sc field in TEILA.
<i>Good Data</i> =	sc field in TEDB.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 25 Error Code: SPID

Error Code:	SPID
Error Description:	The cpeprot is DMNIP_CPE but the spid_pop field is DBNO.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.

<i>Bad Data</i> =	spid_pop in TEILA.
<i>Good Data</i> =	DBYES.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 26 Error Code: TEDBLINK

Error Code:	TEDBLINK
Error Description:	The TEILA tuple is supposed to link to TEDB but the implied linkage is wrong.
Possible Error Effect:	The application codes may assert and messages may be erroneously delivered.
Corrective Action Taken:	The audit idles the rTEILA tuple and should result in the recovery of the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	TEILA key in the TEDB.
<i>Good Data</i> =	Key of TEILA.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump 1</i> =	rTEILA tuple.
<i>Dump 2</i> =	rTEDB tuple.

Table 27 Error Code: TEIBAD

Error Code:	TEIBAD
Error Description:	There is an inconsistency between the bind type of the TEILA tuple and the resource it represents. If the TEILA tuple represents a fixed terminal endpoint identifier (FTEI), its binding is not consistent with this type of resource. If the TEILA tuple does not represent a FTEI resource, its binding indicates the type reserved for a FTEI resource.
Possible Error Effect:	The terminal on this DSL may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	The address of the rTEILA tuple.
<i>Good Data</i> =	This is 1 if this is a FTEI resource and 0 otherwise.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump 1</i> =	rTEILA tuple.
<i>Dump 2</i> =	If good data is 1, this is the rPPFIXTEI tuple.

Table 28 Error Code: TID

Error Code:	TID
Error Description:	For all custom DSL types, the tei and tid fields should not both be set to broadcast TEI (PCBRDTEI). For the standard DSL type, the tid field should be in the range of RITID_MIN_STD to RITID_MAX_STD when the CPE is bound. The audit determined that this was not the case.
Possible Error Effect:	The terminal may not perform properly and result in call processing failures.
Corrective Action Taken:	The TEILA is recovered to the idle list and the terminal is resynchronized by restoring the DSL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rTEILA.
<i>Bad Data</i> =	PCBRDTEI.
<i>Good Data</i> =	Valid values are: 1 = Standard DSL. 2 = Custom DSL.
<i>Logical Index</i> =	Logical key of rTEILA.
<i>Dump</i> =	rTEILA tuple.

Table 29 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	A rTEILA is in the idle state but it still links to rTEDB by implied linkage.
Possible Error Effect:	TEDB is linked to a rTEILA on the idle list and may be unusable by call processing.
Corrective Action Taken:	The audit idles the rTEILA tuple and zeros the linkages involved.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTEILA.
<i>Bad Data =</i>	The first linkage field to TEDB.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rTEILA.
<i>Dump =</i>	rTEILA tuple.

TELEM

Software Release: 5E10 and later

1. TIMER ELEMENT (TELEM) AUDIT

The timer element audit audits all the timer elements and their associated linkage structures. These timer elements can be associated with a specified message transport (MT) data structure, currently, logical link control block (LLCB) or peripheral channel control block (PCCB). The MT general purpose timers are used as part of the implementation of link access protocol for D-channel (LAPD) protocol in the second message handler (MH) of the digital networking unit-synchronous optical network (SONET) (DNU-S).

The DNU-S is a new peripheral for the 5ESS[®] switch that provides a cost of goods (COGS) reduction and synchronous optical network (SONET) capabilities. The second message handler provides all of the MH resident operational and maintenance software for establishing and maintaining communication paths using the peripheral control and timing (PCT) link between the SMP of SM-2000 and the DNU-S. LAPD will be used to communicate to the DNU-S using the PCT links over three fixed timeslots as control to each common control (CC) of DNU-S.

The TELEM audit runs in segmented mode and is part of the second MH base level loop. This base level loop provides for loading and unloading FIFOs used for communicating to DNU-S. It also provides time for routine work such as mate (CC) update, hashsum checks for active and mate MHs and for audits.

2. DATA STRUCTURES

The TELEM audit checks the MT timer element array(MT telem[]) made up of MTTELEM data structure elements. It also checks the associated linked list array(MT link[]) made up of MTRLINKAGE data structure elements. There is a one to one linkage by index between the timer elements and its linkage structures. The MTTELEM data structure consists of:

- *stype* field representing structure type being timed
- *sidx* field representing structure index being timed
- *ttype* field representing the timer type
- *tcount* field representing the timeout countdown. This field is not audited.

The MTRLINKAGE data structure consists of:

- *next* field representing the index of next linked list element
- *prev* field representing the index of previous linked list element
- *owner* field representing the owning linked list

3. AUDIT CHECKS

The audit checks the MTthead[] array and prints audit errors if needed as follows:

BADIDLE	= Check for null values for the timer element on the free list.
BADLINK	= Check the stype and sidx fields.
HEAD	= Range check owner field.

ONE2ONE = Check the previous and next linkages.

OUTOFRANGE = Range check the ttype field.

ZAP = Check for system interrupt handling.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	The <i>owner</i> field indicates that this timer element is on the free list yet one of the timer element fields, <i>stype</i> , <i>sidx</i> , <i>ttype</i> or <i>tcount</i> , failed the null value check.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	Recover this timer element and put it on the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The stype field.
<i>Bad Data</i> =	The <i>sidx</i> field.
<i>Good Data</i> =	The <i>owner</i> field.
<i>Logical Index</i> =	Timer element index.
<i>Dump 1</i> =	The timer element under audit.
<i>Dump 2</i> =	The linkage data structure with which this timer element was attached to.

Table 2 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The <i>stype</i> field indicates which data structure the timer element is attached to, PCCB or LLCB and the <i>sidx</i> field indicates the index of this data structure. The TELEM audit failed these linkage checks for the timer element which is not on the free list. To determine which check failed, look at the good data field description below.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	Recover this timer element and put it on the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>sidx</i> field.
<i>Bad Data</i> =	The <i>stype</i> field.
<i>Good Data</i> =	This field dumps the value of local variable, <i>err_type</i> which tells which type of error caused this error. The numerical numbers below corresponds to the error type.
	(1) This error means the timer element was attached to PCCB and the <i>sidx</i> failed validity checks or the link to PCCB is not good, that is, when audit accessed the PCCB pointed to by <i>sidx</i> that PCCB is not pointing back to this timer element. (2) This error means the timer element was attached to LLCB and the <i>sidx</i> failed validity checks or the link to LLCB is not good, that is, when audit accessed the PCCB pointed to by <i>sidx</i> that LLCB is not pointing back to this timer element. (3) This error means the <i>stype</i> field indicates that this timer element is not on the free list and is not attached to either PCCB or LLCB. The

	timer element can attach to only these two structures when not on the free list.
<i>Logical Index</i> =	The timer head cell index.
<i>Dump</i> =	The timer element under audit.

Table 3 Error Code: HEAD

Error Code:	HEAD
Error Description:	The <i>owner</i> field of the linkage structure of this timer element failed the range check.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	Recover this timer element and put it on the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the linkage data structure of this timer element.
<i>Bad Data</i> =	The <i>owner</i> field.
<i>Good Data</i> =	The null value of the <i>owner</i> field, MTTMAXHDCS.
<i>Logical Index</i> =	Timer element index.
<i>Dump</i> =	The linkage data structure with which this timer element was attached to.

Table 4 Error Code: ONE2ONE

Error Code:	ONE2ONE
Error Description:	The audit prints this error when any of the <i>next</i> and <i>prev</i> field of the linkage structure attached to this timer element failed checks. These checks are point to and point back checks with the <i>next</i> and previous timer element in the linked list and range checks. To determine which check failed, look at the logical key field description below.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	Recover this timer element and put it on the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>next</i> field of the linkage data attached to this timer element.
<i>Bad Data</i> =	The <i>prev</i> field of the linkage data attached to this timer element.
<i>Good Data</i> =	The <i>owner</i> field of the linkage data attached to this timer element.
<i>Logical Index</i> =	This field is of size long and dumps two values. In the upper most two bytes(right) it dumps the error type and on the remaining two bytes it dumps the timer element index being audited. The algorithm used is, timer element << 16 error type. This error type value tells which type of error caused this error. The numerical numbers below corresponds to the error type. (1) The <i>next</i> timer element attached to this timer element either does not point back to this timer element or is linked to a different owner than this timer element. (2) The <i>next</i> field of the timer element is null, indicating this is the last element of the linked list but when audit checked the <i>tail</i> field of its owner, they don't match. (3) The <i>next</i> field of the timer element failed the range check. (4) The previous timer element attached to this timer element either does not point back to this timer element or it is linked to a different owner than this timer element. (5) The <i>prev</i> field of the timer element is null, indicating this is the first element of the linked list but when audit checked the <i>first</i> field of its owner, they don't match. (6) The <i>prev</i> field of the timer element failed the range check.

<i>Dump 1</i> =	The next timer element attached to this timer element.
<i>Dump 2</i> =	The previous timer element attached to this timer element.

Table 5 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The timer element's <i>ttype</i> field failed the range check.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.
Corrective Action Taken:	Null out the <i>ttype</i> field with its null value, MTT_NULL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The <i>owner</i> field.
<i>Bad Data</i> =	The <i>ttype</i> field.
<i>Good Data</i> =	The null value of the <i>ttype</i> field, MTT_NULL.
<i>Logical Index</i> =	Timer element index.
<i>Dump</i> =	The timer element under audit.

Table 6 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audits will take this error as the result of a write protection violation, out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Audits may have caused the interrupt so action must be taken not to roll in the interrupts.
Corrective Action Taken:	If the THC index is in range, an interface will be called to reconstruct this linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of SMmhrunning, telling which MH process is running.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	MTTMAXTE, maximum number of timer elements.
<i>Logical Index</i> =	The THC index when the error occurred.
<i>Dump</i> =	None.

32. TF - TK

TFD

Software Release: 5E10 and later

1. TRANSMIT FRAME DESCRIPTOR (TFD) AUDIT

Transmit frame descriptors (TFDs) are used by the port processor (PP) when the PP transmits frames. The PP seizes a TFD, places it into the channel high or low priority transmit queue, and notifies the direct memory access (DMA) processor. DMA processor removes the TFDs from transmit queue, copies the TFD information into a frame structure, passes them to sport, and returns the TFD to the TFD free list.

2. DATA STRUCTURES

PSS_TFD

3. AUDIT CHECKS

The TFD audit performs audits of the TFD free list, per channel TFD high and low priority lists, collects lost TFDs, and puts them back on the free list.

4. ERROR HANDLING

The error handling of this audit involves setting the head and tail pointers of free and channel TFD queues; checking the total number of TFDs per channel; handling merged and circular lists; and returning the lost TFDs to the free list.

The audit produces the following error code(s).

Table 1 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	The audit is auditing the TFD channel list. It found the "ntfd" field of the first TFD to be null and the tail offset does not point to this TFD.
Possible Error Effect:	This error will result in loss of frames on that channel as tail offset does not point properly to the tail of the list. This can result in interrupts from operational code.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to the first TFD in the channel list which should be equal to the head offset here.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the tail of this channel's TFD queue being audited (high or low).
<i>Bad Data =</i>	The value of tail offset of the channel list.
<i>Good Data =</i>	The address (offset) of the first TFD in the channel list. Here it will be equal to head offset.
<i>Logical Index =</i>	The number of the channel whose TFD list we are auditing.
<i>Dump =</i>	None.

Table 2 Error Code: BACKLINK1

Error Code:	BACKLINK1
Error Description:	The audit is auditing the TFD free list. It found the "ntfd" field of the first TFD to be null and the tail offset does not point to this TFD.
Possible Error Effect:	This error will result in running out of TFDs as the TFDs will be returned at the tail of the free list and the tail offset of the free list does not point to the end of the free list.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to the first TFD in the free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the last TFD in the free list.
<i>Bad Data =</i>	The value of tail offset of the free list.
<i>Good Data =</i>	The address (offset) of the first TFD in the free list. Here it will be equal to head

	offset.
<i>Logical Index</i> =	This value is "PSMAXCHN + 2" (130) indicating that audit was auditing the free list.
<i>Dump</i> =	None.

Table 3 Error Code: BACKLINK2

Error Code:	BACKLINK2
Error Description:	While going through the TFD list for the channel, audit found the owner field of a TFD other than the first TFD, not equal to the channel number that is, bad value of owner field. This means either some other channel TFD list has merged with this list or the free list has merged with this list or the operational code did not update the owner when it put this TFD on the channel list or there is wild write.
Possible Error Effect:	This error may result in interrupts from operational code.
Corrective Action Taken:	The audit will report the error and will set the tail offset to previous TFD, will update the "tfdcnt" field and set the "ntfd" field of the previous TFD to null.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the current TFD where audit found bad owner value.
<i>Bad Data</i> =	The value of "owner" field of the current TFD.
<i>Good Data</i> =	The channel number of the TFD list being audited as that should be the value of the owner field.
<i>Logical Index</i> =	The channel number whose TFD channel list is being audited.
<i>Dump</i> =	None.

Table 4 Error Code: BADOWNER

Error Code:	BADOWNER
Error Description:	The owner field of the first TFD in the TFD free list has bad value of owner (other than free list owner value). This is possible if any of the channel TFD queue has merged with free list or the value of owner field was not updated correctly when put back to free list or the value of owner field had wild write.
Possible Error Effect:	This error may result in a resource problem of TFD and that will be reported by operational code (PP). This can also result in loss of frames on a channel if channel queue has merged with the free list. Because the TFD will be taken off the channel list rather than the free list. Operational code may take interrupts for more than one reason.
Corrective Action Taken:	The audit will report the error and will set the head offset to point to tail offset or a valid TFD if tail offset is also bad. All the TFDs that the head was pointing to, would be put back to free list reporting them as "LOST" TFDs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the first TFD in the free list.
<i>Bad Data</i> =	The value of owner field of the first TFD in the free list.
<i>Good Data</i> =	PSMAXCHN, the owner value when TFD is on the free list.
<i>Logical Index</i> =	This value is "PSMAXCHN + 2" (130) indicating that audit was auditing the free list.
<i>Dump</i> =	None.

Table 5 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The tail offset of the TFD free list does not point to the TFD whose "ntfd" field has null value.
Possible Error Effect:	We may have a loss of resource problem as the TFDs are returned at the tail offset and tail offset does not point to the actual tail of the TFD free list.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to this TFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the variable which holds the value of the tail offset of TFD free list.
<i>Bad Data</i> =	The value of the tail offset of TFD free list.

<i>Good Data</i> =	The offset value of the address of the current TFD.
<i>Logical Index</i> =	This value is "PSMAXCHN + 2" (130) indicating that audit was auditing the free list.
<i>Dump</i> =	None.

Table 6 Error Code: BLINK

Error Code:	BLINK
Error Description:	The head offset of this channel's TFD queue has null value and audit found that tail offset does not have null value which it should have.
Possible Error Effect:	This may result in no frames being transferred on that channel as DMAP is putting TFDs at the end and PP thinks there are no TFDs to transfer on this channel. The operational code may take interrupts.
Corrective Action Taken:	The audit will report the error and will set the value of the tail offset to null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the tail for the TFD list (high or low) of this channel.
<i>Bad Data</i> =	The value of "rqt", tail offset, of this channel.
<i>Good Data</i> =	PSNULLOFF, the null value of the tail offset.
<i>Logical Index</i> =	The number of the channel whose tail offset has bad value.
<i>Dump</i> =	None.

Table 7 Error Code: BUSYCHAN

Error Code:	BUSYCHAN
Error Description:	While auditing the TFD free list, audit found the value of head and tail offset to be out of range and it did not find a single TFD whose owner field has a value of free list owner that is, it is attached to the free list. This condition is very rare and we should never see this error because finding this error means all the TFDs are currently in use and the value of head and tail offsets of the TFD free list have bad value.
Possible Error Effect:	This error represents an emergency situation as there are no more TFDs left on the TFD free list. The operational code will report various error reports and interrupts including resource error.
Corrective Action Taken:	The audit will report the error and will take the very first TFD no matter where it is by going through index and set the head and tail of the TFD free list to point to this TFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. NA.
<i>Bad Data</i> =	0. NA.
<i>Good Data</i> =	PSMAXCHN. The value of the TFD free list owner. This value is put here to say that no TFD of this owner was found which should not have happened.
<i>Logical Index</i> =	PSMAXCHN +2. Representing that TFD free list is being audited at that point.
<i>Dump</i> =	None.

Table 8 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The head offset of this channel's TFD queue has null value and audit found a non-zero value to the "tfdcnt" field which represents total number of TFDs currently on the TFD channel queue.
Possible Error Effect:	No significant effect other than if the value is not zero, it might get updated to a wrong value when TFDs are attached to that list which in turn may lead to false value of TFDs being allowed on that list.
Corrective Action Taken:	The audit will report the error and will set the value of the "tfdcnt" to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the count variable for the TFD queue (high or low) of this channel.
<i>Bad Data</i> =	The value of "tfdcnt", TFD count, of this channel.
<i>Good Data</i> =	0, no TFDs on this channel.
<i>Logical Index</i> =	The number of the channel whose "tfdcnt" had bad value.

<i>Dump</i> =	None.

Table 9 Error Code: HEAD

Error Code:	HEAD
Error Description:	The audit started auditing the TFD free list and the value of head offset (the pointer to the first TFD of the TFD free list) failed the range and boundary check. This means it is not pointing to an address which is a valid starting address of any of the TFDs.
Possible Error Effect:	This error will immediately result in a resource problem of TFD and will be reported by operational code (DMAP). If not corrected, this audit will be called in direct mode to correct the problem.
Corrective Action Taken:	The audit will report the error and will set the head offset to point to tail offset or a valid TFD if tail offset is also bad. All the TFDs that the head was pointing to, would be put back to free list reporting them as "LOST" TFDs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. The head offset which is an address in itself is bad and is dumped in Bad data field.
<i>Bad Data</i> =	The value of head offset.
<i>Good Data</i> =	0. The value chosen here is zero but any value which is a valid beginning address of the TFD is a good data.
<i>Logical Index</i> =	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump</i> =	None.

Table 10 Error Code: HEADBID

Error Code:	HEADBID
Error Description:	The head offset of this channel's TFD queue failed the range and boundary check. That is it is not pointing to an address which is the starting address of any of the TFDs.
Possible Error Effect:	This may result in no frames being processed at this channel as PP processor will always think that there is no TFD in the queue.
Corrective Action Taken:	The audit will report the error, will set the value of the head and tail offset to null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the channel control block for this channel.
<i>Bad Data</i> =	The value of "tqhd", head offset, of this channel.
<i>Good Data</i> =	PSNULLOFF, the null value of the head offset. Also any valid TFD offset can be a good data.
<i>Logical Index</i> =	The number of the channel whose head offset has bad value.
<i>Dump</i> =	None.

Table 11 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	While going through the TFD list for the channel, audit found the "ntfd" field of the current TFD it is auditing to be null and the tail offset of this TFD channel queue not to be pointing to this TFD, which it should be.
Possible Error Effect:	This error may result in interrupts from operational code. We may lose some frames as the tail offset has bad value and any TFDs attached to this frame will be lost.
Corrective Action Taken:	The audit will report the error and will set the tail offset to point to this TFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of this channel's TFD queue tail offset.
<i>Bad Data</i> =	The value of the tail offset.
<i>Good Data</i> =	The offset value of the current TFD audit is auditing.
<i>Logical Index</i> =	The channel number whose TFD channel list is being audited.
<i>Dump</i> =	None.

Table 12 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	While going through the TFD free list, the value of head pointer or the value of owner field of the first TFD in the free list has bad value. So when audit tried to check the tail offset value of the TFD free list, it was also found to have the bad value that is, its value did not pass the range and boundary checks.
Possible Error Effect:	This error may result in interrupts from operational code. It can also cause the resource error by the operational code.
Corrective Action Taken:	The audit will report the error and will set the head and tail offset to a valid TFD whose owner field says it is on the free list while going through TFD list by index.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. As we have found out that the value of tail offset is out of range anyway.
<i>Bad Data =</i>	The value of the tail offset.
<i>Good Data =</i>	0. The value is chosen zero but the valid value should be any starting address value of the TFD in the form of offset.
<i>Logical Index =</i>	PSMAXCHN+2. Representing that the TFD free list is being audited.
<i>Dump =</i>	None.

Table 13 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The audit has found a TFD whose "touch" field is not set that is not equal to 1 when it was going through all the TFDs by index again at the end of the audit. This means that neither audit nor the operational code came across this TFD during the current run of this audit and therefore it is "LOST". One of the possible causes of "LOST" TFD could be that it is linked to the list and the "ntfd" field of the previous TFDs was bad. So the audit updated the list making that or the previous TFD in the list and now all the TFDs following that TFD are "LOST". This can happen in free list or channel list.
Possible Error Effect:	This error has no immediate effect on the system but if it is not detected and if TFD is not put back to proper place (free list) it may result in resource problem of TFDs.
Corrective Action Taken:	The audit will report the error and will put this "LOST" TFD back to free list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the "LOST" TFD.
<i>Bad Data =</i>	The value of "owner" field of this TFD.
<i>Good Data =</i>	PSMAXCHN. The value of the owner field when on free list. It is assumed here that this TFD, should at least be on the free list.
<i>Logical Index =</i>	The TFD number which is found "LOST".
<i>Dump =</i>	The TFD which found to be lost.

Table 14 Error Code: MASTER_LINK

Error Code:	MASTER_LINK
Error Description:	The owner field of the first TFD in the TFD channel list has bad value of owner (other than channel number).
Possible Error Effect:	This can result in loss of frames on the channel or channels if the channel list has merged with this channel's list. It can also cause resource problem if TFD free list has merged with this channel's list as we may run out of TFDs or operational code may complain about those TFDs as certain fields would not have been set.
Corrective Action Taken:	The audit will report the error and will set the head and tail offset to null value and the "tfdcnt" to zero value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the first TFD in the channel list.
<i>Bad Data =</i>	The value of owner field of the first TFD in this channel's TFD list.
<i>Good Data =</i>	The channel number whose list is being audited as that is the owner value for all

	TFDs on that list.
<i>Logical Index</i> =	The channel number whose list is being audited.
<i>Dump</i> =	None.

Table 15 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	The audit found the "ntfd" field of the very first TFD in the TFD free list pointing to the head offset. Thus the TFD free list has become a circular list of one member.
Possible Error Effect:	This error may result in a resource problem of TFD if operational code cannot find more TFDs on the free list.
Corrective Action Taken:	The audit will report the error, will set the "ntfd" field to null value and set the tail offset to point to this TFD which will be equal to the head offset here.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the first TFD in the free list.
<i>Bad Data</i> =	The value of "ntfd" field of the first TFD on the free list.
<i>Good Data</i> =	PSNULLOFF. The value chosen here is null but any value which is a valid beginning address of the TFD is a good data.
<i>Logical Index</i> =	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump</i> =	None.

Table 16 Error Code: NULL4LINK

Error Code:	NULL4LINK
Error Description:	The audit found the "ntfd" field of the very first TFD in the TFD channel list pointing to the head offset. Thus the TFD channel list has become a circular list of one member.
Possible Error Effect:	This error may result in losing some of the frames on that channel and the operational code may take an interrupt.
Corrective Action Taken:	The audit will report the error, will set the head and tail offset to null value and the "tfdcnt" to zero value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the first TFD in the channel list.
<i>Bad Data</i> =	The value of "ntfd" field of the first TFD on the channel list.
<i>Good Data</i> =	PSNULLOFF. The value chosen here is null but any value which is a valid beginning address of the TFD is a good data.
<i>Logical Index</i> =	The value of the channel whose TFD list is being audited.
<i>Dump</i> =	None.

Table 17 Error Code: NULL5LINK

Error Code:	NULL5LINK
Error Description:	While going through the TFD list for the channel, audit found the running count of the total number of TFDs on this channel exceeded by the maximum value allowed. This means that either there are more number of TFDs on this channel than allowed or the queue has become a circular queue.
Possible Error Effect:	This error may result in interrupts from operational code.
Corrective Action Taken:	The audit will report the error and will set both the tail and head offset to null and TFD count field to zero.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. As we do not know where and which error occurred.
<i>Bad Data</i> =	0. Not Applicable.
<i>Good Data</i> =	The null value of "ntfd" field as we should have encountered that condition in any of the TFD before we reached at this point.
<i>Logical Index</i> =	The channel number whose TFD channel list is being audited.
<i>Dump</i> =	None.

Table 18 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	The audit has found a TFD whose "ntfd" field neither passed the range and boundary checks nor it is null when audit was traversing this TFD by index at the beginning of the audit.
Possible Error Effect:	This error could also have been detected by operational code resulting in an assert.
Corrective Action Taken:	The audit will report the error and will set the "ntfd" field to a null value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the TFD whose "ntfd" field was not good.
<i>Bad Data =</i>	The value of "ntfd" field of this TFD.
<i>Good Data =</i>	PSNULLOFF. The value chosen to dump here is null value but along with this null value, any value which satisfies the range and boundary check is a good value. By satisfying range and boundary checks means this TFD is pointing to a TFD address which is the starting address of any one of the total TFDs.
<i>Logical Index =</i>	The TFD number whose "ntfd" field is bad.
<i>Dump =</i>	None.

Table 19 Error Code: OWNER

Error Code:	OWNER
Error Description:	The owner field of the current TFD audit is visiting in the TFD free list has bad value of owner (other than free list owner value). This is possible if any of the channel TFD queue has merged with free list or the value of owner field was not updated correctly when put back to free list or the value of owner field had wild write.
Possible Error Effect:	This can result in loss of frames on a channel if channel queue has merged with the free list. Because the TFD will be taken off the channel list rather than the free list. The operational code may take interrupts.
Corrective Action Taken:	The audit will report the error and will set the "ntfd" field of the TFD to null and set the tail offset to point to this TFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the current TFD audit is visiting in the free list.
<i>Bad Data =</i>	The value of owner field of the TFD audit is visiting in the free list.
<i>Good Data =</i>	PSMAXCHN, the owner value when TFD is on the free list.
<i>Logical Index =</i>	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	None.

Table 20 Error Code: PQCNT

Error Code:	PQCNT
Error Description:	When audit found that the "ntfd" field of the first TFD of the TFD channel list has null value, the value of running count and the queue's actual count did not match, that is, the value of "tfdcncnt" was not one.
Possible Error Effect:	This error may result in an interrupt by operational code.
Corrective Action Taken:	The audit will report the error and will set the value of the "tfdcncnt" equal to one.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the channel control block of the channel whose TFD channel list is being audited.
<i>Bad Data =</i>	The value of "tfdcncnt" field of the channel control block for this channel.
<i>Good Data =</i>	rng_cnt, running count value which should be 1.
<i>Logical Index =</i>	The channel number whose TFD channel list is being audited.
<i>Dump =</i>	None.

Table 21 Error Code: R_COUNTS

Error Code:	R_COUNTS
Error Description:	The running count of the TFDs for the free list exceeded the total number of the TFDs per channel. Hence the TFD free list has become a circular list.

Possible Error Effect:	This can result in loss of resource of TFDs since all the TFDs are returned at the tail and the tail is not pointing where it is supposed to point (at the end).
Corrective Action Taken:	The audit will report the error, will set the "ntfd" field of the current TFD to null and set the tail offset to point to this TFD.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. We do not have any idea of where in the free list we found this to be the circular so we cannot dump the address of this TFD.
<i>Bad Data =</i>	0. We do not know which TFD's "ntfd" field is bad.
<i>Good Data =</i>	PSNULLOFF. null value of the "ntfd" field. This could have prevented the free list being a circular list.
<i>Logical Index =</i>	This value is "PSMAXCHN +2" (130) indicating that audit was auditing the free list.
<i>Dump =</i>	None.

Table 22 Error Code: RQCNT

Error Code:	RQCNT
Error Description:	While going through the TFD list for the channel, audit found the "ntfd" field of the current TFD it is auditing to be null and the value of TFD count does not match the value of the running count kept by the audit.
Possible Error Effect:	This error may result in interrupts from operational code. We may have more or less than allowable number of TFDs on the queue or the tfdcnt variable has a wild write.
Corrective Action Taken:	The audit will report the error and will set the TFD count equal to the running count kept by the audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of this channel's TFD queue's TFD count.
<i>Bad Data =</i>	The value of the TFD count.
<i>Good Data =</i>	The running count value kept by audit.
<i>Logical Index =</i>	The channel number whose TFD channel list is being audited.
<i>Dump =</i>	None.

Table 23 Error Code: ZAP

Error Code:	ZAP
Error Description:	The TFD audit entered in the ZAP mode. This means that when the processor (PP) took a non-returning error, audit control decided that TFD audit was running at the time.
Possible Error Effect:	The error is printed to notify that TFD audit was entered in the ZAP mode. It causes resets and asserts.
Corrective Action Taken:	The audit will decide from the value of AUcount, semi-global variable in au/AUups module where audit was working when it got zapped and accordingly it will take appropriate action.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA. Audit does not know the address where the processor took error.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	AUcount. The value of AUcount variable shows where audit was when it got zapped. The following is the list of possible values of AUcount and what is the meaning of it. 0-127 = The audit was auditing the TFD channel queue for channel equal to AUcount. Make sure that the write protection of this channel is turned off. 128 = The audit was in the middle of going from one task to other. Do nothing. 129 = The audit was zapped while going through the TFDs by its index value. Nothing to recover.

	130 = The audit was zapped while it was auditing the TFD free list. Nothing to recover. 131 = The audit was zapped while returning "lost" TFDs. Return them here. 132 and higher = This is an invalid value. Turn off write protection of all the TFD channel queues for precaution.
<i>Dump</i> =	None.

TGC

Software Release: 5E10 and later

1. TRUNK GROUP CONTROLS (TGC) AUDIT

Trunk group controls are a set of regulators used to protect an office from excessive routing requests and to expand routing patterns during busy periods. These regulators control the number of calls that can access a trunk group. A hierarchy exists within the trunk group controls. The group trunk group controls relation (RLGRP_TGC) maps active trunk groups in an office to trunk groups that have controls associated with them. The trunk group controls relation (RLTGC) maps the trunk groups and their associated controls to a specific control. The extended route relation (RLEXRR) maps a specific reroute control to a series of trunk groups designated to be the trunks on which the call attempts termination.

The trunk group controls audit (TGC) verifies that tuples in the trunk group controls relation (RLTGC) are correctly linked to tuples in the group trunk group controls relation (RLGRP_TGC) and the extended reroute relation (RLEXRR). The audit also checks for lost resources by verifying linkages to the database head cells relation (RLDB_HEADC).

This audit is part of the routing and terminal allocation audits subgroup. It operates in the OKP environment in the initialization and segmented modes.

2. DATA STRUCTURES

The data structures checked by the TGC audit are the trunk group controls relation (RLTGC), the group trunk group controls relation (RLGRP_TGC), the extended reroute relation (RLEXRR), and the database head cell relation (RLDB_HEADC). Figure 1 illustrates the layout of RLTGC and shows the items checked by the audit. The RLTGC relation is a dynamic relation that contains information about all trunk group controls. Included in the information within the tuples is the control type, the specific information about the control, the link to the extended reroute relation, and the group key, which is the point-back link to the RLGRP_TGC relation.

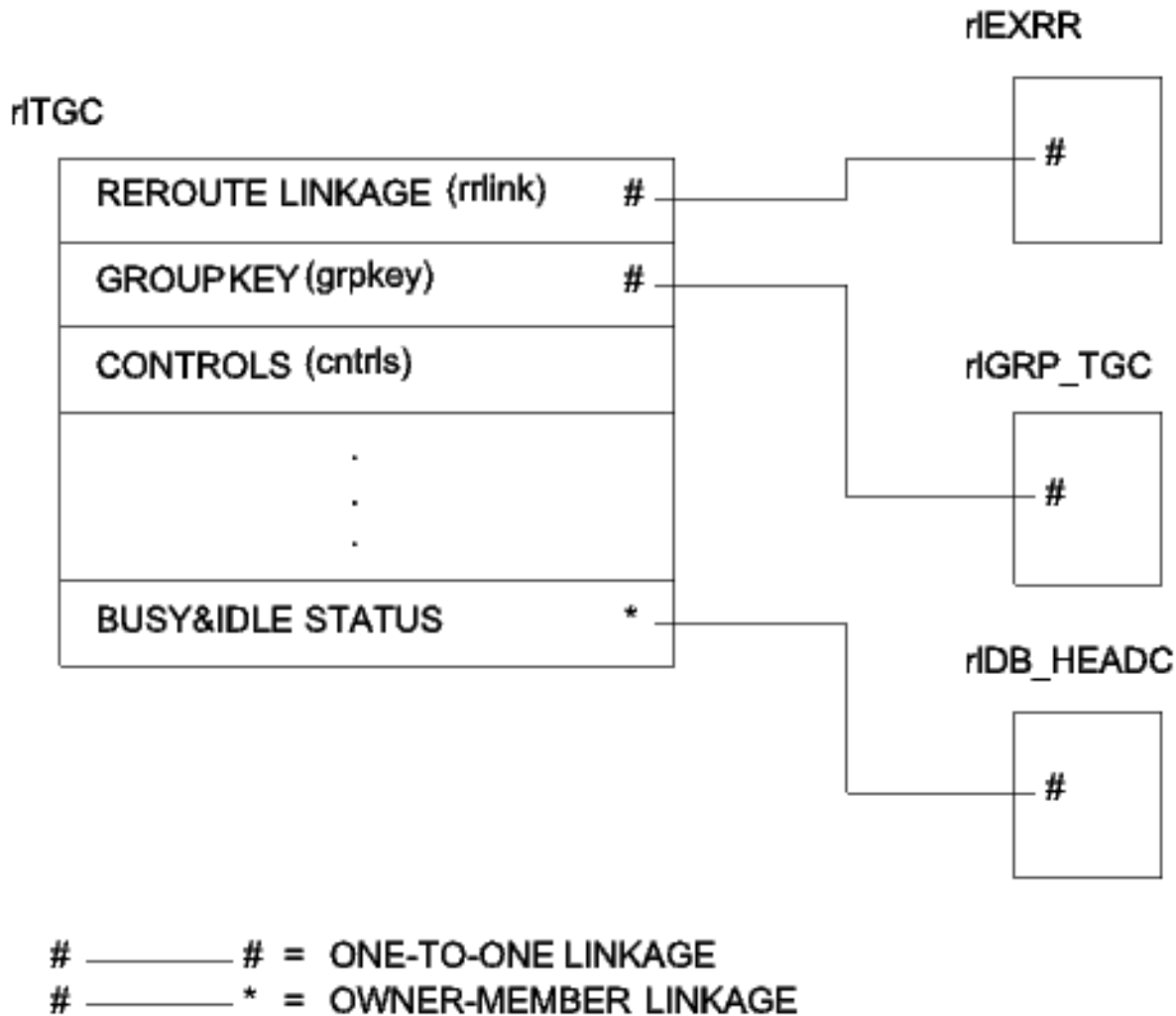


Figure 1 (TGC-1) rITGC Lineages

3. AUDIT CHECKS

To verify the linkages in the trunk group controls relation, the TGC audit calls the linkage driver (DRLINK) once for each audit segment. The DRLINK driver ensures that each tuple in rITGC has valid one-to-one linkages to the rGRP_TGC and rEXRR relations. The DRLINK driver also verifies the linkages to the rIDB_HEADC relation.

The DRLINK driver invokes the semantic check function after verifying the linkages of each RL TGC relation. The audit performs the following checks:

- (1) The audit verifies that all active rITGC tuples have a non-null link back to the RLGRP_TGC relation.
- (2) The audit verifies the consistency between the control field (cntrls) in rITGC and the control ID field in rGRP_TGC.
- (3) The audit verifies that the tuples in RL TGC are either busy or idle and that they are linked to the busy/idle lists in RLDB_HEADC.
- (4) The audit verifies the one-to-one linkage (rrlink) to the RLEXRR relation.

4. ERROR HANDLING

The audit reports errors when any linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples and links them to the appropriate idle list.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: NOTBUSY

Error Code:	NOTBUSY
Error Description:	A tuple with valid linkages and data to qualify as busy is not on the busy list.
Possible Error Effect:	If this tuple is on the idle list the data could be lost when the supposedly idle tuple is made active.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITGC tuple.
<i>Bad Data =</i>	Contents of st_owrk field.
<i>Good Data =</i>	Busy list head cell.
<i>Logical Index =</i>	Key to rITGC tuple.
<i>Dump =</i>	Contents of rITGC tuple.

Table 2 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	A tuple that has no link into rIGRP_TGC is not fully idled.
Possible Error Effect:	When the tuple is made active again some fields may be carried over.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITGC tuple.
<i>Bad Data =</i>	Contents of st_owrk field.
<i>Good Data =</i>	Idle list head cell.
<i>Logical Index =</i>	Key of rITGC tuple.
<i>Dump =</i>	Contents of rITGC tuple.

Table 3 Error Code: NULLCNTL

Error Code:	NULLCNTL
Error Description:	A tuple on the busy list has a non active control type.
Possible Error Effect:	Unknown.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITGC tuple.
<i>Bad Data =</i>	Contents of cntltyp field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of tuple.
<i>Dump =</i>	Contents of rITGC tuple.

Table 4 Error Code: RRHTYP

Error Code:	RRHTYP
Error Description:	The rr hunt type and the link into rLEXRR are inconsistent for a rr control. This can happen in one of three ways. + A spray or ordered hunt type has no link to rLEXRR. + A regular hunt type has a link to rLEXRR. + A rr tuple has an undefined hunt type.
Possible Error Effect:	The rr controls will not terminate properly.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of rITGC tuple.
<i>Bad Data</i> =	Contents of rrdata.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	Contents of rITGC tuple.

Table 5 Error Code: RRLINK

Error Code:	RRLINK
Error Description:	A tuple that is not a re_route control has a populated link into rEXRR.
Possible Error Effect:	This link will not be followed by network management but there is an rEXRR tuple that is being kept active and never used.
Corrective Action Taken:	The audit will reinitialize the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITGC tuple.
<i>Bad Data</i> =	Contents of rrlink field.
<i>Good Data</i> =	Zero.
<i>Logical Index</i> =	Key to rITGC tuple.
<i>Dump</i> =	Contents of rITGC tuple.

Table 6 Error Code: SEMANTIC

Error Code:	SEMANTIC
Error Description:	A tuple does not conform to the correct switching domain list semantics.
Possible Error Effect:	If the tuple is left alone, it may be improperly matched (or not-matched) to a call that it is not supposed to control (or that it is supposed to control). This matching is performed by NMprehunt() and NMposthunt().
Corrective Action Taken:	The audit will remove the tuple and place it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITGC tuple.
<i>Bad Data</i> =	Contents of sw_dom1, sw_dom2, sw_dom3, sw_dom4, sw_dom5, and include fields.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key to rITGC tuple.
<i>Dump</i> =	Contents of rITGC tuple.

THC

Software Release: 5E10 and later

1. TIMER HEAD CELL (THC) AUDIT

Timer head cells are the structures that a message transport (MT) general purpose timer can possibly attach to. This includes timing, timeout and free or idle head cells. The THC audit will audit the linkages of timer head cells with the timer structures. The MT general purpose timers are used as part of the implementation of link access protocol for D-channel (LAPD) protocol in the second message handler (MH) of the digital networking unit-synchronous optical network (SONET) (DNU-S).

The DNU-S is a new peripheral for the 5ESS[®] switch that provides a cost of goods (COGS) reduction and synchronous optical network (SONET) capabilities. The second message handler provides all of the MH resident operational and maintenance software for establishing and maintaining communication paths using the peripheral control and timing (PCT) link between the SMP of SM-2000 and the DNU-S. Link access protocol for D-channel (LAPD) will be used to communicate to the DNU-S using the PCT links over three fixed timeslots as control to each common control (CC) of DNU-S.

The THC audit runs in segmented mode and is part of the second MH base level loop. This base level loop provides for loading and unloading FIFOs used for communicating to DNU-S. It also provides time for routine work such as mate (CC) update, hashsum checks for active and mate MHs and for audits.

2. DATA STRUCTURES

The THC audit checks the MT timer head cell array(MTthead[]) made up of MTRHEADCELL data structure elements. The MTRHEADCELL data consists of:

- **first** field representing first resource on the linked list
- **last** field representing last resource on the linked list

3. AUDIT CHECKS

The audit checks the MTthead[] array and prints audit errors if needed as follows:

BAD_HEAD_TAIL = Check whether any checks with this head cell failed

ZAP = Check for system interrupt handling.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_HEAD_TAIL

Error Code:	BAD_HEAD_TAIL
Error Description:	The audit prints this error when any of the first and last field related checks for this timer head cell failed. These checks are null value, validity, owner linkages and linkage checks with the timer elements. To determine which check failed, look at the good data field description below.
Possible Error Effect:	Possible message failure between SM-2000 and DNU-S.

Corrective Action Taken:	Reconstruct the linked list of timer elements that should attach to this head cell.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The first field.
<i>Bad Data =</i>	The last field.
<i>Good Data =</i>	This field dumps the value of local variable, err_type which tells which type of error caused this error. The numerical numbers below corresponds to the error type.
	(1) This error means one of the fields of this head cell, either first or last, had null value, MTTNULLTE. For a valid linked list, either both are null or none is null. (2) This error means one of the fields of this head cell, either first or last, failed range check. (3) This error means one of the timer elements linked to this head cell and accessed by first and last, has its owner value not pointing to this head cell. Thus it failed the owner linkage check. (4) This error means either the prev field of the first timer element or next field of the last timer element of this head cell's linked list does not have a null value, MTTNULLTE. Thus it failed the linkage checks.
<i>Logical Index =</i>	The timer head cell index.
<i>Dump 1 =</i>	NA for error type 1 or 2 in <i>Good Data</i> . If the value of error type dumped in the good data field is either 3 or 4, then the linkage structure of type MTRLINKAGE of the first timer element attached to this head cell.
<i>Dump 2 =</i>	NA for error type 1 or 2 in <i>Good Data</i> . If the value of error type dumped in the good data field is either 3 or 4, then the linkage structure of type MTRLINKAGE of the last timer element attached to this head cell.

Table 2 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audits will take this error as the result of a write protection violation, out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Audits may have caused the interrupt so action must be taken not to roll in the interrupts.
Corrective Action Taken:	If the THC index is in range, an interface will be called to reconstruct this linked list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	the value of SMmhrunning, telling which MH process is running.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	MTTMAXHDCS, maximum number of timer head cells.
<i>Logical Index =</i>	The THC index when the error occurred.
<i>Dump 1 =</i>	None.

TIMER7

Software Release: 5E10 and later

1. CCS7 TIMER LIST HEAD CELL (TIMER7) AUDIT

The TIMER7 audit is responsible for verifying the linkage data and other semantic data which is stored in the RLTIMER7 relation. This audit runs in both the switching module (SM) and the administrative module (AM). Refer to Figure 1 for a pictorial view of the linkages related to the rTIMER7 tuple.

2. DATA STRUCTURES

TIMER7 is a dynamic relation which contains the CCS7 message consolidation and acknowledgement handler (MCAH) timer list head cells. Each timer list contains the MCAH table entries scheduled to expire at the time associated with the list (using a circular timer value to determine the list associated with the current time).

A special "IDLE" timer list is used to maintain the IDLE MCAH table entries (rENTRY7 tuples).

A special "RAR" timer list is used to maintain the reset acknowledge received (RAR) list of table entries (rENTRY7 tuples).

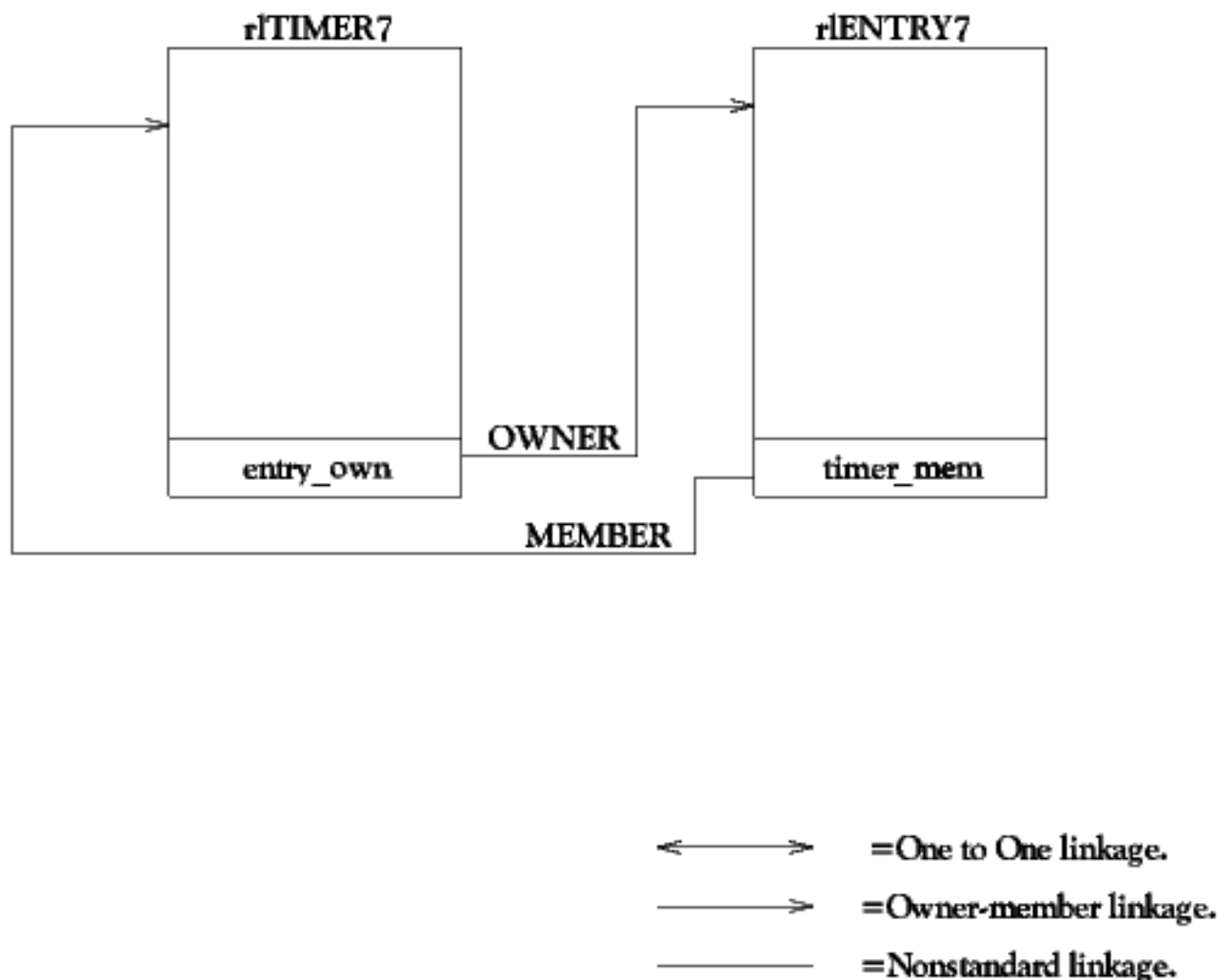


Figure 1 (TIMER7-1) RLTIMER7 Linkages

3. AUDIT CHECKS

The TIMER7 audit uses the DRINASEK audit driver to step through the rTIMER7 tuples and perform the following checks.

- COUNTS = Verify the *length* field value in the rTIMER7 tuple against the number of ENTRY7's linked to it.
- TWOLINKS = Verify that the linkages between the TIMER7 and ENTRY7 are correct.
- BDCOUNT = Verify that TMmc7icnt agrees with the value of *length* in the "IDLE" TIMER7.
- BAD_COUNTS = In the AM for the "RAR" TIMER7, verify that TMmc7acnt is 0 and the *length* is 0. Otherwise, in the SM for the "RAR" TIMER7, verify that TMmc7acnt agrees with the *length* field.
- EVEN_COUNT = Verify the TMmc7tcnt is at least the value of an individual TIMER7's *length* field.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	In the AM, for the "RAR" TIMER7 either TMmc7acnt is not zero or the <i>length</i> field in the rTIMER7 tuple is not zero. In the SM, for the "RAR" TIMER7 the value of TMmc7acnt does not agree with the <i>length</i> field in the rTIMER7 tuple.
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	In all cases, the TMmc7acnt value will be updated to either zero or the <i>length</i> in the "RAR" TIMER7. In the AM, the <i>length</i> field will be set to zero for the "RAR" TIMER7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rTIMER7 tuple.
<i>Bad Data</i> =	The value of TMmc7acnt.
<i>Good Data</i> =	Value of <i>length</i> in rTIMER7 tuple.
<i>Logical Index</i> =	The key of the rTIMER7 tuple (<i>timer</i>).
<i>Dump</i> =	Contents of the rTIMER7 tuple being audited.

Table 2 Error Code: BIDCOUNT

Error Code:	BIDCOUNT
Error Description:	For the "IDLE" TIMER7, the value of TMmc7icnt does not agree with the <i>length</i> field in this rTIMER7 tuple.
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	The TMmc7icnt value will be updated to the <i>length</i> in the "IDLE" TIMER7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rTIMER7 tuple.
<i>Bad Data</i> =	The value of TMmc7icnt.
<i>Good Data</i> =	Value of <i>length</i> in rTIMER7 tuple.
<i>Logical Index</i> =	The key of the rTIMER7 tuple (<i>timer</i>).
<i>Dump</i> =	Contents of the rTIMER7 tuple being audited.

Table 3 Error Code: COUNTS

Error Code:	COUNTS

Error Description:	The <i>length</i> field in TIMER7 was found to be inconsistent with regards to the number of ENTRY7's linked to the TIMER7.
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	The <i>length</i> field will be updated to reflect the correct number of ENTRY7's linked to this TIMER7.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rTIMER7 tuple.
<i>Bad Data</i> =	Value of <i>length</i> in rTIMER7 tuple.
<i>Good Data</i> =	Local count of ENTRY7's linked to this TIMER7.
<i>Logical Index</i> =	The key of the rTIMER7 tuple (<i>timer</i>).
<i>Dump</i> =	Contents of the rTIMER7 tuple being audited.

Table 4 Error Code: EVEN_COUNT

Error Code:	EVEN_COUNT
Error Description:	The value of TMmc7tcnt was found to be less than the value of the current rTIMER7 tuples <i>length</i> .
Possible Error Effect:	Incorrect resource allocation.
Corrective Action Taken:	The TMmc7tcnt value will be updated to the value of <i>length</i> in the rTIMER7 tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rTIMER7 tuple.
<i>Bad Data</i> =	The value of TMmc7tcnt.
<i>Good Data</i> =	Value of <i>length</i> in rTIMER7 tuple.
<i>Logical Index</i> =	The key of the rTIMER7 tuple (<i>timer</i>).
<i>Dump</i> =	Contents of the rTIMER7 tuple being audited.

Table 5 Error Code: TWOLINKS

Error Code:	TWOLINKS
Error Description:	The set member linkages between the TIMER7 and the associated ENTRY7 were found to be incorrect.
Possible Error Effect:	Errors in traversal of data.
Corrective Action Taken:	Related audits will run to clean up any other linkage problems which could exist.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the associated rTIMER7 tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The key of the rTIMER7 tuple (<i>timer</i>).
<i>Dump</i> =	Contents of the rTIMER7 tuple being audited.

TKOWNER

Software Release: 5E10 and later

1. TRUNK OWNER (TKOWNER) AUDIT

The trunk owner relation (RLTKOWNER) may reside in the communication module processor (CMP) for global trunk groups, or in a switching module (SM) for local trunk groups. In the CMP, the RLTKOWNER relation contains tuples for all global ports that exist in more than one SM. During normal operation, when this relation resides in an SM it contains only tuples for local port groups that exist in that SM. During stand-alone operation, the tuples for global ports are inserted into RLTKOWNER; when the SM leaves stand-alone operation, these tuples are deleted. The RLTKOWNER relation contains tuples that own lists of first-in, first-out (FIFO) and ROTATE hunted ports. The lists are arranged in groups of the same types of hunted ports (such as in service and out of service); together the lists form the trunk queue relation (RLTKQUE).

The trunk owner audit (TKOWNER) verifies that all trunks hunted by FIFO and ROTATE methods are represented in the RLTKOWNER relation in the appropriate module. It also verifies the owner-member linkages to the RLTKQUE relation. For an RLTKOWNER relation in the CMP, the audit also verifies its one-to-one linkage to the group trunk group controls relation (RLGRP_TGC).

This audit is part of the routing and terminal allocation audits subgroup. In the CMP, the audit operates in the OKP environment in the initialization and segmented modes; in an SM, the audit operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented mode.

2. DATA STRUCTURES

The data structures checked by the TKOWNER audit are the trunk owner relation (RLTKOWNER), the trunk queue relation (RLTKQUE), the group trunk group controls relation (RLGRP_TGC), and the trunk group information relation (RLRT_TRKG). Figure 1 shows the correspondence between the relations.

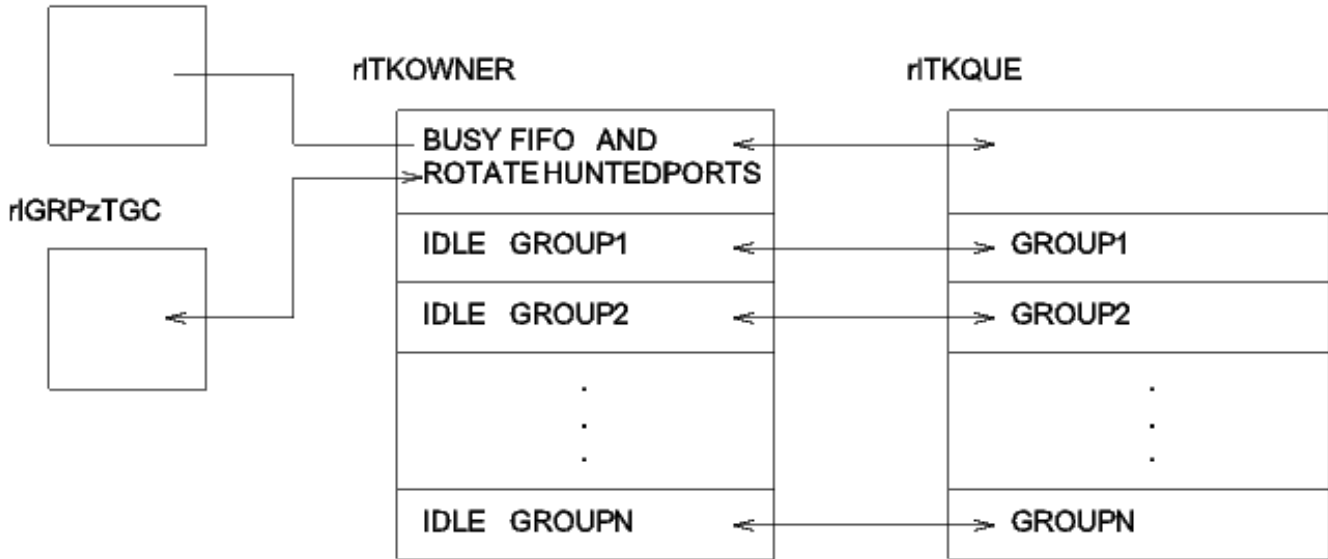
One tuple in RLTKOWNER owns the list of all busy FIFO and ROTATE hunted ports. When a hunted port changes from the idle state to the busy state, its linkage is changed and it is added to the list of busy FIFO and ROTATE hunted ports.

The other tuples in RLTKOWNER contain head and tail pointers to the groups (or lists) of the same type of idle FIFO and ROTATE hunted ports in RLTKQUE. Each RLTKOWNER tuple also contains a one-to-one link to the RLGRP_TGC relation for the trunk group controls feature.

SM & RSM*

AM

rIRT_TRKG



↔ = ONE-TO-ONE LINKAGE
 —→ = OWNER-MEMBER LINKAGE
 — = NONSTANDARD LINKAGE

* IN AN RSM ENVIRONMENT, rTKOWNER RESIDES IN BOTH THE AM AND RSM. BUT IN THE RSM, THE LINKAGE BETWEEN rTKOWNER AND RLGRP_TGC DOES NOT EXIST.

Figure 1 (TKOWNER-1) Trunk Owner Relation

3. AUDIT CHECKS

The TKOWNER audit calls the static driver (DRSTATIC) to step through the write-protected trunk group information relation (RLRT_TRKG). The audit verifies the linkages between RLTKOWNER and its member relation, RLTKQUE. The audit also verifies the one-to-one linkage between the RLTKOWNER and RLGRP_TGC relations in the CMP.

For each trunk group that the DRSTATIC driver retrieves, the TKOWNER audit checks the corresponding hunt type for that group. If the hunt type is FIFO or ROTATE and the group exists in the appropriate module, a corresponding rTKOWNER tuple should exist. The audit looks for extra tuples, lost tuples, and wrong processor assignments.

After all linkages have been checked, the DRSTATIC driver executes a semantic check function for each tuple. The semantic check function ensures that announcement-type trunks do not have trunk group controls associated with them. The semantic check function is not executed for the busy-list head cell tuple because it is not found in the static relation.

If there is an analog console busy lamp associated with a trunk group, the audit compares the number of idle

numbers with the automatic count kept by call processing. Unlike TKSTAT and LGTKSTAT, TKOWNER does this by counting the number of idle rITKQUE tuples connected to its idle list head call. In addition to the idle count (idle number of trunks), it checks that the status of the make-busy key and busy lamp is consistent. If the make-busy key is pressed, the lamp should be on regardless of the idle count.

If there is an ISDN attendant console busy lamp associated with a trunk group, the audit checks that the busy lamp and make-busy key of this console are consistent with the idle count (idle number of trunks). If the ISDN attendant console is equipped with a warning lamp, this lamp's state is checked against the idle count and the preset value in the trunk group information relation rIRT_TRKG. If the make-busy key is pressed, the busy and warning lamps should be on regardless of the idle-count.

4. ERROR HANDLING

This audit reports errors when any linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples, and the tuples are linked to the appropriate idle list.

Tuples with FIFO or ROTATE hunt status that should exist but do not are created in the rITKOWNER relation. Any tuples in rITKOWNER that have a hunt type other than FIFO or ROTATE or are not in the appropriate module are deleted.

The audit corrects count and lamp errors.

The audit corrects ISDN attendant console busy lamp and warning lamp inconsistencies if a trunk group is equipped with such lamps.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: ANNCMNT

Error Code:	ANNCMNT
Error Description:	An announcement trunk has a trunk group control associated with it.
Possible Error Effect:	From zero to one hundred percent of the calls attempting to terminate to the trunk will not be able to terminate. The percentage depends on the percent implemented for the associated control.
Corrective Action Taken:	The tuple is completely zeroed and the tuple's logical key initialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITKOWNER tuple.
<i>Bad Data =</i>	Link into rIGRP_TGC.
<i>Good Data =</i>	Null link.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	rITKOWNER tuple.

Table 2 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	In the process of calculating the idle count (number of idle trunk members in the group as expressed by the number of rITKQUE members tuples linked to this rITKOWNER tuple idle list head cell), the audit was unable to access the rITKQUE member tuples nor rITKOWNER tuple owner list head cell. Accessing this list is necessary in order to properly audit the group's idle count, console busy lamp, and console warning lamp.
Possible Error Effect:	The idle count, console busy lamp and console warning lamp data (rITKOWNER.idle_count, rITKOWNER.tgbl_on, and rITKOWNER.tgw_on) cannot be audited for correctness. Should these fields become corrupted, the customer would not receive proper trunk group status from the status lamps on

	the console monitoring this group. If the error occurred because the rITKOWNER tuple, itself, has been corrupted, trunk routing failures may occur.
Corrective Action Taken:	The rITKOWNER dynamic tuple was recovered to an idle state, or is deleted if it is an extra tuple.
Manual Action Required:	If this error continues to fire, it indicates an ODD inconsistency exists. Using ODBE, examine the corresponding rIRT_TRKG static tuple for this group (where the key rIRT_TRKG.grpnum = rITKOWNER.grpnum). The key for the rIRT_TRKG static tuple is dumped as the logical key when this audit error fires. Once the rIRT_TRKG static tuple has been obtained, examine the hunt type (rIRT_TRKG.htype). It should reflect one of the legal FIFO hunting algorithm hunt types. The legal FIFO hunt types are listed in the RLRT_TRKG relation poprules, along with the rules for their use. After carefully reviewing the poprules for RLRT_TRKG for the use of FIFO hunt types, select a NON-PACKET hunt type from the list of available FIFO hunt types. Using ODBE, correct the hunt type in RLRT_TRKG to match a FIFO hunt type selected from the poprules.
Dump Description:	
<i>Error Address =</i>	The rITKOWNER tuple address.
<i>Bad Data =</i>	0 if an ODD inconsistency exists. Negative if the rITKOWNER tuple has become corrupted or a database error has occurred, in which case the specific error received from DB is dumped here.
<i>Good Data =</i>	The trunk group number (key) that was found in the corresponding rIRT_TRKG static tuple (rIRT_TRKG.grpnum).
<i>Logical Index =</i>	Group number (key) found in the rITKOWNER dynamic tuple (rITKOWNER.grpnum).
<i>Dump =</i>	The contents of the rITKOWNER dynamic tuple.

Table 3 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	A count of the idle group members did not equal the stored idle count.
Possible Error Effect:	Measurements would be in error for this group. Possible mishandling of the group busy and warning lamps -- if equipped.
Corrective Action Taken:	The idle count was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rITKOWNER tuple address.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Correct count of idle members.
<i>Logical Index =</i>	Group number.
<i>Dump =</i>	rITKOWNER tuple.

Table 4 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The hunt type for this group is of the wrong type and therefore should not have a tuple in the RLTKOWNER relation.
Possible Error Effect:	If the group's hunt type is changed to RTH_FIFO, RTH_ROTATE, RTH_PRFIFO, RTH_PSUCD, or RTH_PLFIFO, an assert will fire.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITKOWNER tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Hunt type.
<i>Logical Index =</i>	grpnum of this tuple.
<i>Dump =</i>	NA.

Table 5 Error Code: LAMP

Error Code:	LAMP
Error Description:	This trunk group's analog and/or ISDN attendant console group busy lamp was found to be in the wrong state.

Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITKOWNER tuple.
<i>Bad Data</i> =	Lamp state as found (0 = OFF; 1 = ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	rITKOWNER tuple.

Table 6 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The hunt type for this group is FIFO or ROTATE and therefore should have a tuple in the RLTKOWNER relation.
Possible Error Effect:	Inability to hunt members in the group or possible assert failures.
Corrective Action Taken:	The tuple is created.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Hunt type.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	NA.

Table 7 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the CMP, only tuples from global hunt groups should be present. In an SM, only tuples from local hunt groups on that SM should be present.
Possible Error Effect:	More than one processor could route to this group.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITKOWNER tuple.
<i>Bad Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	NA.

Table 8 Error Code: TLAMP

Error Code:	TLAMP
Error Description:	This trunk group's ISDN attendant console threshold warning lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of RLTKOWNER tuple.
<i>Bad Data</i> =	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	RLTKOWNER tuple.

TKQUE

Software Release: 5E10 and later

1. TRUNK QUEUE (TKQUE) AUDIT

The trunk queue relation (RLTKQUE) resides in the communications module processor (CMP) for global trunk groups, and in a switching module (SM) for local groups. In the CMP, the RLTKQUE relation contains tuples for all global ports that exist in more than one SM. During normal operation, when this relation resides in an SM it contains only tuples for local port groups that exist in that SM. During stand-alone operation, the tuples for global ports are inserted into RLTKQUE, but when the SM leaves stand-alone operation, these tuples are deleted. The RLTKQUE relation contains one tuple for each first-in, first-out (FIFO) and ROTATE hunted port. The ports are arranged in groups by type of port; the structure of the groups is maintained by the linkages among the tuples. The trunk queue audit (TKQUE) verifies the existence of each tuple and the linkages among them.

This audit is part of the routing and terminal allocation audits subgroup.

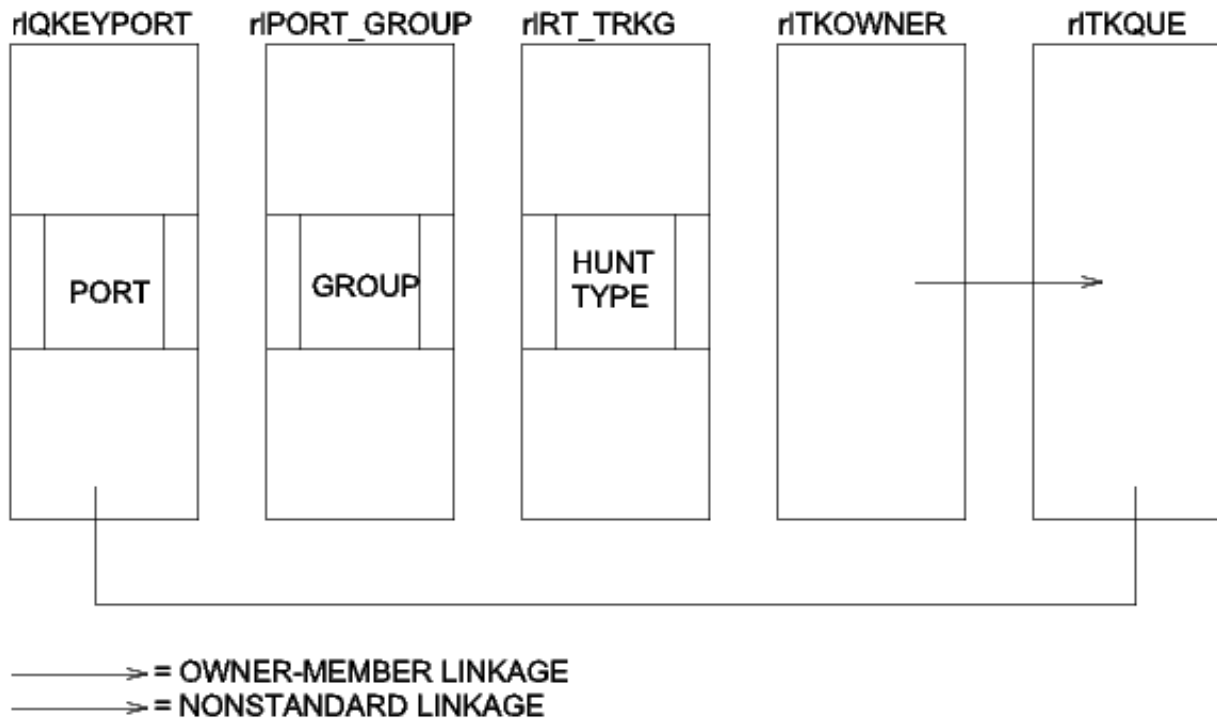
2. DATA STRUCTURES

The data structures used by the TKQUE audit are the queue key port relation (RLQKEYPORT), the port group relation (RLPORT_GROUP), the trunk group information relation (RLRT_TRKG), and the trunk queue relation (RLTKQUE). The RLQKEYPORT, RLPORT_GROUP, and RLRT_TRKG relations are write-protected and subject only to recent changes. They reflect the correct current state of the tuples and therefore dictate the state of the RLTKQUE tuples.

The trunk queue audit also verifies the linkage between the RLTKQUE relation, which is a member relation, and the trunk owner relation (RLTKOWNER). Figure 1 shows the correspondence between the relations.

SM & RSM*

AM



* IN AN RSM ENVIRONMENT, THE rITKQUE RELATION RESIDES IN BOTH THE AM AND RSM.

Figure 1 (TKQUE-1) Trunk Queue Audit

3. AUDIT CHECKS

The TKQUE audit uses the static driver (DRSTATIC) to step through the write-protected RLQKEYPORT relation, which looks at two other write-protected relations (RLPORT_GROUP and RLRT_TRKG) to determine which port should have corresponding tuples in the trunk queue relation (RLTKQUE). If a port is found in RLQKEYPORT that belongs in RLTKQUE but is not there, the audit creates a tuple for that port. The newly created tuple will have bad linkages; the linkage is zeroed, and the tuple is placed on the appropriate list of idle ports.

The TKQUE audit also ensures that all ports in RLTKQUE have the proper hunt type. (This is considered a syntactic check because the hunt type field may be considered part of the logical linkage.) Tuples that have wrong hunt types are removed from the RLTKQUE relation. The audit also ensures that the tuples exist for the appropriate processor (CMP or SM).

The audit finds tuples in the trunk queue relation that have no predecessors or no successors, which means they are either the first or last members of a list. The audit verifies that these tuples are linked to the trunk owner relation (RLTKOWNER). If the list is properly linked, a key in RLTKQUE points back to the tuple in RLTKOWNER. If the key in RLTKQUE does not point back to RLTKOWNER, the tuple in RLTKQUE is in error, and the audit links the erroneous tuple to the appropriate idle list.

The TKQUE audit calls a semantic check function that performs the following checks:

- (1) The audit verifies that the global port name (GPORT) of the rITKQUE tuple matches the GPORT in the

respective rIQKEYPORT tuple.

- (2) The audit verifies that the rITKQUE tuple is linked to the proper owner in the trunk owner relation (RLTKOWNER).
- (3) The audit looks for extra tuples and lost tuples.
- (4) The audit verifies that rIRT_TRKG tuple has a tuple for each port group indicated by the RLPORT_GROUP relation.
- (5) The audit checks the sport structure of RLTKQUE against that of RLQKEYPORT.
- (6) The audit ensures that each rITKQUE tuple is in the right processor.
- (7) The audit ensures that the **fast_inh** field in the rITKQUE tuple is the same as the value defined in the rIQKEYPORT tuple. This field is the fast inhibit bit that is used in digital networking unit - SONET (DNU_S) facilities.

4. ERROR HANDLING

The audit reports errors when any owner-member linkage check or semantic check fails. The audit zeroes the linkage areas of the erroneous tuples and links them to the appropriate idle list. Excess tuples and trunk queue tuples with incorrect hunt types and incorrect processor numbers are deleted.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The hunt type for the group of which this port is a member is not FIFO or ROTATE and therefore should not have a tuple in the RLTKQUE relation.
Possible Error Effect:	When a port with this qkey is assigned as a member of a FIFO or ROTATE hunt group, an assert will fire.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITKQUE tuple.
<i>Bad Data =</i>	Global port member.
<i>Good Data =</i>	Hunt type.
<i>Logical Index =</i>	qkey of this tuple.
<i>Dump =</i>	NA.

Table 2 Error Code: GPORT

Error Code:	GPORT
Error Description:	The "gport structure" of the rITKQUE tuple does not match the "gport structure" in the rIQKEYPORT tuple.
Possible Error Effect:	Inability to hunt this port and possible assert failures.
Corrective Action Taken:	The rITKQUE gport structure is set equal to the rIQKEYPORT gport structure.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the gport structure.
<i>Bad Data =</i>	gport structure of the rITKQUE tuple.
<i>Good Data =</i>	gport structure of the rIQKEYPORT tuple.
<i>Logical Index =</i>	qkey of the trunk group.
<i>Dump =</i>	rITKQUE tuple.

Table 3 Error Code: INHSTATE

Error Code:	INHSTATE
Error Description:	The fast_inh fast inhibit bit in the rITKQUE tuple does not equal the value in the static rIQKEYPORT tuple.
Possible Error Effect:	Inability to inhibit DNU-S facilities.
Corrective Action Taken:	The rITKQUE fast_inh bit is set equal to the value contained in the rIQKEYPORT tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITKQUE tuple.
<i>Bad Data =</i>	The value of fast_inh before correction.
<i>Good Data =</i>	The value of fast_inh in the rIQKEYPORT tuple.
<i>Logical Index =</i>	qkey of the trunk group.
<i>Dump 1 =</i>	rITKQUE tuple.
<i>Dump 2 =</i>	rIQKEYPORT tuple.

Table 4 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	The hunt type for the group of which this port is a member is FIFO or ROTATE and therefore should have a tuple in the RLTKQUE relation.
Possible Error Effect:	Inability to hunt this port and possible assert failures.
Corrective Action Taken:	The tuple is created.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	Hunt type.
<i>Logical Index =</i>	qkey of this tuple.
<i>Dump =</i>	NA.

Table 5 Error Code: OWNLINK

Error Code:	OWNLINK
Error Description:	The rITKQUE tuple was found to have the wrong owner rITKOWNER.
Possible Error Effect:	Inability to hunt to this port and possible assert failures.
Corrective Action Taken:	The rITKQUE tuple is deleted if it exists.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITKQUE tuple.
<i>Bad Data =</i>	Owner link of rITKQUE tuple.
<i>Good Data =</i>	Group number of the port group.
<i>Logical Index =</i>	qkey of the rITKQUE tuple.
<i>Dump =</i>	rITKQUE tuple.

Table 6 Error Code: PCRID

Error Code:	PCRID
Error Description:	The rITKQUE tuple should not exist in this processor. In the CMP, only tuples from global groups should be present. In an SM, only tuples from local groups on the SM should be present.
Possible Error Effect:	More than one processor could attempt to route to this port.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITKQUE tuple.
<i>Bad Data =</i>	Module number if the port belongs to a local group or 0 if it belongs to a global group.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	qkey of this tuple.
<i>Dump =</i>	NA.

Table 7 Error Code: PORT

Error Code:	PORT
Error Description:	Valid values are: (1) The port indicated by the RLQKEYPORT relation does not have a corresponding tuple in the RLPORT_GROUP relation or the RLSPRT_GP relation for packet switching;
Possible Error Effect:	Failure for call processing to hunt this port and possible ASSERT failures.
Corrective Action Taken:	The rITKQUE tuple is deleted if one exists.
Manual Action Required:	Determine the nature of the database inconsistency and correct it as follows: - For non-packet ports, insert the corresponding rLPORT_GROUP static tuple for the rLQKEYPORT tuple. - For packet ports, insert the corresponding rLSPRT_GP static tuple for the rLQKEYPORT tuple, OR - If the rLQKEYPORT tuple is populated illegally (for example, the key dumped is 0), delete the rLQKEYPORT static tuple. This message will repeat until corrective action is taken by the user.
Dump Description:	
<i>Error Address =</i>	If this is a packet port then the module number, otherwise 0, for non-packet ports.
<i>Bad Data =</i>	If this is a packet port then the PH number, otherwise 0, for non-packet ports.
<i>Good Data =</i>	If this is a packet port then the channel number.
<i>Logical Index =</i>	Global port name.
<i>Logical Index =</i>	The key of the rLQKEYPORT static tuple for which there should have been a corresponding rLGROUP_PORT or rLSPRT_GP static tuple. If the key is 0, then it is the rLQKEYPORT static tuple which is invalidly populated, and this tuple should be deleted.
<i>Dump =</i>	rLQKEYPORT tuple as found.

Table 8 Error Code: TRKG

Error Code:	TRKG
Error Description:	The port group indicated by the RLPORT_GROUP or RLSPRT_GP relation does not have a corresponding tuple in the RLRT_TRKG relation. This error will repeat until it is fixed manually.
Possible Error Effect:	Inability to hunt this port and possible assert failures.
Corrective Action Taken:	The rITKQUE tuple is deleted if it exists.
Manual Action Required:	Determine the nature of the data base inconsistency and correct it. This message will repeat until corrective action is taken by maintenance personnel.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	If this is a packet port, 1. Otherwise, 0.
<i>Good Data =</i>	If this is a packet port, group number. Otherwise, NA.
<i>Logical Index =</i>	Group number of the trunk group.
<i>Dump 1 =</i>	rLQKEYPORT tuple.
<i>Dump 2 =</i>	rLPORT_GROUP tuple.
<i>Dump 3 =</i>	rLSPRT_GROUP tuple.

TKSTAT

Software Release: 5E10 and later

1. TRUNK STATUS (TKSTAT) AUDIT

To process a call, the routing and terminal allocation (RTA) routing process in the communications module processor (CMP) and in a switching module (SM) must determine which ports are available for terminating calls. Information about the type of hunted ports is kept on bit maps. A bit map is included in each tuple in the trunk status relation (RLTKSTAT), which contains information about small and average trunk groups (up to 96 trunks per group). The RLTKSTAT relation resides in the CMP, but it may also reside in an SM. In the CMP, the RLTKSTAT relation contains tuples for all global ports that exist in more than one SM. During normal operation, when this relation resides in an SM it contains only tuples for local port groups that exist in that SM. During stand-alone operation, the tuples for global ports are inserted into RLTKSTAT; when the SM leaves stand-alone operation, these tuples are deleted.

The trunk status audit (TKSTAT) checks the dynamic relation RLTKSTAT for extra tuples. For information about larger trunk groups, refer to the large trunk status audit (LGTKSTAT). For information about groups of hunted lines, refer to the line status audit (LNSTAT) and the large line status audit (LGLNSTAT).

This audit is part of the routing and terminal allocation audits subgroup.

2. DATA STRUCTURES

The data structures checked by the TKSTAT audit are the trunk status relation (RLTKSTAT) and the trunk group information relation (RLRT_TRKG). Figure 1 illustrates the two relations.

The RLRT_TRKG relation is a static (write-protected) relation that contains information about all hunted trunks. This information includes a key (group number), the type of trunk group (two-way or incoming, and so forth), the module (CMP or SM) that controls it, and the hunt type. The TKSTAT audit steps through the RLTKSTAT relation and verifies it against the information contained in the RLRT_TRKG relation. The RLRT_TRKG relation is also used in the large trunk status audit (LGTKSTAT), the trunk group information audit (RTTRKG), the trunk queue audit (TKQUE), and the trunk owner audit (TKOWNER).

SM & RSM*

AM

TKSTAT AUDIT

rIRTzTRKG

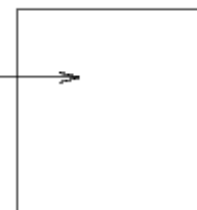
1	LG	AM		
2	T	AM		
3	T	AM		
4	LG	RSM 2		
⋮				
87	LG	AM		
88	T	AM		



rTKSTAT

2				
3				
⋮				
88				

rGRP_TGC



T = TRUNK

LG = LARGE TRUNK

= ONE-TO-ONE LINKAGE

*IN AN RSM ENVIRONMENT, THE rTKSTAT RELATION RESIDES IN BOTH THE AM AND RSM. BUT IN THE RSM, THE LINKAGE BETWEEN rTKSTAT AND rGRP_TGC DOES NOT EXIST.

Figure 1 (TKSTAT-1) Trunk Status Audit

3. AUDIT CHECKS

The TKSTAT audit calls the static driver (DRSTATIC) to check the correspondence between the rTKSTAT and rIRT_TRKG relations and then performs the following checks:

- The audit verifies each rTKSTAT tuple against the corresponding rIRT_TRKG tuple by using the key (group number) of the rTKSTAT tuple to read RLRT_TRKG relation. The audit looks specifically for extra rTKSTAT tuples.
- The audit checks the group type (RTTWOWAY) and hunt type (RTH_BWDLIN, RTH_FWDLIN, RTH_UCD, RTH_GUCD, RTH_BGUCD, RTH_PRBLIN, or RTH_PRFLIN) of the rIRT_TRKG tuple.

- (c) In the CMP, the audit also checks the link to the group trunk group controls relation (RLGRP_TGC).
- (d) The audit ensures that the module number is correct for the tuple.
- (e) If there is an analog console busy lamp associated with a trunk group, the audit compares its idle bit count with the automatic count made by call processing. In addition to the idle count (idle number of trunks) it checks that the status of the make-busy key and the busy lamp are consistent. If the make-busy key is pressed, the lamp should be on regardless of the idle count.

If there is an ISDN attendant console BUSY lamp associated with a trunk group, the audit checks that the busy lamp and make-busy key of this console are consistent with the idle count (idle number of trunks). If the ISDN attendant console is equipped with a warning lamp, this lamp's state is checked against the idle count and the preset value in the trunk group information relation rIRT_TRKG. If the make-busy key is pressed, the busy and warning lamps should be on regardless of the idle-count.

4. ERROR HANDLING

Excess tuples, tuples not in the appropriate module, and tuples with wrong hunt types are deleted from the relation. If a linkage check fails, the audit zeroes the tuple and the tuple is reconstructed. The audit corrects count and lamp errors.

The audit corrects ISDN attendant console busy lamp and warning lamp inconsistencies if a trunk group is equipped with such lamps.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADIDLE

Error Code:	BADIDLE
Error Description:	In the process of calculating the idle count (number of idle trunk members in the group as expressed by the number of bit positions in rTKSTAT.gmflg[] which are marked as "1"), the global trunk group tuple buffer AUtrkgbuf has been found to contain a corrupted hunt type for this group (AUtrkgbuf.htype). This tuple buffer is used to store a local copy of the corresponding rIRT_TRKG static tuple associated with the rTKSTAT dynamic tuple under audit (where rIRT_TRKG.grpnum = rTKSTAT.grpnum).
Possible Error Effect:	The idle count, console busy lamp and console warning lamp data (rTKSTAT.idle_count, rTKSTAT.tgbl_on, and rTKSTAT.tgwl_on) cannot be audited for correctness. Should these fields become corrupted, the customer would not receive proper trunk group status from the status lamps on the console monitoring this group.
Corrective Action Taken:	Audit processing is aborted for this group, the RTTRKG audit is scheduled, and in the process, the contents of the AUtrkgbuf global buffer are reinitialized.
Manual Action Required:	If this error continues to fire, it indicates that an ODD inconsistency exists. Using ODBE, examine the corresponding rIRT_TRKG static tuple for this group (where the key rIRT_TRKG.grpnum = rTKSTAT.grpnum). The key for the rIRT_TRKG static tuple is dumped as the logical key when this audit error fires. Once the rIRT_TRKG static tuple has been obtained, examine the hunt type (rIRT_TRKG.htype). It should reflect one of the legal LINEAR hunting algorithm hunt types. The legal LINEAR hunt types are listed in the RLRT_TRKG relation poprules, and are identified as RTH_XXXLIN or RTH_XXXUCD, where XXX denotes the particular hunting method (FWD for forward, BWD for backward,

	and so forth) Using ODBE, correct the hunt type in RLRT_TRKG to match one of the LINEAR hunt types listed in the poprules.
Dump Description:	
<i>Error Address</i> =	The rITKSTAT tuple address.
<i>Bad Data</i> =	The hunt type that was found in the global trunk group tuple buffer (AUtrkgbuf.htype).
<i>Good Data</i> =	The trunk group number (key) that was found in the global trunk group tuple buffer (AUtrkgbuf.grpnum) for which there should exist a corresponding rIRT TRKG static tuple, and rITKSTAT dynamic tuple.
<i>Logical Index</i> =	Group number (key) found in the rITKSTAT dynamic tuple (rITKSTAT.grpnum).
<i>Dump</i> =	The contents of the rITKSTAT dynamic tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	A count of the idle group members did not equal the stored idle count.
Possible Error Effect:	Measurements would be in error for this group. Possible mishandling of the group busy and warning lamps, if equipped.
Corrective Action Taken:	The idle count was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	RLTKSTAT tuple address.
<i>Bad Data</i> =	idle_count field.
<i>Good Data</i> =	Correct count of idle members.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	rITKSTAT tuple.

Table 3 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The group number for this tuple is not assigned.
Possible Error Effect:	An assert will fire if this tuple is not found after trying to retrieve it.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Calculated grpnum of this tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Calculated grpnum of this tuple.
<i>Dump</i> =	rITKSTAT tuple.

Table 4 Error Code: LAMP

Error Code:	LAMP
Error Description:	This trunk group's analog and/or ISDN attendant console group busy lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITKSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 = OFF; 1 = ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	rITKSTAT tuple.

Table 5 Error Code: PCRID

Error Code:	PCRID
Error Description:	The tuple should not exist in this processor. In the CMP, only tuples from global trunk groups should be present. In an SM, only tuples from local trunk groups on that SM should be present except during stand-alone operation when tuples from global trunk groups are allowed.
Possible Error Effect:	More than one processor could route calls to this group.
Corrective Action Taken:	The tuple is deleted.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITKSTAT tuple.
<i>Bad Data</i> =	Module number for local group; 0 if global.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rITKSTAT tuple.

Table 6 Error Code: TLAMP

Error Code:	TLAMP
Error Description:	This trunk group's ISDN attendant console threshold warning lamp was found to be in the wrong state.
Possible Error Effect:	Bad group status information was presented to the attendant.
Corrective Action Taken:	The lamp state was corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of RLTKSTAT tuple.
<i>Bad Data</i> =	Lamp state as found (0 - OFF; 1 - ON).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Group number.
<i>Dump</i> =	RLTKOWNER tuple.

Table 7 Error Code: TYPE

Error Code:	TYPE
Error Description:	The group is of the wrong hunt type to have a tuple in this relation.
Possible Error Effect:	An assert will fire if this group's hunt type is not the type used in rITKSTAT (that is, RTH_FWDLIN, RTH_BWDLIN, RTH_UCD, RTH_GUCD, RTH_BGUCD, RTH_PRFLIN, or RTH_PRBLIN) and the tuple could not be retrieved.
Corrective Action Taken:	The tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Hunt type of the group.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	grpnum of this tuple.
<i>Dump</i> =	rITKSTAT tuple.

33. TL - Z

TMRSM

Software Release: 5E13 and later

1. C7 LEVEL 3 TIMER FOR SM (TMRSM) AUDIT

The TMRSM audit was developed for the SS7MRE feature. The CCtmr_sm table contains level 3 timers for GSM. Data in this table is audited by the TMRSM audit which runs in GSMs to make sure that the data is correctly populated from static relation rIC7L3TMRS.

2. DATA STRUCTURES

For details on the fields contained in the RLC7L3TMRS relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx). For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: ARRAY_ENTRY_BAD

Error Code:	ARRAY_ENTRY_BAD
Error Description:	The audit found that one or more of the following field has a different value than found in rIC7L3TMRS tuple. • strmt1 • strmt2 • strmt3 • strmt4 • strmt5 • strmt6 • strmt7 • strmt8 • strmt10 • strmt12 • strmt13 • strmt14 • strmt22

	• strmt23 • strmt33 • strmt27 • strmt28 • strmt29 • strmt30 • strmt31 • slmt1 • slmt2 • slmt7 • slmt17 • smht1 • smht2 • slmansit19 • slmansit31 • t15_conv • t16_conv • t32_conv
Possible Error Effect:	Varies according to timer.
Corrective Action Taken:	This audit will report the error and correct the value of the fields based on the rIC7L3TMRS tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value of CCDEF100.
<i>Bad Data</i> =	Value of CCDEF200.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into rIC7L3TMRS[].
<i>Dump 1</i> =	Contents of CCTmr_sm[].
<i>Dump 2</i> =	Contents of rIC7L3TMRS[index].

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	The audit failed to read a static rIC7L3TMRS tuple for an equipped (in service) global switching module (GSM).
Possible Error Effect:	There is an office-dependent data (ODD) problem for this GSM.
Corrective Action Taken:	None.
Manual Action Required:	Office personnel should insert rIC7L3TMRS tuple using ODBE.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	The return value of the DB read for tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Index into rIC7L3TMRS[].

<i>Dump =</i>	NA.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	This error occurs as a result of a write protect violation, an out-of-range read, or an invalid operation by audit code.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	This error will be printed on the receive-only printer (ROP) and the audit will return.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump 1 =</i>	Contents of rIC7L3TMRS[index].
<i>Dump 2 =</i>	Contents of CCtmr_sm[].

TMSLNK

Software Release: 5E10 and later

1. TIME-MULTIPLEXED SWITCH LINK (TMSLNK) AUDIT

The time-multiplexed switch link (TMSLNK) is the fiber optic link between the time-multiplexed switch (TMS) in the office network and timing complex common (ONTCCOM) and the dual link interface (DLI) in the switching module (SM). Each local switching module contains two TMSLNKs per office network and timing complex (ONTC) side. Figure 1 shows these circuits.

Several audits check the peripheral configuration database (PCD) including the communication link audit (CLNK), the dual link interface audit (DLI), the peripheral configuration database audit (PCD), and the time-multiplexed switch link audit (TMSLNK).

The TMSLNK audit is part of the Switch Maintenance audits subgroup. It runs in the SMKP environment in the segmented mode.

2. DATA STRUCTURES

The TMSLNK audit checks the SMdlnktms array in the peripheral configuration database. The status of the TMSLNK is kept in the array; a static and a dynamic entry exist for each TMSLNK. The administrative module path data relation (RLPHDATA) contains the TMSLNK numbers and additional status information. The audit ensures that data in the PCD that is duplicated from data in the module data relation (RLMODATT) and the switching module time-multiplexed switch link relation (RLSTMSLNK) is consistent. These data structures are depicted in Figure 2 .

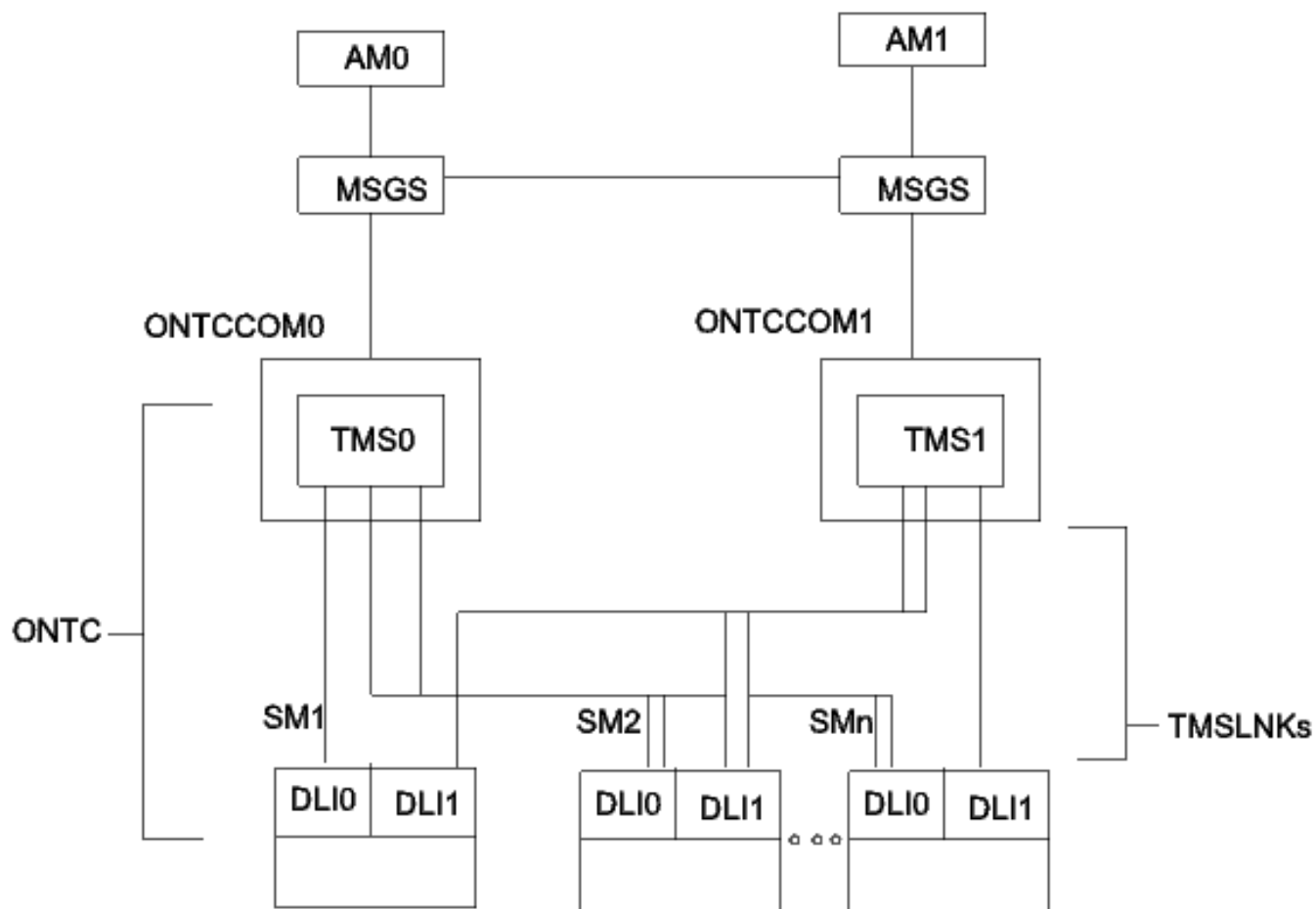


Figure 1 (TMSLNK-1) Time-Multiplexed Switch Link Circuits

ODD

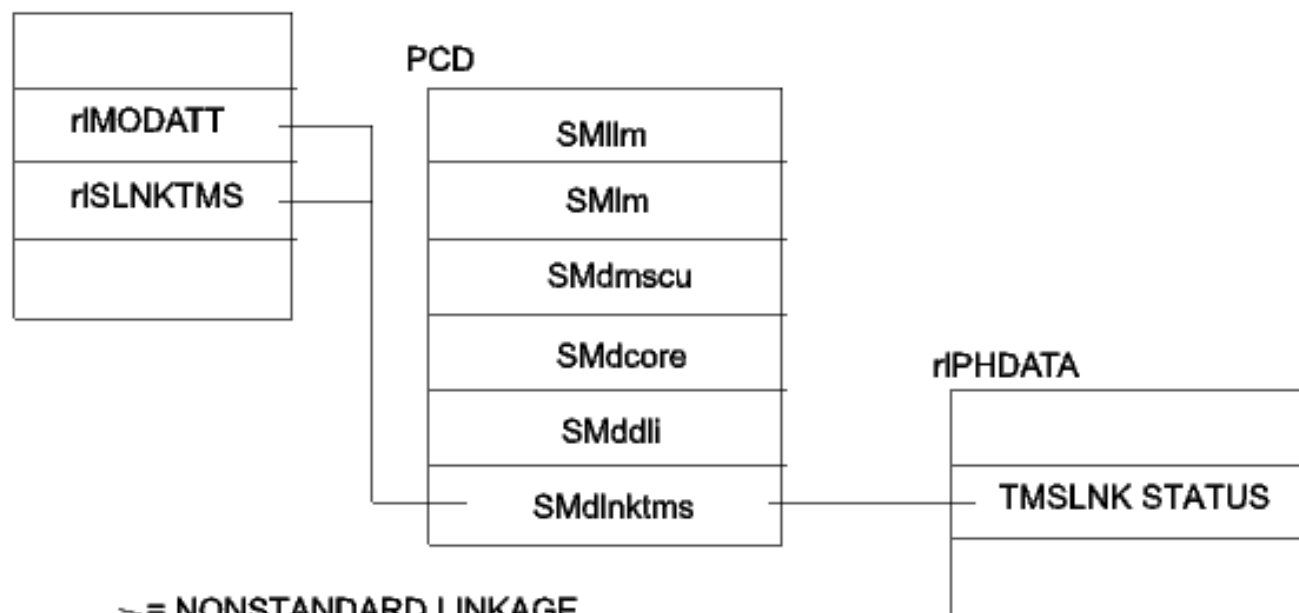


Figure 2 (TMSLNK-2) Data Structures of the TMSLNK Audit**3. AUDIT CHECKS**

The audit makes software checks, vertical checks, external checks, and configuration checks.

- (a) The audit checks for the internal consistency within the dynamic peripheral configuration database for the TMSLNK and checks the consistency between the office-dependent data (ODD) and the PCD. The TMSLNK must be in an allowable state. Allowable states include active (ACT) if the TMSLNK is operational and out-of-service unequipped null (OOS UNEQ NULL) if the TMSLNK is unequipped.
- (b) The vertical check ensures the consistency of the TMSLNK state in the PCD with the state of its controlling hardware units, the parent DLI and the ONTCCOM.
- (c) The audit makes external checks on the SMDlnkTms array and special data. The call processing status data in rIPHDATA must agree with the TMSLNK state data in the PCD. The control time slots (CTS) and SM number associated with the TMSLNK in the PCD must agree with the information in rIMODATT and rISLNKTMS.
- (d) The audit checks the consistency between the software state of the TMSLNK and its hardware configuration. The audit makes this check for communication module, model 2 (CM2) only. It does not make it for communication module, model 1 (CM1).

Before taking corrective action, the audit ensures that the TMSLNK is not in a transient state by checking the switch maintenance and recovery software (SMARS) queue. The SMARS queue contains a record of all maintenance activities. If fault recovery is in progress, the audit is rescheduled. If fault recovery is not in progress, an error is reported.

4. ERROR HANDLING

The type of error determines the actions the audit takes.

If a software internal check or a tree structure check fails, the audit places the software in a valid state.

If the external check fails, the audit assumes that the state of the TMSLNK is valid. The audit makes any data that corresponds to the TMSLNK state agree with the TMSLNK state. Any data that should correspond to data in the ODD is updated to agree with the ODD.

If the hardware/software configuration check fails, the audit assumes the software data is correct and tries to make the hardware configuration agree with the software. The audit may request a restoration of the TMSLNK.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: INDEX

Error Code:	INDEX
Error Description:	The audits cursor is out-of-range.
Possible Error Effect:	Audit could assert because it's trying to read information for a unit which does not exist.
Corrective Action Taken:	The cursor is reset to a valid value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.

<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Audits cursor value found to be bad.
<i>Dump</i> =	NA.

Table 2 Error Code: LNK2ALST

Error Code:	LNK2ALST
Error Description:	The time-multiplexed switch link (TMSLNK) state differs for the data which is used by switch maintenance to determine if service is degraded.
Possible Error Effect:	The service degrade indicator for a switching module could be incorrect.
Corrective Action Taken:	The audit updates the service degrade data to agree with the switch maintenances TMSLNK state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 3 Error Code: LNK_CONFIG

Error Code:	LNK_CONFIG
Error Description:	The TMSLNK hardware state differs for the switch maintenance TMSLNK state.
Possible Error Effect:	New call processing two way connect orders could be failing.
Corrective Action Taken:	The audit sends a message to restore the TMSLNK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 4 Error Code: LNK_CTS_NUM

Error Code:	LNK_CTS_NUM
Error Description:	The control time slot (CTS) numbers which are used by switch maintenance do not agree with the CTS numbers which are in the RLMODATT relation.
Possible Error Effect:	Switch maintenance may not be recovering the TMSLNK correctly since it is use bad CTS numbers.
Corrective Action Taken:	The audit updates the CTS numbers which switch maintenance uses to agree with the ones found in the RLMODATT relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 5 Error Code: LNK_EXTERNAL_LNKSTAT

Error Code:	LNK_EXTERNAL_LNKSTAT
Error Description:	The call processing TMSLNK state differs for the switch maintenance TMSLNK state.
Possible Error Effect:	New call processing two way connect orders are not being sent to an active TMSLNK or are being sent to an out-of-service TMSLNK.
Corrective Action Taken:	The audit updates the call processing TMSLNK state to agree with the switch maintenances TMSLNK state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.

<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 6 Error Code: LNK_FE

Error Code:	LNK_FE
Error Description:	The TMSLNK was in a out of service family state but the parent dual link interface (DLI) is in service.
Possible Error Effect:	The TMSLNK is out of service when it should not be.
Corrective Action Taken:	The audit restores the TMSLNK.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 7 Error Code: LNK_PCD

Error Code:	LNK_PCD
Error Description:	A TMSLNK has been found to be in an invalid state.
Possible Error Effect:	Fault recovery may be asserting on bad data. This could be the result of purging a fault recovery process in SMKP.
Corrective Action Taken:	The audit places the unit in a valid state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 8 Error Code: LNK_SM_NUM

Error Code:	LNK_SM_NUM
Error Description:	The switching module (SM) numbers which are used by switch maintenance do not agree with the SM numbers which are in the RLSLNKTMS relation.
Possible Error Effect:	Switch maintenance may not be recovering the TMSLNK correctly since it is use a bad SM number.
Corrective Action Taken:	The audit updates the SM numbers which switch maintenance uses to agree with the ones found in the RLSLNKTMS relation.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error is output here.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 9 Error Code: MODATT

Error Code:	MODATT
Error Description:	The audit has found a problem with the RLMODATT or RLSLNKTMS relation in the office-dependent database (ODD).
Possible Error Effect:	A SM may be having communication problems.
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats, check the RLMODATT relation tuple whose key is the SM number given in <i>Good Data</i> . Also check the RLSLNKTMS relation tuples whose TMSLNK number is dumped in <i>Logical Key</i> .
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	SM number used to read the RLMODATT relation.
<i>Logical Index</i> =	Number of the TMSLNK which was found in error.

<i>Dump 1</i> =	Local buffer used in DBfrdtup() call.
<i>Dump 2</i> =	Audit's copy of the data read.

Table 10 Error Code: PCD

Error Code:	PCD
Error Description:	The audit has found an error which it cannot correct this may be due to a bad ODD, bad static data or the audit recovery table needs to be changed.
Possible Error Effect:	Audit will not recover this error.
Corrective Action Taken:	The audit will not recover this error.
Manual Action Required:	Check if the ODD is correct for the peripheral configuration data (PCD). The relations this includes are: RLDSORE = For the ONTC. RLSLNKTMS = For the TMSLNK. RLMODATT = For the SMs.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Number of the TMSLNK which was found in error.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump</i> =	Audit's copy of the data read.

Table 11 Error Code: PCD_SLNKTMS

Error Code:	PCD_SLNKTMS
Error Description:	The audit has found a problem with the RLSLNKTMS relation in the ODD.
Possible Error Effect:	An TMSLNK may be out of service (OOS).
Corrective Action Taken:	None.
Manual Action Required:	If the error repeats check the RLSLNKTMS relation tuple whose key is the TMSLNK number and side give in <i>Good Data</i> and <i>Logical Index</i> .
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	DBfrdtup return code.
<i>Good Data</i> =	Number of the TMSLNK which was found in error is output here.
<i>Logical Index</i> =	ONTC side the error was found on (that is, 0 or 1).
<i>Dump 1</i> =	Equipment identifier of the unit.
<i>Dump 2</i> =	Local buffer used in DBfrdtup() call.
<i>Dump 3</i> =	Audits copy of the data read.

Table 12 Error Code: RSMNTS

Error Code:	RSMNTS
Error Description:	A failure occurred when attempting to read an rHSMRSM tuple for a host switching module (HSM). The rLMODATT tuple's SM type, for the specified module, may be incorrect or the database is missing the rHSMRSM tuple. This error will continue to print out until fixed by the user.
Possible Error Effect:	A SM may be having communication problems.
Corrective Action Taken:	None.
Manual Action Required:	Investigate the database and resolve the RLMODATT and RLHSMRSM inconsistencies with ODBE. Use the SM number dumped in <i>Bad Data</i> of this error report as the key to the two relations.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	SM number used to read the RLMODATT relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Number of the TMS link which was found in error.
<i>Dump</i> =	Unit's state (AUBIG ST structure).

Table 13 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	Causes resets, and or asserts.
Corrective Action Taken:	Return audit complete, wait for next go around.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The TMSLNK audit cursor.
<i>Dump =</i>	NA.

TNINFO

Software Release: 5E10 and later

1. TERMINATION NOTIFICATION INFORMATION (TNINFO) AUDIT

The advanced system platform (ASP) family of services includes the ability to handle calls with multiple legs. Termination notification information for each leg is kept in a tuple of RLTN_INFO.

2. DATA STRUCTURES

The TNINFO audit references dynamic relations:

RLNS_IDX = One tuple of this relation acts as the idle list headcell for relation RLTN_INFO. Other tuples may own busy lists of rITN_INFO tuples.

RLTN_INFO = The relation under audit.

3. AUDIT CHECKS

The TNINFO audit invokes the DRINASEK audit driver. In addition to those checks made by the driver, the TNINFO audit does the following:

If the tuple is not idle:

OUTOFRANGE = Range check the *leg_num* attribute.

BADUSE = Validate the *user* attribute.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	An rITN_INFO tuple is not idle (rITN_INFO.nsidx_mem.st_ownk != DBTNINFOIDLE + DBSFNS_IDX) and has an invalid <i>user</i> attribute (rITN_INFO.user != DMPVN U).
Possible Error Effect:	The call leg may be handled incorrectly.
Corrective Action Taken:	The rITN_INFO tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rITN_INFO.tninfo_id) and is then linked to the idle list (owned by the rINS_IDX tuple with an <i>nsidxkey</i> attribute DBTNINFOIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITN_INFO tuple.
<i>Bad Data</i> =	The invalid user (rITN_INFO.user).
<i>Good Data</i> =	The rITN_INFO's owner (rITN_INFO.nsidx_mem.st_ownk).
<i>Logical Index</i> =	Key of the rITN_INFO tuple (rITN_INFO.tninfo_id).
<i>Dump</i> =	The rITN_INFO tuple.

Table 2 Error Code: OUTOFRANGE

Error Code:	OUTOFRANGE
Error Description:	An rITN_INFO tuple is not idle (rITN_INFO.nsidx_mem.st_ownk !=

	DBTNINFOIDLE + DBSFNS_IDX) and has an invalid <i>leg_num</i> attribute (rITN_INFO.leg_num > FCMXSTRIG).
Possible Error Effect:	Billing for the call leg associated with this tuple will not be correct.
Corrective Action Taken:	The rITN_INFO tuple is unlinked from any list it may be in, has all fields zeroed out, reinitializes the key (rITN_INFO.tninfo_id) and is then linked to the idle list (owned by the rINS_IDX tuple with an <i>nsidxkey</i> attribute DBTNINFOIDLE).
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITN_INFO tuple.
<i>Bad Data</i> =	The invalid call leg (rITN_INFO.leg_num).
<i>Good Data</i> =	The rITN_INFO's owner (rITN_INFO.nsidx_mem.st_ownk).
<i>Logical Index</i> =	Key of the rITN_INFO tuple (rITN_INFO.tninfo_id).
<i>Dump</i> =	The rITN_INFO tuple.

TONECH-A

Software Release: 5E10 - 5E11

1. TONE TO CHANNEL (TONECH) AUDIT

The universal tone generator (UTG) circuit, located on the local digital service unit (LDSU), performs tone generating functions required for call processing. The circuit is capable of generating tones such as audible ringing, busy tone, dial tone, call-waiting tone, and receiver off-hook tone. Some tones require tone generation circuits (PCUTGs); others require transmission test facilities generator circuits (PCTTFGENs). The PCTTFGEN circuits are located on the global digital service unit (GDSU).

The tone to channel relation (RLTONECH) contains information about the type of circuit being used for a particular tone. If PCTTFGEN circuits are used, the RLTONECH relation also contains information about the peripheral side time slots being used. The tone to channel audit (TONECH) identifies inconsistencies, extra tuples, and bad keys in the RLTONECH relation.

This audit is part of the peripheral control audits subgroup. It runs in the operating system for distributed switch-switching module (ODSD-SM) environment.

2. DATA STRUCTURES

The data structures checked by the TONECH audit are the dynamic tone to channel relation (RLTONECH), the dynamic circuit data block common relation (RLCDBCOM), the dynamic channel data block relation (RLCHDB), and the local digital services unit (LDSU) unit type peripheral control global variable, PCar_utype. The audit also uses the static circuit data relation (RLCKTDATA), the static peripheral interface control bus relation (RLPICB), and the dynamic switch maintenance equipment status table relation (RLSMEST) to gather needed information. Figure 1 shows the fields in rLTONECH checked by the audit and its relationship to rLCDBCOM and rLCHDB.

3. AUDIT CHECKS

The tone to channel audit uses the dynamic driver (DRDYN) to retrieve tuples sequentially. The DRDYN driver calls a semantic check function to perform the following checks on rLTONECH tuples.

- (a) If a tone is associated with a PCTTFGEN circuit, the audit performs an existence check. If the switching module (SM) does not contain a GDSU containing a PCTTFGEN circuit, then the rLTONECH tuple for this circuit should be vacant.
- (b) If the tone is associated with a universal tone generator, the audit determines from the rLPICB relation on which board it exists. The circuit name is constructed and rLCKTDATA is read to find the associated rLSMEST tuple. If the equipage states of rLSMEST and rLCKTDATA agree, the audit performs the following checks.
- (c) The audit performs a key check on the tone field.
- (d) The audit performs point-to/point-back checks between rLTONECH and rLCDBCOM tuples.
- (e) If the tone is a tone associated with audible, the channel assignment field is compared with the global variable Glaudring. Also, the unit type of the LDSU is compared with the global variable PCar_utype.
- (f) If the tone is associated with a PCTTFGEN circuit, the audit performs a point-to/point-back check between the rLTONECH tuple and an rLCHDB tuple.
- (g) If the circuit is idle, the audit compares the channel assignment field with the global variable Glidlech.
- (h) The audit compares the base address field (baseadd) in rLTONECH against the base address field in rLCKTDATA.

4. ERROR HANDLING

If the key is incorrect, the correct value will be inserted into the tone field, and the audit will continue performing the remaining semantic checks. Any extra rITONECH tuples are deleted.

If an inconsistency exists between an rITONECH and rICKTDATA tuple the rITONECH tuple is changed to agree with the rICKTDATA tuple.

If a point-to-point-back check finds an inconsistency between rITONECH and rICDBCOM tuples, or between rITONECH and rICHDB tuples, the rITONECH tuple is zeroed and assigned a new rICDBCOM or rICHDB tuple.

If PCar_utype is found inconsistent with the LDSU type, PCar_utype is updated to indicate the current LDSU type.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: AUD_CH

Error Code:	AUD_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing uses.
Possible Error Effect:	Audible ringing tone is not available in this SM.
Corrective Action Taken:	The audible ring tone tuple will be recovered to the value of the global variable, Glaudring. Audible ring has been restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Contents of the channel field found in the rITONECH tuple.
<i>Good Data =</i>	Contents of the Glaudring variable.
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The rITONECH tuple.

Table 2 Error Code: AUD2_CHDB

Error Code:	AUD2_CHDB
Error Description:	Integrity checks on the phase 2 audible ring rICHDB tuple failed.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rICHDB tuple to the rITONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rICHDB tuple's link to rITONECH (tone field).
<i>Bad Data =</i>	rICHDB tuple's input process code (inproc field).
<i>Good Data =</i>	Phase 2 audible ring time slot.
<i>Logical Index =</i>	Logical key of the rITONECH tuple (tone field).
<i>Dump 1 =</i>	rICHDB tuple.
<i>Dump 2 =</i>	rITONECH tuple.

Table 3 Error Code: AUD3_CHDB

Error Code:	AUD3_CHDB
Error Description:	Integrity checks on the phase 3 audible ring rICHDB tuple failed.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rICHDB tuple to the rITONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rICHDB tuple's link to rITONECH (tone field).
<i>Bad Data =</i>	rICHDB tuple's input process code (inproc field).
<i>Good Data =</i>	Phase 3 audible ring time slot.
<i>Logical Index =</i>	Logical key of the rITONECH tuple (tone field).
<i>Dump 1 =</i>	rICHDB tuple.

<i>Dump 2 =</i>	rITONECH tuple.

Table 4 Error Code: AVN_AUD_RING

Error Code:	AVN_AUD_RING
Error Description:	Precedence audible ringing (AUTOVON's hyperactive version of audible ringing) has been found to be on when it should be off, or vice versa. This determination was made through a scan of the RLPP_LTM relation. If no tuples were found, the tone should be off. Otherwise it should be on. The results of this scan was matched against PC's semi-global AUTOVON activity indicator, and it was found that the indicator had the opposite (and wrong) value.
Possible Error Effect:	Precedence audible ringing is either: (1) Off, but there are customers needing it, or (2) On, and using up a peripheral channel needlessly.
Corrective Action Taken:	PCautovon has been corrected, and the tone has been turned on or off, as appropriate.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. NA.
<i>Bad Data =</i>	Contents of PCautovon before correction.
<i>Good Data =</i>	Contents of locally-generated AUTOVON indicator. This is what PCautovon (the <i>Bad Data</i>) should match.
<i>Logical Index =</i>	0. NA. NOTE: Both the GOOD and BAD data are booleans: 0 = DBNO (tone off) 1 = DBYES (tone on).
<i>Dump =</i>	NA.

Table 5 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	Either the linkage from the rITONECH tuple to the rICDBCOM tuple has been corrupted or a rICDBCOM tuple exists when the static rICKTDATA tuple does not exist.
Possible Error Effect:	Loss of audible ring to the switch.
Corrective Action Taken:	A new UTG circuit will be assigned for auditable ring.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Circuit name of bad UTG.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The offending rITONECH tuple.

Table 6 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global variable PCIdsu_type is in error. PCIdsu_type is used to store circuit names for equipped DSU(2) circuits (tone generators, tone decoders, tone transceivers, VBS transceivers, and revertive pulsing transceivers).
Possible Error Effect:	Foreground digit outputpulsing may not be executing.
Corrective Action Taken:	Correct the value of PCIdsu_type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCIdsu_type
<i>Bad Data =</i>	Value currently in PCIdsu_type

<i>Good Data</i> =	Value that should be in PCldsu_type
<i>Logical Index</i> =	0
<i>Dump</i> =	NA

Table 7 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The base address in the dynamic rITONECH tuple does not match the base address in the static rICKTDATA tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The base address in the rITONECH tuple will be recovered to the value of the base address field in the static rICKTDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Value of base address in the rITONECH tuple.
<i>Good Data</i> =	Value of base address in the rICKTDATA tuple.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 8 Error Code: BASEADD1

Error Code:	BASEADD1
Error Description:	The audit failed to get an associated rICKTDATA tuple or the base address in the dynamic rITONECH tuple does not match the base address in the static rICKTDATA tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rICKTDATA tuple read return value.
<i>Bad Data</i> =	Value of base address in the rITONECH tuple.
<i>Good Data</i> =	Value of base address in the rICKTDATA tuple.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump 1</i> =	rITONECH tuple.
<i>Dump 2</i> =	rICKTDATA tuple.

Table 9 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The rITONECH tuple is linked to a rCDBCOM tuple that does not have an associated rICKTDATA tuple.
Possible Error Effect:	The digital service unit (DSU) type cannot be identified.
Corrective Action Taken:	A rCDBCOM tuple which has an rICKTDATA tuple is assigned to the rITONECH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Key of the rCDBCOM tuple found in rITONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rITONECH tuple exists but the global variable GI_genttf indicates that there is not any TTF generators on this SM.
Possible Error Effect:	PC may erroneous use the rITONECH tuple.
Corrective Action Taken:	The rITONECH tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	GI_genttf.

<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 11 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing and peripheral control use.
Possible Error Effect:	Idle code (silent tone) is not available in this SM.
Corrective Action Taken:	The idle code tuple has been recovered to the value of the global variable, Glidlech. Idle code is restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Contents of the channel field found in the rITONECH tuple.
<i>Good Data</i> =	Contents of the Glidlech variable.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	The rITONECH tuple.

Table 12 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The idle tone was not initialized correctly.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The idle tone will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	rITONECH tuple's channel .
<i>Good Data</i> =	rICHDB tuple's channel number (chnum field).
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump 1</i> =	rICHDB tuple.
<i>Dump 2</i> =	rITONECH tuple.

Table 13 Error Code: LDSUCLNN

Error Code:	LDSUCLNN
Error Description:	The baseadd or ldsucInn field of this rITONECH tuple is not valid (other than zero) for the non-autovon precedence audible ring channel.
Possible Error Effect:	Precedence audible ringing tone is not available in this SM.
Corrective Action Taken:	The tone tuple will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Contents of the ldsucInn field found in the rITONECH tuple.
<i>Good Data</i> =	Zero (value the ldsucInn field should be).
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	The rITONECH tuple.

Table 14 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A rITONECH tuple with a null key was found to have non-zero values.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The rITONECH tuple is set to all zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Physical key (auP_KEY) of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 15 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	An in-service LDSF or GDSU circuit's r1CHDB tuple was found to have a null r1CDBCOM tuple link.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	Related audits CHDB and CDBCOM will be run to correct the linkage problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	r1TONECH tuple's channel.
<i>Bad Data =</i>	r1CHDB tuple's link to the r1CDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of r1TONECH tuple.
<i>Dump =</i>	NA.

Table 16 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	A r1TONECH tuple is inconsistent with an out of service LDSF.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The r1TONECH tuple is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of base address in the r1TONECH tuple.
<i>Bad Data =</i>	r1TONECH tuple's ldsu channel (ldsuc1nn field).
<i>Good Data =</i>	r1TONECH tuple's TSI peripheral data bus (channel field).
<i>Logical Index =</i>	r1TONECH tuple's tone type (tone field).
<i>Dump =</i>	Associated r1CHDB tuple.

Table 17 Error Code: PPAUD_CH

Error Code:	PPAUD_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing uses, or if this is a module that is not providing AUTOVON service currently, it is not set to PCTONEOS.
Possible Error Effect:	Precedence audible ringing tone is not available in this SM, if this is a module in which it should be being provided, or it is being provided when it shouldn't be and it's using up a peripheral channel needlessly.
Corrective Action Taken:	The audible ring tone tuple will be recovered to the value of the global variable G1paudring, or to PCTONEOS. Either way, if it should be on, it is, otherwise it's off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the r1TONECH tuple.
<i>Bad Data =</i>	Contents of the channel field found in the r1TONECH tuple.
<i>Good Data =</i>	Contents of the G1paudring variable or PCTONEOS (this is what the channel field should match).
<i>Logical Index =</i>	Logical key of the r1TONECH tuple.
<i>Dump =</i>	The r1TONECH tuple.
	NOTE: PCTONEOS currently has a value of 0x5ee5, but see header file pc/PCtonerqst.h for value now.

Table 18 Error Code: SMEST

Error Code:	SMEST
Error Description:	An inconsistent r1SMEST tuple key (unit_id) was encountered when trying to verify if a LDSF common board was in a transient state.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	CKTDATA related audit will run to correct problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LDSF unit index.
<i>Bad Data =</i>	r1SMEST tuple key (unit_id field).

<i>Good Data</i> =	LDSF common board circuit name (cdbcom_key).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 19 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A valid rITONECH tuple has an invalid value in its tone field.
Possible Error Effect:	The state of the rITONECH tuple is unknown.
Corrective Action Taken:	This error is flagged with an output message.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Bad tone value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	The offending rITONECH tuple.

Table 20 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rITONECH tuple and its associated rICDBCOM tuple failed the point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rITONECH tuple will be assigned a new rICDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Key of the rICDBCOM tuple found in rITONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump 1</i> =	rITONECH tuple.
<i>Dump 2</i> =	rICDBCOM tuple.

Table 21 Error Code: TONE_CHDB

Error Code:	TONE_CHDB
Error Description:	The rITONECH tuple and its associated rICHDB tuple failed the point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rICHDB tuple to the rITONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	The key of the rICHDB tuple found in rITONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump 1</i> =	rITONECH tuple.
<i>Dump 2</i> =	rICHDB tuple.

Table 22 Error Code: TONEBAD1

Error Code:	TONEBAD1
Error Description:	A non-operational universal tone generator's rITONECH tuple was found to have an in service ldsucInn or baseadd field.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rITONECH tuple.
<i>Bad Data</i> =	rITONECH tuple's ldsu channel (ldsucInn field).
<i>Good Data</i> =	Out of service field value (PCTONEOS).
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 23 Error Code: TONEBAD2

Error Code:	TONEBAD2
Error Description:	An out of service tone's rTONECH tuple was found to have an in service ldsucInn or baseadd field.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rTONECH tuple.
<i>Bad Data</i> =	rTONECH tuple's ldsu channel (ldsucInn field).
<i>Good Data</i> =	Out of service field value (PCTONEOS).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 24 Error Code: TYPE

Error Code:	TYPE
Error Description:	The global variable (PCar_utype) does not contain the correct LDSU unit type.
Possible Error Effect:	The switch would not be able to give immediate audible to a terminating call.
Corrective Action Taken:	The audit will update PCar_utype to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Dump of PCar_utype.
<i>Good Data</i> =	The correct value for LDSU unit type.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	Logical key of the rTONECH tuple.

TONECH-B

Software Release: 5E12 only

1. TONE TO CHANNEL (TONECH) AUDIT

The universal tone generator (UTG) circuit, located on the local digital service unit (LDSU), performs tone generating functions required for call processing. The circuit is capable of generating tones such as audible ringing, busy tone, dial tone, call-waiting tone, and receiver off-hook tone. Some tones require tone generation circuits (PCUTGs); others require transmission test facilities generator circuits (PCTTFGENs). The PCTTFGEN circuits are located on the global digital service unit (GDSU).

The tone to channel relation (RLTONECH) contains information about the type of circuit being used for a particular tone. If PCTTFGEN circuits are used, the RLTONECH relation also contains information about the peripheral side time slots being used. The tone to channel audit (TONECH) identifies inconsistencies, extra tuples, and bad keys in the RLTONECH relation.

This audit is part of the peripheral control audits subgroup. It runs in the ODSD-M environment.

2. DATA STRUCTURES

The data structures checked by the TONECH audit are the dynamic tone to channel relation (RLTONECH), the dynamic circuit data block common relation (RLCDBCOM), the dynamic channel data block relation (RLCHDB), and the local digital services unit (LDSU) unit type peripheral control global variable, PCar_otype. The audit also uses the static circuit data relation (RLCKTDATA), the static peripheral interface control bus relation (RLPICB), and the dynamic switch maintenance equipment status table relation (RLSMEST) to gather needed information. Figure 1 shows the fields in rLTONECH checked by the audit and its relationship to rLCDBCOM and rLCHDB.

3. AUDIT CHECKS

The tone to channel audit uses the dynamic driver (DRDYN) to retrieve tuples sequentially. The DRDYN driver calls a semantic check function to perform the following checks on rLTONECH tuples.

- (a) If a tone is associated with a PCTTFGEN circuit, the audit performs an existence check. If the switching module (SM) does not contain a GDSU containing a PCTTFGEN circuit, then the rLTONECH tuple for this circuit should be vacant.
- (b) If the tone is associated with a universal tone generator, the audit determines from the RLPICB relation on which board it exists. The circuit name is constructed and rLCKTDATA is read to find the associated rLSMEST tuple. If the equipage states of rLSMEST and rLCKTDATA agree, the audit performs the following checks.
- (c) The audit performs a key check on the tone field.
- (d) The audit performs point-to/point-back checks between rLTONECH and rLCDBCOM tuples.
- (e) If the tone is a tone associated with audible, the channel assignment field is compared with the global variable Glaudring. Also, the unit type of the LDSU is compared with the global variable PCar_otype.
- (f) If the tone is associated with a PCTTFGEN circuit, the audit performs a point-to/point-back check between the rLTONECH tuple and an rLCHDB tuple.
- (g) If the circuit is idle, the audit compares the channel assignment field with the global variable Glidlech.
- (h) The audit compares the base address field (baseadd) in rLTONECH against the base address field in rLCKTDATA.

4. ERROR HANDLING

If the key is incorrect, the correct value will be inserted into the tone field, and the audit will continue performing the remaining semantic checks. Any extra rITONECH tuples are deleted.

If an inconsistency exists between an rITONECH and rICKTDATA tuple the rITONECH tuple is changed to agree with the rICKTDATA tuple.

If a point-to-point-back check finds an inconsistency between rITONECH and rICDBCOM tuples, or between rITONECH and rICHDB tuples, the rITONECH tuple is zeroed and assigned a new rICDBCOM or rICHDB tuple.

If PCar_utype is found inconsistent with the LDSU type, PCar_utype is updated to indicate the current LDSU type.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: AUD_CH

Error Code:	AUD_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing uses.
Possible Error Effect:	Audible ringing tone is not available in this SM.
Corrective Action Taken:	The audible ring tone tuple will be recovered to the value of the global variable, Glaudring. Audible ring has been restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Contents of the channel field found in the rITONECH tuple.
<i>Good Data =</i>	Contents of the Glaudring variable.
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The rITONECH tuple.

Table 2 Error Code: AUD2_CHDB

Error Code:	AUD2_CHDB
Error Description:	Integrity checks on the phase 2 audible ring rICHDB tuple failed.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rICHDB tuple to the rITONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rICHDB tuple's link to rITONECH (<i>tone</i> field).
<i>Bad Data =</i>	rICHDB tuple's input process code (<i>inproc</i> field).
<i>Good Data =</i>	Phase 2 audible ring time slot.
<i>Logical Index =</i>	Logical key of the rITONECH tuple (<i>tone</i> field).
<i>Dump 1 =</i>	rICHDB tuple.
<i>Dump 2 =</i>	rITONECH tuple.

Table 3 Error Code: AUD3_CHDB

Error Code:	AUD3_CHDB
Error Description:	Integrity checks on the phase 3 audible ring rICHDB tuple failed.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rICHDB tuple to the rITONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rICHDB tuple's link to rITONECH (<i>tone</i> field).
<i>Bad Data =</i>	rICHDB tuple's input process code (<i>inproc</i> field).
<i>Good Data =</i>	Phase 3 audible ring time slot.
<i>Logical Index =</i>	Logical key of the rITONECH tuple (<i>tone</i> field).
<i>Dump 1 =</i>	rICHDB tuple.
<i>Dump 2 =</i>	rITONECH tuple.

Table 4 Error Code: AVN_AUD_RING

Error Code:	AVN_AUD_RING
Error Description:	Precedence audible ringing (AUTOVON's hyperactive version of audible ringing) has been found to be on when it should be off, or vice versa. This determination was made through a scan of the PP_LTM relation. If no tuples were found, the tone should be off. Otherwise it should be on. The results of this scan was matched against PC's semi-global AUTOVON activity indicator, and it was found that the indicator had the opposite (and wrong) value.
Possible Error Effect:	Precedence audible ringing is either: (1) Off, but there are customers needing it, or (2) On, and using up a peripheral channel needlessly.
Corrective Action Taken:	PCautovon has been corrected, and the tone has been turned on or off, as appropriate.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0. NA.
<i>Bad Data =</i>	Contents of PCautovon before correction.
<i>Good Data =</i>	Contents of locally-generated AUTOVON indicator. This is what PCautovon (that is, the <i>Bad Data</i>) should match.
<i>Logical Index =</i>	0. NA. NOTE: Both the <i>Good Data</i> and <i>Bad Data</i> are booleans: 0 = DBNO (tone off). 1 = DBYES (tone on).
<i>Dump =</i>	NA.

Table 5 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	Either the linkage from the rITONECH tuple to the rICDBCOM tuple has been corrupted or a rICDBCOM tuple exists when the static rICKTDATA tuple does not exist.
Possible Error Effect:	Loss of audible ring to the switch.
Corrective Action Taken:	A new UTG circuit will be assigned for auditable ring.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Circuit name of bad UTG.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The offending rITONECH tuple.

Table 6 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global variable PCldsu_type is in error. PCldsu_type is used to store circuit names for equipped DSU(2) circuits (tone generators, tone decoders, tone transceivers, VBS transceivers, and reverive pulsing transceivers).
Possible Error Effect:	Foreground digit outpulsing may not be executing.
Corrective Action Taken:	Correct the value of PCldsu_type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCldsu_type.
<i>Bad Data =</i>	Value currently in PCldsu_type.
<i>Good Data =</i>	Value that should be in PCldsu_type.
<i>Logical Index =</i>	0

<i>Dump</i> =	NA

Table 7 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The base address in the dynamic rITONECH tuple does not match the base address in the static rICKTDATA tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The base address in the rITONECH tuple will be recovered to the value of the base address field in the static rICKTDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Value of base address in the rITONECH tuple.
<i>Good Data</i> =	Value of base address in the rICKTDATA tuple.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 8 Error Code: BASEADD1

Error Code:	BASEADD1
Error Description:	The audit failed to get an associated rICKTDATA tuple or the base address in the dynamic rITONECH tuple does not match the base address in the static rICKTDATA tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rICKTDATA tuple read return value.
<i>Bad Data</i> =	Value of base address in the rITONECH tuple.
<i>Good Data</i> =	Value of base address in the rICKTDATA tuple.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump 1</i> =	rITONECH tuple.
<i>Dump 2</i> =	rICKTDATA tuple.

Table 9 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The rITONECH tuple is linked to a rICDBCOM tuple that does not have an associated rICKTDATA tuple.
Possible Error Effect:	The digital service unit (DSU) type cannot be identified.
Corrective Action Taken:	A rICDBCOM tuple which has an rICKTDATA tuple is assigned to the rITONECH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Key of the rICDBCOM tuple found in rITONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rITONECH tuple exists but the global variable GI_genttf indicates that there is not any TTF generators on this SM.
Possible Error Effect:	PC may erroneous use the rITONECH tuple.
Corrective Action Taken:	The rITONECH tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	GI_genttf.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 11 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing and peripheral control use.
Possible Error Effect:	Idle code (silent tone) is not available in this SM.
Corrective Action Taken:	The idle code tuple has been recovered to the value of the global variable, Glidlech. Idle code is restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Contents of the channel field found in the rITONECH tuple.
<i>Good Data =</i>	Contents of the Glidlech variable.
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The rITONECH tuple.

Table 12 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The idle tone was not initialized correctly.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The idle tone will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	rITONECH tuple's <i>channel</i> .
<i>Good Data =</i>	rCHDB tuple's channel number (<i>chnum</i> field).
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump 1 =</i>	rCHDB tuple.
<i>Dump 2 =</i>	rITONECH tuple.

Table 13 Error Code: LDSUCLNN

Error Code:	LDSUCLNN
Error Description:	The <i>baseadd</i> or <i>ldsucnn</i> field of this rITONECH tuple is not valid (other than zero) for the non-autovon precedence audible ring channel.
Possible Error Effect:	Precedence audible ringing tone is not available in this SM.
Corrective Action Taken:	The tone tuple will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Contents of the <i>ldsucnn</i> field found in the rITONECH tuple.
<i>Good Data =</i>	Zero (value the <i>ldsucnn</i> field should be).
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The rITONECH tuple.

Table 14 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A rITONECH tuple with a null key was found to have non-zero values.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The rITONECH tuple is set to all zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Physical key (auP_KEY) of the rITONECH tuple.
<i>Dump =</i>	rITONECH tuple.

Table 15 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	An in-service LDSF or GDSU circuit's rCHDB tuple was found to have a null

	riCDBCOM tuple link.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	Related audits CHDB and CDBCOM will be run to correct the linkage problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	riTONECH tuple's channel.
<i>Bad Data =</i>	riCHDB tuple's link to the riCDBCOM tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of riTONECH tuple.
<i>Dump =</i>	NA.

Table 16 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	A riTONECH tuple is inconsistent with an out of service LDSF.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The riTONECH tuple is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Value of base address in the riTONECH tuple.
<i>Bad Data =</i>	riTONECH tuple's ldsu channel (<i>ldsucInn</i> field).
<i>Good Data =</i>	riTONECH tuple's TSI peripheral data bus (<i>channel</i> field).
<i>Logical Index =</i>	riTONECH tuple's tone type (<i>tone</i> field).
<i>Dump =</i>	Associated riCHDB tuple.

Table 17 Error Code: PPAUD_CH

Error Code:	PPAUD_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing uses, or if this is a module that is not providing AUTOVON service currently, it is not set to PCTONEOS.
Possible Error Effect:	Precedence audible ringing tone is not available in this SM, if this is a module in which it should be being provided, or it is being provided when it shouldn't be and it's using up a peripheral channel needlessly.
Corrective Action Taken:	The audible ring tone tuple will be recovered to the value of the global variable Glpaudring, or to PCTONEOS. Either way, if it should be on, it is, otherwise it's off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the riTONECH tuple.
<i>Bad Data =</i>	Contents of the channel field found in the riTONECH tuple.
<i>Good Data =</i>	Contents of the Glpaudring variable or PCTONEOS (this is what the channel field should match).
<i>Logical Index =</i>	Logical key of the riTONECH tuple.
<i>Dump =</i>	The riTONECH tuple.
	NOTE: PCTONEOS currently has a value of 0x5ee5, but see header file pc/PCTonerqst.h for value now.

Table 18 Error Code: SMEST

Error Code:	SMEST
Error Description:	An inconsistent riSMEST tuple key (<i>unit_id</i>) was encountered when trying to verify if a LDSF common board was in a transient state.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	CKTDATA related audit will run to correct problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	LDSF unit index.
<i>Bad Data =</i>	riSMEST tuple key (<i>unit_id</i> field).
<i>Good Data =</i>	LDSF common board circuit name (<i>cdbcom_key</i>).
<i>Logical Index =</i>	NA.

<i>Dump</i> =	NA.

Table 19 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A valid rTONECH tuple has an invalid value in its tone field.
Possible Error Effect:	The state of the rTONECH tuple is unknown.
Corrective Action Taken:	This error is flagged with an output message.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Bad tone value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	The offending rTONECH tuple.

Table 20 Error Code: TONE_CDB

Error Code:	TONE_CDB
Error Description:	The rTONECH tuple and its associated rCDBCOM tuple failed the point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rTONECH tuple will be assigned a new rCDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Key of the rCDBCOM tuple found in rTONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump 1</i> =	rTONECH tuple.
<i>Dump 2</i> =	rCDBCOM tuple.

Table 21 Error Code: TONE_CHDB

Error Code:	TONE_CHDB
Error Description:	The rTONECH tuple and its associated rCHDB tuple failed the point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rCHDB tuple to the rTONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	The key of the rCHDB tuple found in rTONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump 1</i> =	rTONECH tuple.
<i>Dump 2</i> =	rCHDB tuple.

Table 22 Error Code: TONEBAD1

Error Code:	TONEBAD1
Error Description:	A non-operational universal tone generator's rTONECH tuple was found to have an in service <i>ldsucInn</i> or <i>baseadd</i> field.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rTONECH tuple.
<i>Bad Data</i> =	rTONECH tuple's ldsu channel (<i>ldsucInn</i> field).
<i>Good Data</i> =	Out of service field value (PCTONEOS).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 23 Error Code: TONEBAD2

Error Code:	TONEBAD2

Error Description:	An out of service tone's rTONECH tuple was found to have an in service <i>ldsucInn</i> or <i>baseadd</i> field.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rTONECH tuple.
<i>Bad Data</i> =	rTONECH tuple's ldsu channel (<i>ldsucInn</i> field).
<i>Good Data</i> =	Out of service field value (PCTONEOS).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 24 Error Code: TYPE

Error Code:	TYPE
Error Description:	The global variable (PCar_utype) does not contain the correct LDSU unit type.
Possible Error Effect:	The switch would not be able to give immediate audible to a terminating call.
Corrective Action Taken:	The audit will update PCar_utype to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Dump of PCar_utype.
<i>Good Data</i> =	The correct value for LDSU unit type.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	Logical key of the rTONECH tuple.

TONECH-C

Software Release: 5E13 and later

1. TONE TO CHANNEL (TONECH) AUDIT

The universal tone generator (UTG) circuit, located on the local digital service unit (LDSU), performs tone generating functions required for call processing. The circuit is capable of generating tones such as audible ringing, busy tone, dial tone, call-waiting tone, and receiver off-hook tone. Some tones require tone generation circuits (PCUTGs); others require transmission test facilities generator circuits (PCTTFGENs). The PCTTFGEN circuits are located on the global digital service unit (GDSU).

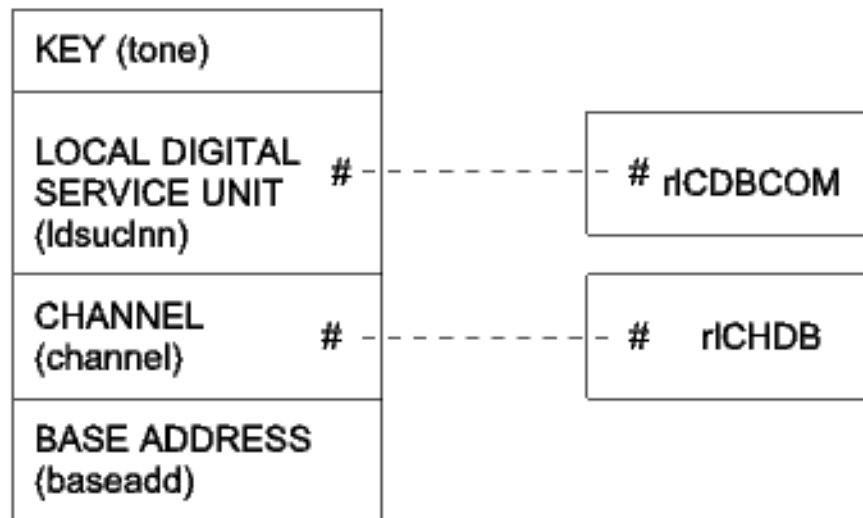
The tone to channel relation (RLTONECH) contains information about the type of circuit being used for a particular tone. If PCTTFGEN circuits are used, the RLTONECH relation also contains information about the peripheral side time slots being used. The tone to channel audit (TONECH) identifies inconsistencies, extra tuples, and bad keys in the RLTONECH relation.

This audit is part of the peripheral control audits subgroup. It runs in the ODSD-M environment.

2. DATA STRUCTURES

The data structures checked by the TONECH audit are the dynamic tone to channel relation (RLTONECH), the dynamic circuit data block common relation (RLCDBCOM), the dynamic channel data block relation (RLCHDB), and the local digital services unit (LDSU) unit type peripheral control global variable, PCar_utype. The audit also uses the static circuit data relation (RLCKTDATA), the static peripheral interface control bus relation (RLPICB), and the dynamic switch maintenance equipment status table relation (RLSMEST) to gather needed information. Figure 1 shows the fields in rITONECH checked by the audit and its relationship to rICDBCOM and rICHDB.

rITONECH



---- # = NONSTANDARD LINKAGE

Figure 1 (TONECH-1) Tone to Channel Relation

3. AUDIT CHECKS

The tone to channel audit uses the dynamic driver (DRDYN) to retrieve tuples sequentially. The DRDYN driver calls a semantic check function to perform the following checks on rITONECH tuples.

- (a) If a tone is associated with a PCTTFGEN circuit, the audit performs an existence check. If the switching module (SM) does not contain a GDSU containing a PCTTFGEN circuit, then the rITONECH tuple for this circuit should be vacant.
- (b) If the tone is associated with a universal tone generator, the audit determines from the RLPICB relation on which board it exists. The circuit name is constructed and rICKTDATA is read to find the associated rISMEST tuple. If the equipage states of rISMEST and rICKTDATA agree, the audit performs the following checks.
 - (c) The audit performs a key check on the tone field.
 - (d) The audit performs point-to/point-back checks between rITONECH and rICDBCOM tuples.
- (e) If the tone is a tone associated with audible, the channel assignment field is compared with the global variable Glaudring. Also, the unit type of the LDSU is compared with the global variable PCar_otype.
- (f) If the tone is associated with a PCTTFGEN circuit, the audit performs a point-to/point-back check between the rITONECH tuple and an rICHDB tuple.
- (g) If the circuit is idle, the audit compares the channel assignment field with the global variable Glidlech.
- (h) The audit compares the base address field (baseadd) in rITONECH against the base address field in rICKTDATA.

4. ERROR HANDLING

If the key is incorrect, the correct value will be inserted into the tone field, and the audit will continue performing the remaining semantic checks. Any extra rITONECH tuples are deleted.

If an inconsistency exists between an rITONECH and rICKTDATA tuple the rITONECH tuple is changed to agree with the rICKTDATA tuple.

If a point-to/point-back check finds an inconsistency between rITONECH and rICDBCOM tuples, or between rITONECH and rICHDB tuples, the rITONECH tuple is zeroed and assigned a new rICDBCOM or rICHDB tuple.

If PCar_otype is found inconsistent with the LDSU type, PCar_otype is updated to indicate the current LDSU type.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: AUD_CH

Error Code:	AUD_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing uses.
Possible Error Effect:	Audible ringing tone is not available in this SM.
Corrective Action Taken:	The audible ring tone tuple will be recovered to the value of the global variable, Glaudring. Audible ring has been restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITONECH tuple.
<i>Bad Data =</i>	Contents of the channel field found in the rITONECH tuple.
<i>Good Data =</i>	Contents of the Glaudring variable.
<i>Logical Index =</i>	Logical key of the rITONECH tuple.
<i>Dump =</i>	The rITONECH tuple.

Table 2 Error Code: AUD2_CHDB

Error Code:	AUD2_CHDB
Error Description:	Integrity checks on the phase 2 audible ring rCHDB tuple failed.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rCHDB tuple to the rTONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCHDB tuple's link to rTONECH (<i>tone</i> field).
<i>Bad Data</i> =	rCHDB tuple's input process code (<i>inproc</i> field).
<i>Good Data</i> =	Phase 2 audible ring time slot.
<i>Logical Index</i> =	Logical key of the rTONECH tuple (<i>tone</i> field).
<i>Dump 1</i> =	rCHDB tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 3 Error Code: AUD3_CHDB

Error Code:	AUD3_CHDB
Error Description:	Integrity checks on the phase 3 audible ring rCHDB tuple failed.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rCHDB tuple to the rTONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rCHDB tuple's link to rTONECH (<i>tone</i> field).
<i>Bad Data</i> =	rCHDB tuple's input process code (<i>inproc</i> field).
<i>Good Data</i> =	Phase 3 audible ring time slot.
<i>Logical Index</i> =	Logical key of the rTONECH tuple (<i>tone</i> field).
<i>Dump 1</i> =	rCHDB tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 4 Error Code: AVN_AUD_RING

Error Code:	AVN_AUD_RING
Error Description:	Precedence audible ringing (AUTOVON's hyperactive version of audible ringing) has been found to be on when it should be off, or vice versa. This determination was made through a scan of the PP_LTM relation. If no tuples were found, the tone should be off. Otherwise it should be on. The results of this scan was matched against PC's semi-global AUTOVON activity indicator, and it was found that the indicator had the opposite (and wrong) value.
Possible Error Effect:	Precedence audible ringing is either: (1) Off, but there are customers needing it, or (2) On, and using up a peripheral channel needlessly.
Corrective Action Taken:	PCautovon has been corrected, and the tone has been turned on or off, as appropriate.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0. NA.
<i>Bad Data</i> =	Contents of PCautovon before correction.
<i>Good Data</i> =	Contents of locally-generated AUTOVON indicator. This is what PCautovon (that is, the <i>Bad Data</i>) should match.
<i>Logical Index</i> =	0. NA.
	NOTE: Both the <i>Good Data</i> and <i>Bad Data</i> are booleans: 0 = DBNO (tone off). 1 = DBYES (tone on).
<i>Dump</i> =	NA.

Table 5 Error Code: BACKLINK

Error Code:	BACKLINK
Error Description:	Either the linkage from the rITONECH tuple to the rCDBCOM tuple has been corrupted or a rCDBCOM tuple exists when the static rICKTDATA tuple does not exist.
Possible Error Effect:	Loss of audible ring to the switch.
Corrective Action Taken:	A new UTG circuit will be assigned for auditable ring.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Circuit name of bad UTG.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	The offending rITONECH tuple.

Table 6 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The global variable PCldsu_type is in error. PCldsu_type is used to store circuit names for equipped DSU(2) circuits (tone generators, tone decoders, tone transceivers, VBS transceivers, and reverive pulsing transceivers).
Possible Error Effect:	Foreground digit outpulsing may not be executing.
Corrective Action Taken:	Correct the value of PCldsu_type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCldsu_type.
<i>Bad Data</i> =	Value currently in PCldsu_type.
<i>Good Data</i> =	Value that should be in PCldsu_type.
<i>Logical Index</i> =	0
<i>Dump</i> =	NA

Table 7 Error Code: BASEADD

Error Code:	BASEADD
Error Description:	The base address in the dynamic rITONECH tuple does not match the base address in the static rICKTDATA tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The base address in the rITONECH tuple will be recovered to the value of the base address field in the static rICKTDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Value of base address in the rITONECH tuple.
<i>Good Data</i> =	Value of base address in the rICKTDATA tuple.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	rITONECH tuple.

Table 8 Error Code: BASEADD1

Error Code:	BASEADD1
Error Description:	The audit failed to get an associated rICKTDATA tuple or the base address in the dynamic rITONECH tuple does not match the base address in the static rICKTDATA tuple.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rICKTDATA tuple read return value.
<i>Bad Data</i> =	Value of base address in the rITONECH tuple.
<i>Good Data</i> =	Value of base address in the rICKTDATA tuple.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump 1</i> =	rITONECH tuple.
<i>Dump 2</i> =	rICKTDATA tuple.

Table 9 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The rTONECH tuple is linked to a rCDBCOM tuple that does not have an associated rICKTDATA tuple.
Possible Error Effect:	The digital service unit (DSU) type cannot be identified.
Corrective Action Taken:	A rCDBCOM tuple which has an rICKTDATA tuple is assigned to the rTONECH.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Key of the rCDBCOM tuple found in rTONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	NA.

Table 10 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A rTONECH tuple exists but the global variable Gl_genttf indicates that there is not any TTF generators on this SM.
Possible Error Effect:	PC may erroneous use the rTONECH tuple.
Corrective Action Taken:	The rTONECH tuple will be deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Gl_genttf.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 11 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing and peripheral control use.
Possible Error Effect:	Idle code (silent tone) is not available in this SM.
Corrective Action Taken:	The idle code tuple has been recovered to the value of the global variable, Glidlech. Idle code is restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Contents of the channel field found in the rTONECH tuple.
<i>Good Data</i> =	Contents of the Glidlech variable.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	The rTONECH tuple.

Table 12 Error Code: INV_IDLE

Error Code:	INV_IDLE
Error Description:	The idle tone was not initialized correctly.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The idle tone will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	rTONECH tuple's channel.
<i>Good Data</i> =	rCHDB tuple's channel number (chnum field).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump 1</i> =	rCHDB tuple.
<i>Dump 2</i> =	rTONECH tuple.

Table 13 Error Code: LDSUCLNN

Error Code:	LDSUCLNN
Error Description:	The baseadd or ldsuclnn field of this rTONECH tuple is not valid (other than zero) for the non-autovon precedence audible ring channel.
Possible Error Effect:	Precedence audible ringing tone is not available in this SM.

Corrective Action Taken:	The tone tuple will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Contents of the <i>ldsucInn</i> field found in the rTONECH tuple.
<i>Good Data</i> =	Zero (value the <i>ldsucInn</i> field should be).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	The rTONECH tuple.

Table 14 Error Code: NON_NULL

Error Code:	NON_NULL
Error Description:	The rTONECH tuple associated with carrier tone was found to have a non-null <i>ldsucInn</i> , <i>channel</i> or <i>baseadd</i> field when the system indicates that carrier tone is inactive (Glcartone = DBNO).
Possible Error Effect:	System resources are allocated to an inactive system tone causing these resources to be lost to the system.
Corrective Action Taken:	The rTONECH tuple is recovered to an idle state and the associated system resources are released.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rTONECH tuple.
<i>Bad Data</i> =	Out of service field value (PCTONEOS).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 15 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A rTONECH tuple with a null key was found to have non-zero values.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The rTONECH tuple is set to all zeros.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Physical key (auP_KEY) of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 16 Error Code: NULL3LINK

Error Code:	NULL3LINK
Error Description:	An inservice LDSF or GDSU circuit's rCHDB tuple was found to have a null rCDBCOM tuple link.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	Related audits CHDB and CDBCOM will be run to correct the linkage problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rTONECH tuple's channel.
<i>Bad Data</i> =	rCHDB tuple's link to the rCDBCOM tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rTONECH tuple.
<i>Dump</i> =	NA.

Table 17 Error Code: OOSINV

Error Code:	OOSINV
Error Description:	A rTONECH tuple is inconsistent with an out of service LDSF.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The rTONECH tuple is recovered.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Value of base address in the rITONECH tuple.
<i>Bad Data</i> =	rITONECH tuple's ldsu channel (<i>ldsucInn</i> field).
<i>Good Data</i> =	rITONECH tuple's TSI peripheral data bus (<i>channel</i> field).
<i>Logical Index</i> =	rITONECH tuple's tone type (<i>tone</i> field).
<i>Dump</i> =	Associated rICHDB tuple.

Table 18 Error Code: PPAUD_CH

Error Code:	PPAUD_CH
Error Description:	The channel field of this tuple is not the same as the channel assignment that call processing uses, or if this is a module that is not providing AUTOVON service currently, it is not set to PCTONEOS.
Possible Error Effect:	Precedence audible ringing tone is not available in this SM, if this is a module in which it should be being provided, or it is being provided when it shouldn't be and it's using up a peripheral channel needlessly.
Corrective Action Taken:	The audible ring tone tuple will be recovered to the value of the global variable Glpaudring, or to PCTONEOS. Either way, if it should be on, it is, otherwise it's off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Contents of the channel field found in the rITONECH tuple.
<i>Good Data</i> =	Contents of the Glpaudring variable or PCTONEOS (this is what the channel field should match).
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	The rITONECH tuple.
	NOTE: PCTONEOS currently has a value of 0x5ee5, but see header file pc/PCtonerqst.h for value now.

Table 19 Error Code: SMEST

Error Code:	SMEST
Error Description:	An inconsistent rSMEST tuple key (<i>unit_id</i>) was encountered when trying to verify if a LDSF common board was in a transient state.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	CKTDATA related audit will run to correct problem.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	LDSF unit index.
<i>Bad Data</i> =	rSMEST tuple key (<i>unit_id</i> field).
<i>Good Data</i> =	LDSF common board circuit name (cdbcom key).
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

Table 20 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	A valid rITONECH tuple has an invalid value in its tone field.
Possible Error Effect:	The state of the rITONECH tuple is unknown.
Corrective Action Taken:	This error is flagged with an output message.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Bad tone value.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	The offending rITONECH tuple.

Table 21 Error Code: TONE_CDB

Error Code:	TONE_CDB

Error Description:	The rTONECH tuple and its associated rCDBCOM tuple failed the point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The rTONECH tuple will be assigned a new rCDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	Key of the rCDBCOM tuple found in rTONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump 1</i> =	rTONECH tuple.
<i>Dump 2</i> =	rCDBCOM tuple.

Table 22 Error Code: TONE_CHDB

Error Code:	TONE_CHDB
Error Description:	The rTONECH tuple and its associated rCHDB tuple failed the point-to-point-back check.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The audit will assign a new rCHDB tuple to the rTONECH tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTONECH tuple.
<i>Bad Data</i> =	The key of the rCHDB tuple found in rTONECH.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump 1</i> =	rTONECH tuple.
<i>Dump 2</i> =	rCHDB tuple.

Table 23 Error Code: TONEBAD1

Error Code:	TONEBAD1
Error Description:	A non-operational universal tone generator's rTONECH tuple was found to have an in service <i>ldsucInn</i> or <i>baseadd</i> field.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rTONECH tuple.
<i>Bad Data</i> =	rTONECH tuple's <i>ldsucInn</i> channel (<i>ldsucInn</i> field).
<i>Good Data</i> =	Out of service field value (PCTONEOS).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 24 Error Code: TONEBAD2

Error Code:	TONEBAD2
Error Description:	An out of service tone's rTONECH tuple was found to have an in service <i>ldsucInn</i> or <i>baseadd</i> field.
Possible Error Effect:	A system tone may be lost.
Corrective Action Taken:	The tone is recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rTONECH tuple.
<i>Bad Data</i> =	rTONECH tuple's <i>ldsucInn</i> channel (<i>ldsucInn</i> field).
<i>Good Data</i> =	Out of service field value (PCTONEOS).
<i>Logical Index</i> =	Logical key of the rTONECH tuple.
<i>Dump</i> =	rTONECH tuple.

Table 25 Error Code: TYPE

Error Code:	TYPE
Error Description:	The global variable (PCar_otype) does not contain the correct LDSU unit type.
Possible Error Effect:	The switch would not be able to give immediate audible to a terminating call.
Corrective Action Taken:	The audit will update PCar_otype to the correct value.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	Address of the rITONECH tuple.
<i>Bad Data</i> =	Dump of PCar utype.
<i>Good Data</i> =	The correct value for LDSU unit type.
<i>Logical Index</i> =	Logical key of the rITONECH tuple.
<i>Dump</i> =	Logical key of the rITONECH tuple.

TPCBUF

Software Release: 5E10 and later

1. TERMINAL PROCESS CONTROL BUFFER (TPCBUF) AUDIT

There are several types of ports in the switch that need administering, including line unit (LU) ports, SLC[®] 96 ports, and integrated services digital network (ISDN) B- and D-channel ports that connect to the line group controllers (LGCs). The integrated services port administrator (ISPA) takes care of port functions such as routine conditioning to ensure that the hardware and software are consistent. Two audits check the tuple buffers associated with the system and terminal processes involved with port administration, the integrated services port administration buffer audit (ISPABUF) and TPCBUF.

The TPCBUF audit is part of the Terminal Maintenance audits subgroup. It runs in the OSDS-M environment in the segmented and nonsegmented modes.

2. DATA STRUCTURES

The TPCBUF audit uses two data structures, the terminal process control buffers relation (RLTPCBUF) and the integrated services port administration buffer relation (RLISPABUF). The tuples of RLTPCBUF contain a process ID field and a timer field. They are linked to the system process tuples of RLISPABUF, as shown in Figure 1.

The terminal processes for which the rLTPCBUF tuples are used include process reports, the restoration and removal of ports, the conditioning of ports, and the performance of loop-around tests.

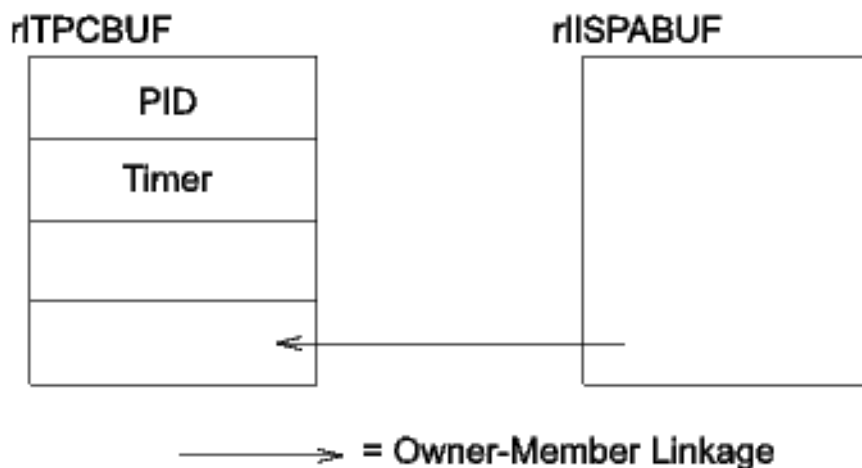


Figure 1 (TPCBUF-1) Data Structures of the TPCBUF Audit

3. AUDIT CHECKS

The TPCBUF audit invokes the DRINASEK audit driver to check the linkages among rLTPCBUF tuples and between rLTPCBUF tuples and rLISPABUF tuples. The driver then calls the semantic check function, which verifies that the process ID is for a live process. The semantic function also checks a timer to ensure that the job is completed within the specified time period.

4. ERROR HANDLING

If the process ID does not have an associated process, the tuple is recovered to the idle list. If the rLTPCBUF tuple reaches its time-out value, it is recovered to the idle list.

The TPCBUF audits schedules the ISPABUF audit as a related audit when it detects errors.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The rITPCBUF tuple is on one of the ISPA busy queues but the terminal process ID found in the tuple is for a process that does not exist.
Possible Error Effect:	The rITPCBUF tuple is a lost resource.
Corrective Action Taken:	The rITPCBUF tuple is returned to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITPCBUF tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITPCBUF tuple.
<i>Dump</i> =	rITPCBUF tuple.

Table 2 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The link from the member rITPCBUF tuple to the rIISPABUF owner tuple is out of range.
Possible Error Effect:	The linkage to the rIISPABUF tuple has been broken.
Corrective Action Taken:	The rITPCBUF tuple is returned to the idle list and the terminal process will be purged by the audit driver.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address to owner link field.
<i>Bad Data</i> =	Value of owner link field.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITPCBUF tuple.
<i>Dump</i> =	rITPCBUF tuple.

Table 3 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	The rITPCBUF tuple is on the idle list; however, it contains the process ID for an alive process.
Possible Error Effect:	A terminal process exists but will not be serviced by ISPA since the rITPCBUF tuples is not on one of the ISPA priority queues.
Corrective Action Taken:	The rITPCBUF tuple is zeroed out, returned to the idle list, and the terminal process is purged by the audit driver.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITPCBUF tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key of rITPCBUF tuple.
<i>Dump</i> =	rITPCBUF tuple.

Table 4 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	The terminal process has exceeded the maximum allowable time to be under control of ISPA system process.
Possible Error Effect:	The rITPCBUF tuple is a lost resource.
Corrective Action Taken:	The rITPCBUF tuple is returned to the idle list and the terminal process is purged by the audit driver.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	Address of time stamp attribute in rTPCBUF tuple.
<i>Bad Data</i> =	Length of time (in msec) terminal process has been alive.
<i>Good Data</i> =	Length of time (in msec) allowed for terminal process to be alive.
<i>Logical Index</i> =	Key of rTPCBUF tuple.
<i>Dump 1</i> =	rTPCBUF tuple.
<i>Dump 2</i> =	rISPABUF tuple.
<i>Dump 3</i> =	Current time.

TRNDC

Software Release: 5E10 and later

1. TRANSACTION DICTIONARY (TRNDC) AUDIT

Transactions are used by the initialization and maintenance code to read port assignments, service status, and so forth, and to manually add recent change information to the database. Transaction information is stored in the transaction dictionary. The transaction dictionary audit (TRNDC) recovers the data structures that keep track of the information associated with transactions.

The TRNDC audit is part of the data base audits subgroup. It runs in the operational kernel process (OKP) in the administrative module (AM), in the operational system for distributed switching-switching (SM) module, and in the communications module processor (CMP) environments in the segmented mode.

2. DATA STRUCTURES

The transaction dictionary array is owned by the database head cells relation (RLDB_HEADC). There are separate entries in RLDB_HEADC for idle transaction dictionary entries, busy transaction dictionary entries, cleanup transaction dictionary entries, and logging entries. Refer to Figure 1. The transaction dictionary has links to the open dictionary (RLDB_OPNDC), the database global parameter dynamic linkage array (RLDB_PARM), the database SM status array (RLDBSMSTAT), the database broadcast message dictionary (RLDBCSTMSG), the buffer data (RLDB_BFDAT), and the coupler (RLDB_CUPLR) relations. The transaction dictionary is the owner in an owner-member linkage of the open dictionary entries. It has owner-member linkages to RLDBSMSTAT and RLDBCSTMSG, depending on whether the transaction is partitioned or redundant. The SM status array contains OSDS process IDs for all SM terminal processes to handle SM-redundant transactions (SM-redundant transactions involve relations that have the same data in all the SMs). The list of process IDs contain entries for each SM terminal process to handle SM-partitioned relations (relations that exist independently in each SM). If a partitioned transaction is accessing data in more than one SM, all the entries are linked together.

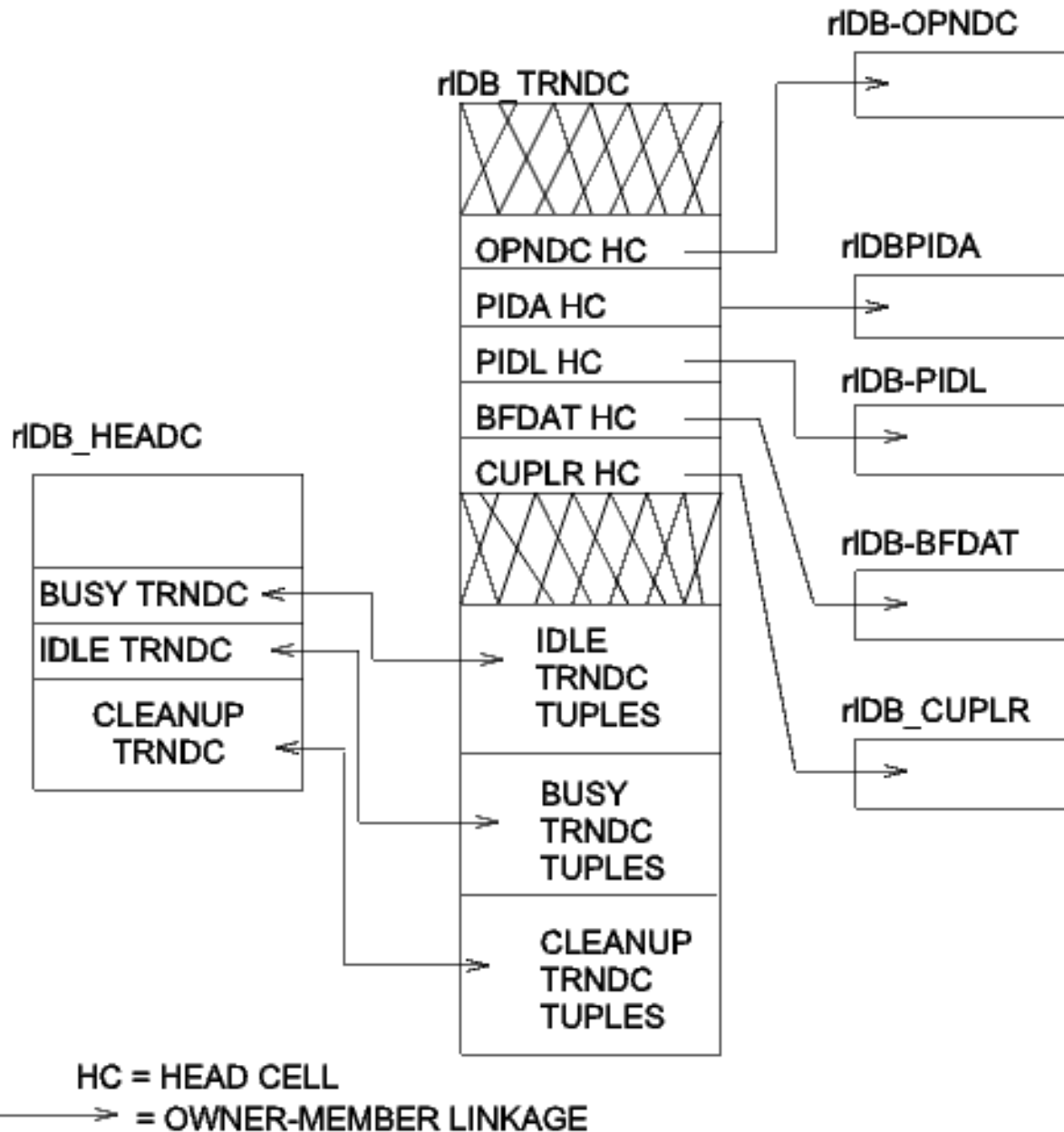


Figure 1 (TRNDC-1) Data Structures

3. AUDIT CHECKS

The transaction dictionary audit performs both syntactic and semantic checks. It calls the dynamic linkage driver (DRLINK) to check the owner-member linkages between RLDB_TRNDC and RLDB_OPNDC, RLDBSMSTAT, RLDB_BFDAT, RLDB_CUPLR, RLDB_PARM, and RLDBCSTMSG. Once the linkage checks are made, the audit performs the following semantic checks.

- (1) In the CMP, check the transaction queuing tables to make sure that the entries are all zero for idle global transactions. This is to avoid undesirable transaction queuing and dead-lock.
- (2) The audit verifies that the transaction process is active.

- (3) The audit checks the transaction time to avoid stuck or lost transactions.
- (4) During an update transaction, the range of the block ID is verified for each processor.
- (5) When the audit is checking the transaction dictionary in the AM, it checks for a valid logging process and insures a valid page count range.
- (6) When the audit is checking the transaction dictionary in the CMP, it steps through the transaction dictionary array and reads the physical distribution flag. The physical distribution flag tells whether the rIDB_TRNDC tuple represents a global physically distributed transaction and verifies the RLDB_OPNDC link.
- (7) Finally, the audit checks the owner of the transaction entry. It ensures that the owner is one of the busy or cleanup or idle or logging head cells in RLDB_HEADC.

4. ERROR HANDLING

The audit takes one of the following recovery actions. If it detects a linkage error, it aborts all active transactions. If there is a semantic error, it reports the error, aborts only the current transaction, and defers cleanup of related resources to a data base process (if possible) or reclaims the related resources (open dictionaries, the relation queue, and so forth).

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: BADLOGBID

Error Code:	BADLOGBID
Error Description:	The block ID of the local log is out-of-range.
Possible Error Effect:	This transaction may not complete successfully.
Corrective Action Taken:	The audit will purge the owning process of this transaction, idle this transaction dictionary entry, and reinitialize the entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_TRNDC tuple.
<i>Bad Data =</i>	Block ID of the local log page.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Transaction ID.
<i>Dump =</i>	rIDB_TRNDC tuple.

Table 2 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The changed page count is out-of-range.
Possible Error Effect:	This transaction may not complete successfully.
Corrective Action Taken:	The audit will purge the owning process of this transaction, idle this transaction dictionary entry, and reinitialize the entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_TRNDC tuple.
<i>Bad Data =</i>	Contents of the changed page count field.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Transaction ID.
<i>Dump =</i>	rIDB_TRNDC tuple.

Table 3 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	This rIDB_TRNDC tuple is used by the logging process which no longer exists.

Possible Error Effect:	The logging system process is not active.
Corrective Action Taken:	The logging process is purged. When the logging process is reinitialized, the rIDB_TRNDC tuple will be reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_TRNDC tuple.
<i>Bad Data =</i>	OSpid of the owning process of this transaction.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Transaction ID.
<i>Dump =</i>	rIDB_TRNDC tuple.

Table 4 Error Code: IMQNULL

Error Code:	IMQNULL
Error Description:	The trn_imq for a global transaction is NULL.
Possible Error Effect:	This transaction may not complete successfully.
Corrective Action Taken:	The audit will purge the owning process of this transaction, idle this transaction dictionary entry, and reinitialize the entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_TRNDC tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Transaction ID.
<i>Dump =</i>	rIDB_TRNDC tuple.

Table 5 Error Code: QCNTXIDLE

Error Code:	QCNTXIDLE
Error Description:	The CMP global transaction queuing count table is non-zero in (X,Y) entry, where X is an idle global transaction. It means that the idle global transaction X is queued behind the global transaction Y.
Possible Error Effect:	The non-zero entry may cause false dead-lock on global transactions.
Corrective Action Taken:	The false queuing table entry is zeroed. The whole tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_TRNDC tuple.
<i>Bad Data =</i>	The incorrect queuing count table entry.
<i>Good Data =</i>	X-INDEX of the queuing count table.
<i>Logical Index =</i>	Y-INDEX of the queuing count table.
<i>Dump =</i>	The rIDB_TRNDC tuple.

Table 6 Error Code: QCNTYIDLE

Error Code:	QCNTYIDLE
Error Description:	The CMP global transaction queuing count table is non-zero in (X,Y) entry, where Y is an idle global transaction. It means that the global transaction X is queued behind the idle global transaction Y.
Possible Error Effect:	The non-zero entry may cause false dead-lock on global transactions.
Corrective Action Taken:	The false queuing table entry is zeroed. The whole tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rIDB_TRNDC tuple.
<i>Bad Data =</i>	The incorrect queuing count table entry.
<i>Good Data =</i>	X-INDEX of the queuing count table.
<i>Logical Index =</i>	Y-INDEX of the queuing count table.
<i>Dump =</i>	The rIDB_TRNDC tuple.

Table 7 Error Code: QTABXIDLE

Error Code:	QTABXIDLE
Error Description:	The CMP global transaction queuing table entry is non-zero for the idle global transaction X, which means the idle global transaction X is queued behind other

	global transactions.
Possible Error Effect:	The non-zero entry may cause false dead-lock on global transactions.
Corrective Action Taken:	The false queuing table entry is zeroed. The whole tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_TRNDC tuple.
<i>Bad Data</i> =	The incorrect queuing table entry.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The X-INDEX to the incorrect queuing table entry.
<i>Dump</i> =	The rIDB_TRNDC tuple.

Table 8 Error Code: QTABYIDLE

Error Code:	QTABYIDLE
Error Description:	The CMP global transaction queuing table entry is non-zero in Y-th bit for the global transaction X, which means that the global transaction X is queued behind a idle global transaction Y.
Possible Error Effect:	The non-zero entry may cause false dead-lock on global transactions.
Corrective Action Taken:	The false queuing table entry is zeroed. The whole tuple is initialized and put back to idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_TRNDC tuple.
<i>Bad Data</i> =	The incorrect queuing table entry.
<i>Good Data</i> =	The mask of the failing Y-th bit in the entry.
<i>Logical Index</i> =	The X-INDEX to the incorrect queuing table entry.
<i>Dump</i> =	The rIDB_TRNDC tuple.

Table 9 Error Code: TID

Error Code:	TID
Error Description:	Some other transaction entry is linked to the logging process.
Possible Error Effect:	The logging process is using other transaction resources. This may cause other transactions to fail.
Corrective Action Taken:	The tuple is recovered and reinitialized when the logging system process is recovered in the AM. In the CMP and SM the tuple will be recovered by the TRNDC audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_TRNDC tuple.
<i>Bad Data</i> =	Transaction ID in rIDB_TRNDC tuple for logging.
<i>Good Data</i> =	Logging transaction ID.
<i>Logical Index</i> =	Logical key of tuple.
<i>Dump</i> =	The rIDB_TRNDC tuple.

Table 10 Error Code: TIMEOUT

Error Code:	TIMEOUT
Error Description:	The transaction has exceeded its timeout by 5 minutes. In the CMP this error will not be reported during a soft switch.
Possible Error Effect:	Other transactions may not be able to run.
Corrective Action Taken:	The audit will purge the owning process of this transaction, idle this transaction dictionary entry, and reinitialize the entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_TRNDC tuple.
<i>Bad Data</i> =	Maximum duration time for this transaction.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	rIDB_TRNDC tuple.

Table 11 Error Code: TPID

Error Code:	TPID

Error Description:	The OSpid of the owning process is invalid.
Possible Error Effect:	This transaction may not complete successfully.
Corrective Action Taken:	If this is the logging process, the process is purged and the tuple reinitialized. For all other rIDB_TRNDC entries, the tuple is put on the DB deferred clean up list if possible. Otherwise the transaction entry is put back to the idle list, and then reinitialized.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rIDB_TRNDC.
<i>Bad Data</i> =	OSpid of the owning process of this transaction.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	rIDB_TRNDC tuple.

Table 12 Error Code: TQNULL

Error Code:	TQNULL
Error Description:	The transaction dictionary is neither on the busy list nor on the idle list.
Possible Error Effect:	This transaction dictionary entry is lost.
Corrective Action Taken:	The audit will purge the owning process of this transaction, idle this transaction dictionary entry, and reinitialize the entry.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rIDB_TRNDC entry.
<i>Bad Data</i> =	Physical key of the owner rIDB_HEADC.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Transaction ID.
<i>Dump</i> =	rIDB_TRNDC tuple.

TSDB-A

Software Release: 5E10 - 5E11

1. TIME SLOT DATA BLOCK (TSDB) AUDIT

Time available on the link from the switching module (SM) to the administrative module (AM) is divided into time slots. Each time slot carries a voice signal, which is composed of 16 bit pulses. Eight of the bits contain the encoding of the voice signal and eight bits are used for signaling. One of these eight signaling bits, the E signaling bit, confirms the connection between the originating and terminating processes. The possible bit values are continuity and discontinuity. E-bit continuity indicates a stable connection, which guarantees that when a process in one SM terminates, so does the corresponding process in the other SM. The state of the time slot data block reflects the state of the time slot. The time slot data block audit (TSDB) verifies the consistency of the time slot data block relation (RLTSDB), which represents the network side channel of the network.

This audit is part of the peripheral control audits subgroup. It operates in the OSDS-M environment in the segmented mode.

2. DATA STRUCTURES

This audit operates on the linkage areas of the time slot data block relation (RLTSDB). The RLTSDB relation has both a one-to-one linkage and a member linkage with the path data block relation (RLPHDB).

The following data structure is used for SM-2000 processors only:

`ccrgm.st_link` = A field in the RLCCBCOM relation representing a one-to-one linkage with the RLCHDB relation.

Figure 1 illustrates the RLTSDB linkages. When this audit operates in a remote switch module (RSM) environment, it also verifies the state of the PCmpttra (intramodule), PCmptertins (intermodule, in service), and PCmptertroos (intermodule, out of service) time slot allocation bit maps.

3. AUDIT CHECKS

To verify the linkages associated with the time slot data blocks, the TSDB audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the driver invokes the TSDB semantic check function to perform the following checks:

- (a) A time slot associated with an idle time slot data block [except RSM control time slots in host switching modules (HSMs)] should be receiving from the idle code channel (silence) or cross talk will be picked up. The audit uses the time slot data block linkages to the path data block to distinguish between idle and stable states. It checks the hardware to determine which channel the time slot is connected to.
- (b) The audit checks the time slot data block linkages to the path data block. The linkages for a time slot data block in a stable state should be error-free; if they are not, a bad linkage may cause an assert. An idle time slot data block should have a zero linkage in its one-to-one linkage block; if it does not, the time slot cannot be used by call processing.
- (c) A time slot data block should not be in a transient state for more than five minutes or a call in progress will not be able to disconnect. The audit times the state of the time slot data block for a change from the transient state.
- (d) If the time slot data block is used in a stable intermodule call, the audit checks for discontinuity. If discontinuity is found, the audit prints an error message.

- (e) The audit verifies the consistency of the RLTSDB relation with the three time slot allocation maps. Table 1 outlines the information that should exist in the allocation maps.

Table 1 Time Slot Allocation Maps

Time Slot Status (rtsstat)	Time Slot Type (rtstype)	On Idle List	Pcmptra	PCmpterins	PCmpteroos
PCTSIDLE	PCTRA	yes	1	0	0
PCTSIDLE	PCTERINS	yes	0	1	0
PCTSIDLE	PCTEROOS	yes	0	0	1
PCTSBUSY	any	no	0	0	0
PCTSALLOC	any	yes	0	0	0
PCTSPREMT	any	yes	0	0	0
PCTSNLDUP	PCTRA	yes	0	0	0

4. ERROR HANDLING

Any tuples that fail these checks are reinitialized and placed on the idle list. All time slots associated with idle time slot data blocks are connected to a source of silence.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 2 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	(a) When no rICBCOM tuple is dumped: when stepping through all tuples in the RLTSDB relation to count the number of busy rITSDB tuples, a rITSDB tuple was found to be linked to a rIPHDB tuple with a mutilated key field (rIPHDB.phdbnum). (b) When rICBCOM tuple is dumped: when stepping through all tuples in the RLTSDB relation associated with peripheral time slot data blocks (TSDBs), to count the number of busy peripheral rITSDB tuples, an rITSDB tuple was found to be linked to a rICBCOM tuple with a mutilated key field (rICBCOM.ccbnum).
Possible Error Effect:	This may cause asserts to be fired by the call processing.
Corrective Action Taken:	The PHDB audit is scheduled as a related audit to detect and recover the key error in the rIPHDB tuple.
Manual Action Required:	None.
Dump Description:	
Error Address =	Address of the rICBCOM tuple.
Bad Data =	Valid values are: (a) Logical key of the rIPHDB tuple (rIPHDB.phdbnum). (b) Logical key of the rICBCOM tuple (rICBCOM.ccbnum).
Good Data =	Valid values are: (a) Correct value of the rIPHDB logical key. (b) Correct value of the rICBCOM logical key.
Logical Index =	Valid values are:

	(a) NA (0).
	(b) rITSDB key or NA (0).
<i>Dump 1 =</i>	rITSDB tuple.
<i>Dump 2 =</i>	rIPHDB tuple.
<i>Dump 3 =</i>	rICBCOM tuple.

Table 3 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	A rITSDB tuple was found which is not a network TSDB tuple. The key (rITSDB.tsnun) is outside the range of network side TSDB keys.
Possible Error Effect:	This may cause asserts to be fired by the call processing.
Corrective Action Taken:	The audit deletes this rITSDB tuple from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPHDB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rITSDB key (rITSDB.tsnun).
<i>Dump =</i>	The rITSDB tuple.

Table 4 Error Code: CKTDSMST

Error Code:	CKTDSMST
Error Description:	Invalid DB return when accessing the RLSMEST relation with a T1 time slot name which may indicate an ODD problem.
Possible Error Effect:	Data on the status of the time slot cannot be accessed.
Corrective Action Taken:	The tuple is assumed to be correct.
Manual Action Required:	If error repeats, investigate the ODD. The problem may be in the RLCKTDATA, RLICL_MSK or RLRTS_HCN relation. If the problem is with the RLICL_MSK it could be that the corresponding bit in this relation should be set to null. The bit position is driven by taking the tsnum field of the tsdb tuple, right shifting it by 1 and bitwise anding it with 0x1f.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLSMEST relation.
<i>Dump =</i>	rISMEST buffer.

Table 5 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	The global variable PCnts_usage is found to be inconsistent with the total count of network rITSDB tuples that are currently busy. PCnts_usage is used to report network time slot usage on various traffic reports.
Possible Error Effect:	If PCnts_usage is incorrect the traffic reports will not be accurate which may cause the SM to be under/over engineered.
Corrective Action Taken:	The global variable PCnts_usage is updated to contain the total number of busy network rITSDB tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCnts_usage.
<i>Bad Data =</i>	Contents of PCnts_usage.
<i>Good Data =</i>	Total number of busy network rITSDB tuples.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 6 Error Code: CRAM_E_UPPERBITS

Error Code:	CRAM_E_UPPERBITS
Error Description:	The value in the upper eight bits of control RAM E (CRAM-E) is invalid for the

	current set of options enabled for this TSIU model 3.
Possible Error Effect:	Unpredictable effects can occur when an option is enabled and the associated (and needed) hardware is not equipped.
Corrective Action Taken:	The correct value of the options covered in this audit has been written into the upper portion of CRAM-E.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the faulty TSDB tuple.
<i>Bad Data =</i>	A composite of the errors found during the check of the upper 8 bits of CRAME-E. Any bit=1 is in error.
<i>Good Data =</i>	The value of the global TSIU3 option indicator, Gltsiopt. NOTE: This register uses negative logic. 0 = Option ENABLED. 1 = Option DISABLED.
<i>Logical Index =</i>	The key of the faulty TSDB tuple.
<i>Dump 1 =</i>	The faulty TSDB tuple.
<i>Dump 2 =</i>	The entire structure of type tsNTSCONFG.

Table 7 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	A stable call is not scanning for discontinuity.
Possible Error Effect:	The call process may be stuck, since it will never see the discontinuity and tear itself down.
Corrective Action Taken:	The audit initializes the rTSDB tuple and places it on the idle list. As part of initialization, the audit zeros the linkage block "to" pointer. The rTSDB is connected to channel zero (silence), and the time slot is set to scan for discontinuity.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTSDB tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the rTSDB tuple.
<i>Dump 1 =</i>	rTSDB tuple.
<i>Dump 2 =</i>	PHDB tuple.

Table 8 Error Code: E_BIT_IN

Error Code:	E_BIT_IN
Error Description:	This audit error can be the result of one of five causes: (1) Timing anomaly - no action needed. (2) Hard switch of TMS - no action needed. (3) Out of date TMS fabric - remove and restore ONTCs. (4) Faulty hardware - check hardware. (5) Bad ODD (RSM only) - check ODD. The first two causes will not cause persistent audit failures. If this audit error persists check whether the error is being reported from an RSM or not. If it is from an RSM, the audit error could be an ODD problem or a hardware problem. If the audit is reported from an SM that is NOT an RSM, it is a result of faulty hardware or an out of date TMS fabric. The KEY to the TSDB relation represents the network side time slot number of the TSI. The KEY, time slot, should be converted to its decimal equivalent when used to verify the

ODD and hardware.

A. Non-persistent audit failures - no action needed.

If this audit error is generated, but clears itself up, no user action is needed. It is assumed to be caused by either a timing anomaly or a hard switch of the TMS.

1. Timing anomaly - no action needed.

If this audit error is generated on one TSDB as an isolated occurrence, this can be treated as a timing anomaly caused by an abnormally long delay in sending a message over the C-links. No action need be taken.

2. Hard switch of TMS links - no action needed.

If this audit error is generated after the hard switch of the TMS links, the error is probably the result of the TMS copy being made active when its fabric was not updated to the current state of connections. This condition will be automatically corrected by a TMS fabric update, so audit errors should cease in a few minutes. The exact amount of time depends on the delay incurred in getting the audit reports to the ROP.

B. Persistent audit failures.

1. Non-RSM audit failures.

If this audit error is reported from an SM that is connected directly to the TMS (not an RSM), it is the result of either an out of date TMS fabric or a faulty DLI.

a. Out of date TMS fabric - remove and restore the ONTC.

The TMS fabric could be out of date on one or both sides because of losing connect orders from the SMs or any failure in updating the TMS fabric during an ONTC restoration. This can be corrected by conditionally removing and restoring both ONTC sides. The restoration should cause the fabric to be brought up to date.

If a conditional removal is denied due to "AUTISS", switch ONTCs and try to conditionally remove the other side. If this is also denied, you are in a degraded mode in the ONTC. This means that there is

faulty equipment in the active major side of the ONTC causing "AUTISS" to be active. This should be corrected as soon as possible. Do not disable the TSDB audit to eliminate the output, because it may be needed to recover from any unassociated software problems in the TSDB data.

b. Faulty hardware - check the hardware.

If this audit error is reported from an SM that is connected directly to the TMS (not an RSM), it could be the result of a faulty DLI anywhere on the switch sending E-bit on the minor ONTC side. When E-bit is seen on this SM, AUTISS would send this through so that the audit would see it. If the ONTC is in a duplex condition, put the minor ONTC out of service and diagnose it. This should resolve the problem.

If this audit error is reported from an RSM, use the KEY and the RLRTS_RTS and RLRTS_HCN relations to find the SM number of the far end of the facility. Wait until traffic permits the removal of one of the facilities. Use the DLTU page (112x) on the MCC to locate all the facilities connecting this RSM to the given far end. One at a time, go through each facility, putting its DFI out of service. After each DFI is put out of service, run the TSDB audit to see whether the audit error is still reported. It will not be reported on out of service time slots. When the correct DFI is found, diagnose it and the associated DFI at the other end to identify the faulty hardware. If the diagnostics pass, the problem may be in the facility.

2. RSM audit failures.

If this audit error is repeatedly reported from an RSM, the problem may be either faulty hardware or bad ODD.

a. Faulty hardware - check the hardware.

If this audit error is reported from an RSM, use the KEY and the RLRTS_RTS and RLRTS_HCN relations to find the SM number of the far end of the facility. Wait until traffic permits

the removal of one of the facilities. Use the DLTU page (112x) on the MCC to locate all the facilities connecting this RSM to the given far end. One at a time, go through each facility, putting its DFI out of service. After each DFI is put out of service, run the TSDB audit to see whether the audit error is still reported. It will not be reported on out of service time slots. When the correct DFI is found, diagnose it and the associated DFI at the other end to identify the faulty hardware. If the diagnostics pass, the problem may be in the facility.

b. Bad ODD - check the ODD.

If this audit error is reported from an RSM, the error could be a result of one or more inconsistencies among the various relations describing the links on this RSM. Using the KEY, check the RLRTS_RTS and RLRTS_HCN relations to determine whether the time slot is on a link to the HSM or to another RSM in the cluster.

If the error from the RSM is on a time slot linked to the HSM, check that the time slot appearance in RLRTS_HCN agrees with RLHCN_RTS. Also check the RLICL_MSK relation in the RSM. The tuple for the HSM should be checked at the bit position corresponding to the time slot. This position should be set to one. Refer to the Audits Overview section for the interpretation of the RLICL_MSK relation. Finally, use RLRTS_HCN to get the himckname for the time slot. Use the himckname as the key into RLCKTDATA at the HSM. Check that the circuit has a type of PCT1_HIM.

If the link is to another RSM in the cluster, compare the RLRTS_RTS and RLICL_MSK relations in the two RSMs. This is done by first checking the RLICL_MSK relation in the RSM that reported the audit error. Examine the tuple corresponding to the other RSM. The bit position corresponding to the time slot should be set to one. Next, compare the RLRTS_RTS relation in the two RSMs to see that they agree on the correlation between the time slot names in the two RSMs and the RSM number at the other end. Finally, check the RLICL_MSK relation at the RSM at the other end of the facility. Using

the time slot name from RLRTS_RTS to check the bit position as was done at the RSM where the audit failed.

3. Reading the RLICL_MSK relation in an RSM.

The RLICL_MSK relation has the following fields: mod, msk01, msk02, msk03, ... msk16. The mod field is the key and signifies the SM number of the RSM or HSM that connects to this RSM.

RLICL_MSK always contains tuples that have a mod field value of zero and the this RSM. These are special tuples that will not be described here.

The msk01-msk16 fields contain an array of 32 bits represented in the ODD by 8 hexadecimal digits. Each bit corresponds to a time slot. Msk01-msk08 contain all the even numbered time slots. Msk09-msk16 contain all the odd numbered time slot. Msk01 contains the even numbered time slots from 0 to 62. Msk02 contains the even numbered time slots from 64 to 126. Msk09 contains the odd numbered time slots from 1 to 63. The following table shows what time slots are contained in each msk?? field.

msk01	- even time slots	0- 62
msk02	- even time slots	64-126
msk03	- even time slots	128-190
msk04	- even time slots	192-254
msk05	- even time slots	256-318
msk06	- even time slots	320-392
msk07	- even time slots	394-456
msk08	- even time slots	458-510
msk09	- odd time slots	1- 63
msk10	- odd time slots	65-127
msk11	- odd time slots	129-191
msk12	- odd time slots	193-255
msk13	- odd time slots	257-319
msk14	- odd time slots	321-393
msk15	- odd time slots	395-457
msk16	- odd time slots	459-511

To find the position of a particular time slot, use the table to find the corresponding field. Read the value of that field for the desired tuple, and expand it from hexadecimal form to binary form. Then one can easily check the value corresponding to the desired time slot. As an example, if you wanted to examine the value of time slot 291 going to RSM 12, you would

	check the table to find that the status of time slot 291 is found in the msk13 field. Let's say that the value for the msk13 field in the tuple for mod 12 is: 20004040 When expanded to binary, this hex value becomes: 0010 0000 0000 0000 0100 0000 0100 0000 By writing out these values next to the corresponding time slots we can easily see the value for time slot 291. 257-259-261-263 <--> 0010 265-267-269-271 <--> 0000 273-275-277-279 <--> 0000 281-283-285-287 <--> 0000 289-291-293-295 <--> 0100 297-299-301-303 <--> 0000 395-307-309-311 <--> 0100 313-315-317-319 <--> 0000 Time slot 291 has a value of 1. So, according to RLICL_MSK, time slot 291 is connected to a T1 facility going to RSM 12.
Possible Error Effect:	See the Error Description.
Corrective Action Taken:	None. This is a manual action required error.
Manual Action Required:	Investigate why we are not receiving the correct E bit status for the give idle time slot.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	The "e_bit_in" field from the TSntsconfig.
<i>Good Data =</i>	The idle state value (ID_BIT).
<i>Logical Index =</i>	Logical key of the rITSDB tuple.
<i>Dump =</i>	NA.

Table 9 Error Code: FIDB_SGN

Error Code:	FIDB_SGN
Error Description:	The audit attempted to obtain a remote inter-module time-slot (RTERTS) circuit name from a time slot on a remote SM, but was not successful.
Possible Error Effect:	Possible error in time slot usage on the SM.
Corrective Action Taken:	None.
Manual Action Required:	Either the UNIT_CKT or FIDB_SGN tuple is missing for this given RTERTS circuit name derived from the time slot number. The given time slot number is listed in the good data of the audit dump. Insert the missing UNIT_CKT or FIDB_SGN tuple for this corresponding time slot using recent change or ODBE.
Dump Description:	
<i>Error Address =</i>	Address of the buffer to hold the corresponding ICL_MSK tuple.
<i>Bad Data =</i>	Return code GLNULCKT.
<i>Good Data =</i>	The value of the time slot.

<i>Logical Index</i> =	NA
<i>Dump</i> =	The rIICL MSK tuple.

Table 10 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An idle peripheral time slot data block tuple (rITSDB) is linked to an path data block tuple (rPHDB) head cell for network TSDBs, or a network rITSDB tuple is in a rPHDB head cell for peripheral rITSDBs.
Possible Error Effect:	This may cause asserts to be fired by call processing.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	Logical key of the rPHDB tuple (rPHDN.phdbnum).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITSDB tuple.
<i>Dump 1</i> =	The rITSDB tuple.
<i>Dump 2</i> =	The rPHDB tuple.

Table 11 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	An idle rITSDB tuple's associated time slot is not receiving from the idle channel (Gldlechl) or other TSI hardware states are not idle. Check the dump of the structure to determine which state(s) is(are) bad.
Possible Error Effect:	Excessive noise may be picked up (that is, crosstalk).
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list. As part of initialization, the audit forces the time slot to receive from channel Gldlechl. Also, set all other related entries to their idle states.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB tuple.
<i>Bad Data</i> =	Channel from which the time slot is receiving.
<i>Good Data</i> =	Gldlechl.
<i>Logical Index</i> =	Logical key of the rITSDB tuple and time slot.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	The pc_nts_cofg structure.
<i>Dump 3</i> =	PHDB tuple.

Table 12 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The remote time slot type attribute (rITSDB.rstype) of the rITSDB tuple is not marked as in service (PCINS), or the corresponding bit in the out-of-service (OOS) time slot map (PCmpoos) is not set.
Possible Error Effect:	The improper type of time slot could be allocated to a call.
Corrective Action Taken:	The audit reinitializes the rITSDB tuple and it is linked to the idle list. The corresponding bit in the OOS time slot map (PCmpoos) is set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	Content of the OOS time slot map (PCmpoos).
<i>Good Data</i> =	Content of rITSDB.rstype.
<i>Logical Index</i> =	Logical key of the rITSDB tuple.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	Owning rPHDB tuple.

Table 13 Error Code: NAILEDUP

Error Code:	NAILEDUP
Error Description:	The <i>rtssstat</i> field of the rITSDB tuple in an RSM is marked as nailed up and it should not be, or it is marked as not nailed up but should be.
Possible Error Effect:	The time slot may not be correctly allocated.

Corrective Action Taken:	The audit initializes the rITSDB tuple. As part of the initialization the rtsstat field is set correctly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITSDB tuple.
<i>Bad Data =</i>	rtsstat field of rITSDB tuple.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rITSDB tuple.
<i>Dump 1 =</i>	rITSDB tuple.
<i>Dump 2 =</i>	PHDB tuple.

Table 14 Error Code: NO_OWN_TS

Error Code:	NO_OWN_TS
Error Description:	The rtstype field of an RSM rITSDB tuple and its linkage or the time slot allocation maps do not agree.
Possible Error Effect:	Time slots will not be correctly allocated.
Corrective Action Taken:	The audit initializes the rITSDB tuple. As part of the initialization the rtsstat and rtstype fields are set correctly and the allocation maps marked accordingly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rITSDB tuple.
<i>Bad Data =</i>	Bit 0 is state of out-of-service map (idle = 1; busy = 0). Bit 1 is state of in-service map.
<i>Good Data =</i>	rtsstat field of rITSDB tuple.
<i>Logical Index =</i>	Logical key of rITSDB tuple.
<i>Dump =</i>	rITSDB tuple.

Table 15 Error Code: PATHSTT

Error Code:	PATHSTT
Error Description:	The path state of the rIPHDB owning a rITSDB is invalid.
Possible Error Effect:	Asserts by call processing.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list. As part of initialization, the audit zeros the linkage block "to" pointer. The rITSDB is connected to channel zero (silence), and the time slot is set to scan for discontinuity.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	Path state of the owning PHDB.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the rITSDB tuple.
<i>Dump 1 =</i>	rITSDB tuple.
<i>Dump 2 =</i>	Owner rIPHDB tuple.

Table 16 Error Code: RSMNTS

Error Code:	RSMNTS
Error Description:	Invalid DB return when accessing RLRSMTS relation, possible ODD problem. This error will continue to print out until fixed manually.
Possible Error Effect:	Relation can not be accessed.
Corrective Action Taken:	None.
Manual Action Required:	If error repeats, investigate the ODD data.
Dump Description:	
<i>Error Address =</i>	Address of the rITSDB tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLRSMTS relation.
<i>Dump =</i>	Local buffer used to access rRSMNTS tuple.

Table 17 Error Code: RTS_HCN

Error Code:	RTS_HCN

Error Description:	Invalid DB return when accessing the RLRTS_HCN relation which may indicate an ODD problem.
Possible Error Effect:	RSM time slot number to HSM DFI channel circuit name translation cannot be made.
Corrective Action Taken:	The tuple is assumed to be correct.
Manual Action Required:	If error repeats, investigate the ODD.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	Bad DB return.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Key used to access the RLRTS_HCN relation.
<i>Dump</i> =	rlRTS_HCN buffer.

Table 18 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	An rITSDB tuple in a stable state has bad linkage.
Possible Error Effect:	The bad linkage may cause an ASSERT before the audit finds the error.
Corrective Action Taken:	The audit idles the rITSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	rITSDB tuple address.
<i>Bad Data</i> =	Contents of linkage.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Physical key of the rITSDB tuple.
<i>Dump</i> =	rITSDB tuple.

Table 19 Error Code: STATUS

Error Code:	STATUS
Error Description:	The time slot is idle or associated with a stable call but the time slot job map associated with immediate audible does not indicate that this time slot is idle.
Possible Error Effect:	This time slot could be connected to audible ring.
Corrective Action Taken:	PCcnaud() is called to idle AR job.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB tuple.
<i>Bad Data</i> =	The state field from the PCtsdbjbf[] table.
<i>Good Data</i> =	The idle state value (PCARIDLE).
<i>Logical Index</i> =	Key used to access the RLTSDB relation.
<i>Dump</i> =	rITSDB tuple.

Table 20 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rITSDB tuple has been in the transient state for more than 5 minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rITSDB tuple is zeroed. The call is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB tuple.
<i>Bad Data</i> =	Valid values are: 0 = If peripheral control is not scanning for E-bit. 1 = If peripheral control is scanning for E-bit. 2 = Not applicable.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of rITSDB tuple.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	PHDB tuple.

Table 21 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	A rITSDB tuple's associated time slot is stuck in some invalid state. The tuple will be corrected and returned to the appropriate idle list.
Possible Error Effect:	A time slot may not be used for call processing.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB tuple.
<i>Bad Data</i> =	0.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of rITSDB tuple and time slot.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	rPHDB tuple.

Table 22 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The peripheral rITSDB tuple has been in the transient state for more than five minutes.
Possible Error Effect:	A peripheral to peripheral call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rITSDB tuple is zeroed. The call is terminated. The rITSDB tuple is placed in rPHDB headcell for idle peripheral rITSDB tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITSDB tuple.
<i>Dump</i> =	The rITSDB tuple.

Table 23 Error Code: TSMAP1

Error Code:	TSMAP1
Error Description:	The rtsstat field of an RSM rITSDB tuple and its linkage or the time slot allocation maps do not agree.
Possible Error Effect:	Time slots will not be correctly allocated.
Corrective Action Taken:	The audit initializes the rITSDB tuple. As part of the initialization the rtsstat field is set correctly and the allocation maps marked accordingly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	Bit 0 is state of INTER/OUT OF SERVICE map (idle = 1; busy = 0). Bit 1 is state of INTER/IN SERVICE map. Bit 2 is state of INTRA map.
<i>Good Data</i> =	rtsstat field of rITSDB tuple.
<i>Logical Index</i> =	Logical key of rITSDB tuple.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	PHDB tuple.

Table 24 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	For network rITSDBs, the index number in this description corresponds to the KEY value. So based on KEY, one of the following apply: (1) The time slot interchanger (TSI) time slot attribute (rITSDB.tsits) of the rITSDB tuple does not contain the proper TSI time slot number. (2) For peripheral rITSDBs, the TSI time slot attribute (rITSDB.tsits) is not equal to value PCIDLETN for a virtual path (rICCB.COM.lptype =

	PCDPN_VIRT).
	(3) For peripheral rITSDBs, the TSI time slot attribute (rITSDB.tsits) is not equal to the logical key of rICHDB (derived from rCCBCOM.ccrgm.st_link).
Possible Error Effect:	(1) This can cause peripheral control (PC) to write to the wrong TSI control RAMs, causing paths through the TSI to be opened erroneously. (2) May cause the rITSDB tuple to be not usable. (3) This can cause PC to write to the wrong TSI control RAMs, causing paths through the TSI to be opened erroneously.
Corrective Action Taken:	1. The rITSDB has been reinitialized and linked to the idle list. The TSI control RAMs associated with this rITSDB tuple have been reinitialized. 3. The rITSDB has been reinitialized and linked to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	Contents of the TSI time slot attribute (rITSDB.tsits).
<i>Good Data</i> =	Correct value for rITSDB.tsits.
<i>Logical Index</i> =	1, 2, or 3.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	rPHDB tuple (for KEY = 2 or 3).
<i>Dump 3</i> =	rCCBCOM tuple (for KEY = 2 or 3).

Table 25 Error Code: TYPE

Error Code:	TYPE
Error Description:	The rtstype field of the rITSDB tuple disagrees with the actual type of the RSM time slot (INTRA, INTER/IN SERVICE, or INTER/OUT OF SERVICE).
Possible Error Effect:	The improper type of RSM time slot could be allocated to a call.
Corrective Action Taken:	The audit initializes the rITSDB tuple. As part of the initialization the rtstype field is set correctly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	rtstype field of rITSDB tuple.
<i>Good Data</i> =	The correct time slot type.
<i>Logical Index</i> =	Logical key of rITSDB tuple.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	PHDB tuple.

Table 26 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rITSDB tuple has nonzero linkage in its one-to-one linkage block ("to" pointer).
Possible Error Effect:	The time slot is unusable.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list. As part of initialization, the audit zeros the linkage block "to" pointer.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB tuple.
<i>Bad Data</i> =	Linkage contents ("to" pointer).
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Logical key of the rITSDB tuple.

<i>Dump 1</i> =	riTSDB tuple.
<i>Dump 2</i> =	riPHDB tuple.

TSDB-B**Software Release:** 5E12 and later**1. TIME SLOT DATA BLOCK (TSDB) AUDIT**

Time available on the link from the switching module (SM) to the administrative module (AM) is divided into time slots. Each time slot carries a voice signal, which is composed of 16 bit pulses. Eight of the bits contain the encoding of the voice signal and eight bits are used for signaling. One of these eight signaling bits, the E signaling bit, confirms the connection between the originating and terminating processes. The possible bit values are continuity and discontinuity. E-bit continuity indicates a stable connection, which guarantees that when a process in one SM terminates, so does the corresponding process in the other SM. The state of the time slot data block reflects the state of the time slot. The time slot data block audit (TSDB) verifies the consistency of the time slot data block relation (RLTSDB), which represents the network side channel of the network.

This audit is part of the peripheral control audits subgroup. It operates in the OSDS-M environment in the segmented mode.

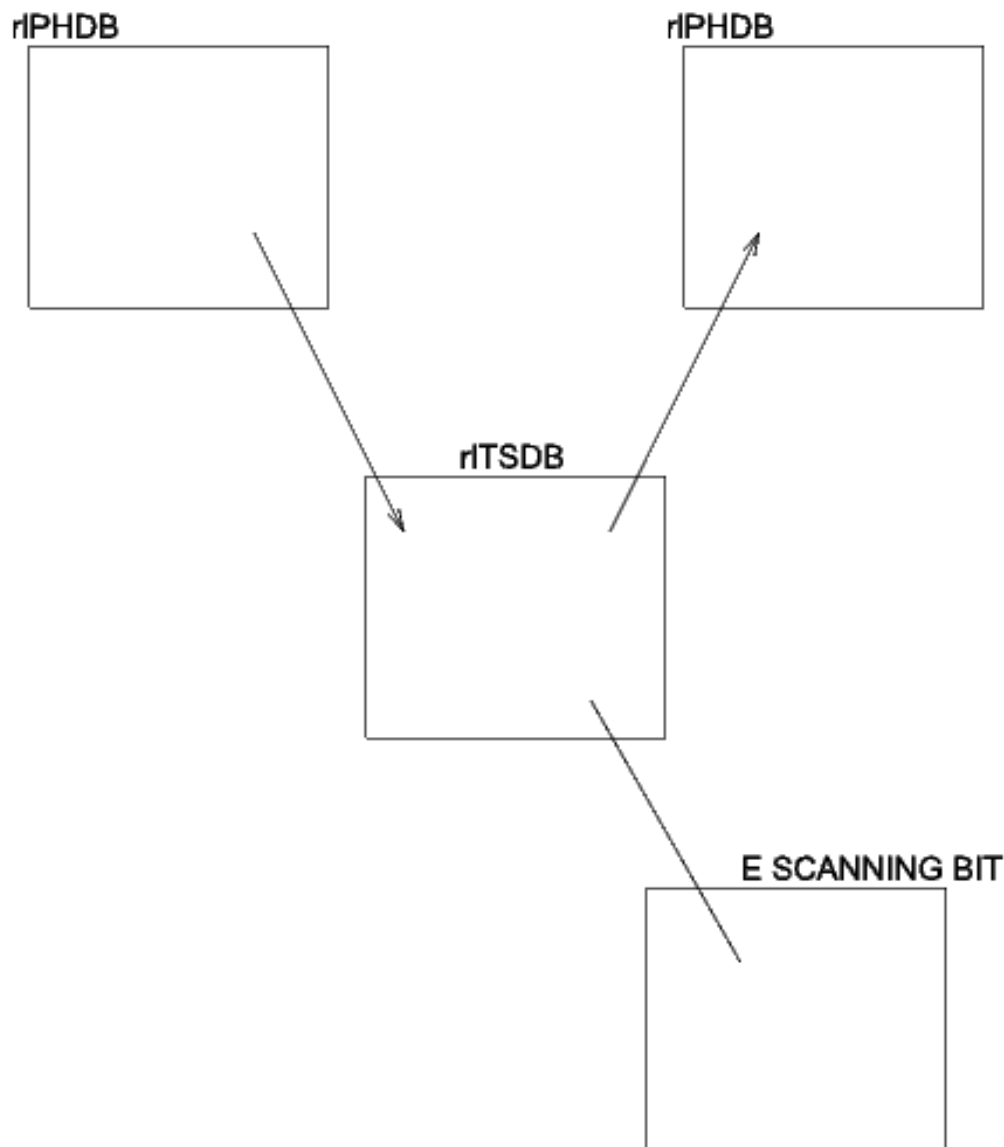
2. DATA STRUCTURES

This audit operates on the linkage areas of the time slot data block relation (RLTSDB). The RLTSDB relation has both a one-to-one linkage and a member linkage with the path data block relation (RLPHDB).

The following data structure is used for SM-2000 processors only:

ccrgm.st_link = A field in the RLCCBCOM relation representing a one-to-one linkage with the RLCHDB relation.

Figure 1 illustrates the RLTSDB linkages. When this audit operates in a remote switch module (RSM) environment, it also verifies the state of the PCmptra (intramodule), PCmpterins (intermodule, in service), and PCmpteroos (intermodule, out of service) time slot allocation bit maps.



SM
AM

 = ONE-TO-ONE LINKAGE
 = OWNER-MEMBER LINKAGE
 = NONSTANDARD LINKAGE

Figure 1 (TSDB-1) rTSDB Linkages

3. AUDIT CHECKS

To verify the linkages associated with the time slot data blocks, the TSDB audit calls the linkage driver (DRLINK) once for each audit segment. After the linkages are verified, the driver invokes the TSDB semantic check function to perform the following checks:

- (a) A time slot associated with an idle time slot data block [except RSM control time slots in host switching

modules (HSMs)] should be receiving from the idle code channel (silence) or cross talk will be picked up. The audit uses the time slot data block linkages to the path data block to distinguish between idle and stable states. It checks the hardware to determine which channel the time slot is connected to.

- (b) The audit checks the time slot data block linkages to the path data block. The linkages for a time slot data block in a stable state should be error-free; if they are not, a bad linkage may cause an assert. An idle time slot data block should have a zero linkage in its one-to-one linkage block; if it does not, the time slot cannot be used by call processing.
- (c) A time slot data block should not be in a transient state for more than five minutes or a call in progress will not be able to disconnect. The audit times the state of the time slot data block for a change from the transient state.
- (d) If the time slot data block is used in a stable intermodule call, the audit checks for discontinuity. If discontinuity is found, the audit prints an error message.
- (e) The audit verifies the consistency of the RLTSDB relation with the three time slot allocation maps. 1 outlines the information that should exist in the allocation maps.

Table 1 Time Slot Allocation Maps

Time Slot Status (rtsstat)	Time Slot Type (rtstype)	On Idle List	Pcmpttra	PCmptersins	PCmpteroos
PCTSIDLE	PCTRA	yes	1	0	0
PCTSIDLE	PCTERINS	yes	0	1	0
PCTSIDLE	PCTEROOS	yes	0	0	1
PCTSBUSY	any	no	0	0	0
PCTSALLOC	any	yes	0	0	0
PCTSPREMT	any	yes	0	0	0
PCTSNLDUP	PCTRA	yes	0	0	0

4. ERROR HANDLING

Any tuples that fail these checks are reinitialized and placed on the idle list. All time slots associated with idle time slot data blocks are connected to a source of silence.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found here.

The audit produces the following error code(s).

Table 2 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	Error is due to one of the following: 1 = When rICBCOM tuple is dumped: when stepping through all tuples in the RLTSDB relation to count the number of busy rITSDB tuples, a rITSDB tuple was found to be linked to a rIPHDB tuple with a mutilated key field (rIPHDB.phdbnum). 2 = When rICBCOM tuple is dumped: when stepping through all tuples in the RLTSDB relation associated with peripheral time slot data blocks (TSDBs), to count the number of busy peripheral rITSDB tuples, an rITSDB tuple was found to be linked to a rICBCOM tuple with a mutilated key field (rICBCOM.ccbnum).
Possible Error Effect:	This may cause asserts to be fired by the call processing.

Corrective Action Taken:	The PHDB audit is scheduled as a related audit to detect and recover the key error in the rIPHDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICBCOM tuple.
<i>Bad Data =</i>	Corresponds to the Error Description as follows: 1 = Logical key of the rIPHDB tuple (rIPHDB.phdbnum). 2 = Logical key of the rICBCOM tuple (rICBCOM.ccbnum).
<i>Good Data =</i>	Corresponds to the Error Description as follows: 1 = Correct value of the rIPHDB logical key. 2 = Correct value of the rICBCOM logical key.
<i>Logical Index =</i>	Corresponds to the Error Description as follows: 1 = NA (0). 2 = rITSDB key or NA(0).
<i>Dump 1 =</i>	rITSDB tuple.
<i>Dump 2 =</i>	rIPHDB tuple.
<i>Dump 3 =</i>	rICBCOM tuple.

Table 3 Error Code: BADSTATE

Error Code:	BADSTATE
Error Description:	A rITSDB tuple was found which is not a network TSDB tuple. The key (rITSDB.tsnun) is outside the range of network side TSDB keys.
Possible Error Effect:	This may cause asserts to be fired by the call processing.
Corrective Action Taken:	The audit deletes this rITSDB tuple from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rIPHDB tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The rITSDB key (rITSDB.tsnun).
<i>Dump =</i>	The rITSDB tuple.

Table 4 Error Code: CKTDSMST

Error Code:	CKTDSMST
Error Description:	Invalid DB return when accessing the RLSMEST relation with a T1 time slot name which may indicate an ODD problem.
Possible Error Effect:	Data on the status of the time slot cannot be accessed.
Corrective Action Taken:	The tuple is assumed to be correct.
Manual Action Required:	If error repeats, investigate the ODD. The problem may be in the RLCKTDATA, RLICL_MSK or RLRTS_HCN relation. If the problem is with the RLICL_MSK it could be that the corresponding bit in this relation should be set to null. The bit position is driven by taking the tsnum field of the tsdb tuple, right shifting it by 1 and bitwise anding it with 0x1f.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLSMEST relation.
<i>Dump =</i>	rSMEST buffer.

Table 5 Error Code: COUNTS

Error Code:	COUNTS

Error Description:	The global variable PCnts_usage is found to be inconsistent with the total count of network rITSDB tuples that are currently busy. PCnts_usage is used to report network time slot usage on various traffic reports.
Possible Error Effect:	If PCnts_usage is incorrect the traffic reports will not be accurate which may cause the SM to be under/over engineered.
Corrective Action Taken:	The global variable PCnts_usage is updated to contain the total number of busy network rITSDB tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of PCnts_usage.
<i>Bad Data =</i>	Contents of PCnts_usage.
<i>Good Data =</i>	Total number of busy network rITSDB tuples.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 6 Error Code: CRAM_E_UPPERBITS

Error Code:	CRAM_E_UPPERBITS
Error Description:	The value in the upper eight bits of control RAM E (CRAM-E) is invalid for the current set of options enabled for this TSIU model 3.
Possible Error Effect:	Unpredictable effects can occur when an option is enabled and the associated (and needed) hardware is not equipped.
Corrective Action Taken:	The correct value of the options covered in this audit has been written into the upper portion of CRAM-E.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the faulty TSDB tuple.
<i>Bad Data =</i>	A composite of the errors found during the check of the upper 8 bits of CRAM-E. Any bit=1 is in error.
<i>Good Data =</i>	The value of the global TSIU3 option indicator, Gltsopt. NOTE: This register uses negative logic: 0 = Option ENABLED. 1 = Option DISABLED.
<i>Logical Index =</i>	The key of the faulty TSDB tuple.
<i>Dump 1 =</i>	The faulty TSDB tuple.
<i>Dump 2 =</i>	The entire structure of type tsNTSCONFG.

Table 7 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	A stable call is not scanning for discontinuity.
Possible Error Effect:	The call process may be stuck, since it will never see the discontinuity and tear itself down.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list. As part of initialization, the audit zeros the linkage block "to" pointer. The rITSDB is connected to channel zero (silence), and the time slot is set to scan for discontinuity.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the rITSDB tuple.
<i>Dump 1 =</i>	rITSDB tuple.
<i>Dump 2 =</i>	PHDB tuple.

Table 8 Error Code: E_BIT_IN

Error Code:	E_BIT_IN

Error Description:

This audit error can be the result of one of five causes:

- (1) Timing anomaly - no action needed.
- (2) Hard switch of TMS - no action needed.
- (3) Out of date TMS fabric - remove and restore ONTCs.
- (4) Faulty hardware - check hardware.
- (5) Bad ODD (RSM only) - check ODD.

The first two causes will not cause persistent audit failures. If this audit error persists check whether the error is being reported from an RSM or not. If it is from an RSM, the audit error could be an ODD problem or a hardware problem. If the audit is reported from an SM that is NOT an RSM, it is a result of faulty hardware or an out of date TMS fabric. The KEY to the TSDB relation represents the network side time slot number of the TSI. The KEY, time slot, should be converted to its decimal equivalent when used to verify the ODD and hardware.

A. Non-persistent audit failures - no action needed.

If this audit error is generated, but clears itself up, no user action is needed. It is assumed to be caused by either a timing anomaly or a hard switch of the TMS.

1. Timing anomaly - no action needed.

If this audit error is generated on one TSDB as an isolated occurrence, this can be treated as a timing anomaly caused by an abnormally long delay in sending a message over the C-links. No action need be taken.

2. Hard switch of TMS links - no action needed.

If this audit error is generated after the hard switch of the TMS links, the error is probably the result of the TMS copy being made active when its fabric was not updated to the current state of connections. This condition will be automatically corrected by a TMS fabric update, so audit errors should cease in a few minutes. The exact amount of time depends on the delay incurred in getting the audit reports to the ROP.

B. Persistent audit failures.

1. Non-RSM audit failures.

If this audit error is reported from an SM that is connected directly to the TMS (not an RSM), it is the result of either an out of date TMS fabric or a faulty DLI.

a. Out of date TMS fabric - remove and restore the ONTC.

The TMS fabric could be out of date on one or both sides because of losing connect orders from the SMs or any failure in updating the TMS fabric during an ONTC restoration. This can be corrected by conditionally removing and restoring both ONTC sides. The restoration should cause the fabric to be brought up to date.

If a conditional removal is denied due to "AUTISS", switch ONTCs and try to conditionally remove the other side. If this is also denied, you are in a degraded mode in the ONTC. This means that there is faulty equipment in the active major side of the ONTC causing "AUTISS" to be active. This should be corrected as soon as possible. Do not disable the TSDB audit to eliminate the output, because it may be needed to recover from any unassociated software problems in the TSDB data.

b. Faulty hardware - check the hardware.

If this audit error is reported from an SM that is connected directly to the TMS (not an RSM), it could be the result of a faulty DLI anywhere on the switch sending E-bit on the minor ONTC side. When E-bit is seen on this SM, AUTISS would send this through so that the audit would see it. If the ONTC is in a duplex condition, put the minor ONTC out of service and diagnose it. This should resolve the problem.

If this audit error is reported from an RSM, use the KEY and the RLRTS_RTS and RLRTS_HCN relations to find the SM number of the far end of the facility. Wait until traffic permits the removal of one of the facilities. Use the DLTU page (112x) on the MCC to locate all the facilities connecting this RSM to the given far end. One at a time, go through each facility, putting its DFI out of service. After each DFI is put out of service, run the TSDB audit to see whether the audit error is still reported. It will not be reported on out of service time slots. When the correct DFI is found, diagnose it and the associated DFI at the other end to identify the faulty hardware. If the diagnostics pass, the problem may be in the facility.

2. RSM audit failures.

If this audit error is repeatedly reported from an RSM, the problem may be either faulty hardware or bad ODD.

a. Faulty hardware - check the hardware.

If this audit error is reported from an RSM, use the KEY and the RLRTS_RTS and RLRTS_HCN relations to find the SM number of the far end of the facility. Wait until traffic permits the removal of one of the facilities. Use the DLTU page (112x) on the MCC to locate all the facilities connecting this RSM to the given far end. One at a time, go through each facility, putting its DFI out of service. After each DFI is put out of service, run the TSDB audit to see whether the audit error is still reported. It will not be reported on out of service time slots. When the correct DFI is found, diagnose it and the associated DFI at the other end to identify the faulty hardware. If the diagnostics pass, the problem may be in the facility.

b. Bad ODD - check the ODD.

If this audit error is reported from an RSM, the error could be a result of one or more inconsistencies among the various relations describing the links on this RSM. Using the KEY, check the RLRTS_RTS and RLRTS_HCN relations to determine whether the time slot is on a link to the HSM or to another RSM in the cluster.

If the error from the RSM is on a time slot linked to the HSM, check that the time slot appearance in RLRTS_HCN agrees with RLHCN_RTS. Also check the RLICL_MSK relation in the RSM. The tuple for the HSM should be checked at the bit position corresponding to the time slot. This position should be set to one. Refer to the Audits Overview section for the interpretation of the RLICL_MSK relation. Finally, use RLRTS_HCN to get the himcktname for the time slot. Use the himcktname as the key into RLCKTDATA at the HSM. Check that the circuit has a type of

PCT1_HIM.

If the link is to another RSM in the cluster, compare the RLRTS_RTS and RLICL_MSK relations in the two RSMs. This is done by first checking the RLICL_MSK relation in the RSM that reported the audit error. Examine the tuple corresponding to the other RSM. The bit position corresponding to the time slot should be set to one. Next, compare the RLRTS_RTS relation in the two RSMs to see that they agree on the correlation between the time slot names in the two RSMs and the RSM number at the other end. Finally, check the RLICL_MSK relation at the RSM at the other end of the facility. Using the time slot name from RLRTS_RTS to check the bit position as was done at the RSM where the audit failed.

3. Reading the RLICL_MSK relation in an RSM.

The RLICL_MSK relation has the following fields: mod, msk01, msk02, msk03, ... msk16. The mod field is the key and signifies the SM number of the RSM or HSM that connects to this RSM. RLICL_MSK always contains tuples that have a mod field value of zero and the this RSM. These are special tuples that will not be described here.

The msk01-msk16 fields contain an array of 32 bits represented in the ODD by 8 hexadecimal digits. Each bit corresponds to a time slot. Msk01-msk08 contain all the even numbered time slots. Msk09-msk16 contain all the odd numbered time slot. Msk01 contains the even numbered time slots from 0 to 62. Msk02 contains the even numbered time slots from 64 to 126. Msk09 contains the odd numbered time slots from 1 to 63. The following table shows what time slots are contained in each msk?? field.

msk01	- even time slots	0- 62
msk02	- even time slots	64-126
msk03	- even time slots	128-190
msk04	- even time slots	192-254
msk05	- even time slots	256-318
msk06	- even time slots	320-392
msk07	- even time slots	394-456
msk08	- even time slots	458-510
msk09	- odd time slots	1- 63

```

msk10 - odd time slots 65-127
msk11 - odd time slots 129-191
msk12 - odd time slots 193-255
msk13 - odd time slots 257-319
msk14 - odd time slots 321-393
msk15 - odd time slots 395-457
msk16 - odd time slots 459-511

```

To find the position of a particular time slot, use the table to find the corresponding field. Read the value of that field for the desired tuple, and expand it from hexadecimal form to binary form. Then one can easily check the value corresponding to the desired time slot. As an example, if you wanted to examine the value of time slot 291 going to RSM 12, you would check the table to find that the status of time slot 291 is found in the msk13 field. Let's say that the value for the msk13 field in the tuple for mod 12 is:

20004040

When expanded to binary, this hex value becomes:

0010 0000 0000 0000 0100 0000 0100 0000

By writing out these values next to the corresponding time slots we can easily see the value for time slot 291.

```

257-259-261-263 <--> 0010
265-267-269-271 <--> 0000
273-275-277-279 <--> 0000
281-283-285-287 <--> 0000
289-291-293-295 <--> 0100
297-299-301-303 <--> 0000
395-307-309-311 <--> 0100
313-315-317-319 <--> 0000

```

Time slot 291 has a value of 1. So, according to RLICL_MSK, time slot 291 is connected to a T1 facility going to RSM 12.

Possible Error Effect:	See the Error Description.
Corrective Action Taken:	None. This is a manual action required error.
Manual Action Required:	Investigate why we are not receiving the correct E bit status for the give idle time slot.
Dump Description:	
<i>Error Address =</i>	Address of rITSDb tuple.
<i>Bad Data =</i>	The "e_bit_in" field from the TSntsconfig.
<i>Good Data =</i>	The idle state value (ID BIT).

<i>Logical Index</i> =	Logical key of the rITSDB tuple.
<i>Dump</i> =	NA.

Table 9 Error Code: FIDB_SGN

Error Code:	FIDB_SGN
Error Description:	The audit attempted to obtain a remote inter-module time-slot (RTERTS) circuit name from a time slot on a remote SM, but was not successful.
Possible Error Effect:	Possible error in time slot usage on the SM.
Corrective Action Taken:	None.
Manual Action Required:	Either the UNIT_CKT or FIDB_SGN tuple is missing for this given RTERTS circuit name derived from the time slot number. The given time slot number is listed in the good data of the audit dump. Insert the missing UNIT_CKT or FIDB_SGN tuple for this corresponding time slot using recent change or ODBE.
Dump Description:	
<i>Error Address</i> =	Address of the buffer to hold the corresponding ICL_MSK tuple.
<i>Bad Data</i> =	Return code GLNULCKT.
<i>Good Data</i> =	The value of the time slot.
<i>Logical Index</i> =	NA
<i>Dump</i> =	The rIICL_MSK tuple.

Table 10 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	An idle peripheral time slot data block tuple (rITSDB) is linked to an path data block tuple (rIPHDB) head cell for network TSDBs, or a network rITSDB tuple is in a rIPHDB head cell for peripheral rITSDBs.
Possible Error Effect:	This may cause asserts to be fired by call processing.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the correct idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rITSDB tuple.
<i>Bad Data</i> =	Logical key of the rIPHDB tuple (rIPHDN.phdbnum).
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rITSDB tuple.
<i>Dump 1</i> =	The rITSDB tuple.
<i>Dump 2</i> =	The rIPHDB tuple.

Table 11 Error Code: IDLE_CH

Error Code:	IDLE_CH
Error Description:	An idle rITSDB tuple's associated time slot is not receiving from the idle channel (Gldlech) or other TSI hardware states are not idle. Check the dump of the structure to determine which state(s) is(are) bad.
Possible Error Effect:	Excessive noise may be picked up (that is, crosstalk).
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list. As part of initialization, the audit forces the time slot to receive from channel Gldlech. Also, set all other related entries to their idle states.
Manual Action Required:	Request the CHDB audit if it has not already run.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB tuple.
<i>Bad Data</i> =	Channel from which the time slot is receiving.
<i>Good Data</i> =	Gldlech.
<i>Logical Index</i> =	Logical key of the rITSDB tuple and time slot.
<i>Dump 1</i> =	rITSDB tuple.
<i>Dump 2</i> =	The pc_nts_cofg structure.
<i>Dump 3</i> =	PHDB tuple.

Table 12 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The remote time slot type attribute (rITSDB.rstype) of the rITSDB tuple is not marked as in service (PCINS), or the corresponding bit in the out-of-service (OOS) time slot map (PCmpoos) is not set.

Possible Error Effect:	The improper type of time slot could be allocated to a call.
Corrective Action Taken:	The audit reinitializes the rTSDB tuple and it is linked to the idle list. The corresponding bit in the OOS time slot map (PCmpoos) is set.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rTSDB tuple.
<i>Bad Data =</i>	Content of the OOS time slot map (PCmpoos).
<i>Good Data =</i>	Content of rTSDB.rtsstat.
<i>Logical Index =</i>	Logical key of the rTSDB tuple.
<i>Dump 1 =</i>	rTSDB tuple.
<i>Dump 2 =</i>	Owning rPHDB tuple.

Table 13 Error Code: NAILEDUP

Error Code:	NAILEDUP
Error Description:	On an SM (switching module, not valid for SM-2000), an rTSDB.rtsstat value is either PCTSNLDUP when it should not be, or is not PCTSNLDUP when it should be. On an SM-2000, one of the following is true: (1) An rTSDB.rtsstat tuple is inconsistent with CTS (control timeslot) values (error address = DBNO_MATCH). (2) An rTSDB.rtsstat value is inconsistent with the type and/or netwk value(s) from the corresponding tuple of RLNTSTYPE (keyed by rINTSTYPE.tsnum = rTSDB.tsnum). (3) An rINTSTYPE.netwk value is invalid, or the audit code needs to be updated to recognize it. (4) An rINTSTYPE.type value is invalid, or the audit code needs to be updated to recognize it.
Possible Error Effect:	The time slot may not be correctly allocated. Depending on what the correct allocation for the time slot is, the SM may have a reduced ability to handle particular kinds of traffic [for example, CTS traffic or message handler (MH) traffic].
Corrective Action Taken:	The audit initializes the rTSDB tuple. As part of the initialization the rtsstat field is set correctly. On an SM-2000, if the error is due to case (3) or (4) listed in the error description, the error is reported each time the TSDB audit runs, until either the static relation rINTSTYPE is corrected (if the value is invalid), or the audit code is updated (if the value is valid). Office personnel should consult the population rules to determine if the values of rINTSTYPE.type and rINTSTYPE.netwk are valid. If they are, office personnel should contact their next line of support to get the audit code updated. Otherwise, office personnel should correct the values.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	For an SM (non SM-2000), the value of global parameter Glimtype. For an SM-2000, return from read of the rINTSTYPE tuple.
<i>Bad Data =</i>	rtsstat field of rTSDB tuple.
<i>Good Data =</i>	CTS value
<i>Logical Index =</i>	Logical key of rTSDB tuple.
<i>Dump 1 =</i>	The rTSDB tuple.
<i>Dump 2 =</i>	The associated PHDB tuple.
<i>Dump 3 =</i>	For an SM (non SM-2000), the rRSMNTS tuple if available. For an SM-2000, return from read of the rINTSTYPE tuple.

Table 14 Error Code: NO_OWN_TS

Error Code:	NO_OWN_TS
Error Description:	The rtstype field of an RSM rTSDb tuple and its linkage or the time slot allocation maps do not agree.
Possible Error Effect:	Time slots will not be correctly allocated.
Corrective Action Taken:	The audit initializes the rTSDb tuple. As part of the initialization the rtstat and rtstype fields are set correctly and the allocation maps marked accordingly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rTSDC tuple.
<i>Bad Data =</i>	Bit 0 is state of out-of-service map (idle = 1; busy = 0). Bit 1 is state of in-service map.
<i>Good Data =</i>	rtstat field of rTSDb tuple.
<i>Logical Index =</i>	Logical key of rTSDb tuple.
<i>Dump =</i>	rTSDb tuple.

Table 15 Error Code: PATHSTT

Error Code:	PATHSTT
Error Description:	The path state of the rIPHDB owning a rTSDb is invalid.
Possible Error Effect:	Asserts by call processing.
Corrective Action Taken:	The audit initializes the rTSDb tuple and places it on the idle list. As part of initialization, the audit zeros the linkage block "to" pointer. The rTSDb is connected to channel zero (silence), and the time slot is set to scan for discontinuity.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTSDb tuple.
<i>Bad Data =</i>	Path state of the owning PHDB.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the rTSDb tuple.
<i>Dump 1 =</i>	rTSDb tuple.
<i>Dump 2 =</i>	Owner rIPHDB tuple.

Table 16 Error Code: RSMNTS

Error Code:	RSMNTS
Error Description:	Invalid DB return when accessing RLRSMTS relation, possible ODD problem. This error will continue to print out until fixed manually.
Possible Error Effect:	Relation can not be accessed.
Corrective Action Taken:	None.
Manual Action Required:	If error repeats, investigate the ODD data.
Dump Description:	
<i>Error Address =</i>	Address of the rTSDb tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLRSMTS relation.
<i>Dump =</i>	Local buffer used to access rLRSMNTS tuple.

Table 17 Error Code: RTS_HCN

Error Code:	RTS_HCN
Error Description:	Invalid DB return when accessing the RLRTS_HCN relation which may indicate an ODD problem.
Possible Error Effect:	RSM time slot number to HSM DFI channel circuit name translation cannot be made.
Corrective Action Taken:	The tuple is assumed to be correct.
Manual Action Required:	If error repeats, investigate the ODD.
Dump Description:	
<i>Error Address =</i>	Address of the rTSDb tuple.
<i>Bad Data =</i>	Bad DB return.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Key used to access the RLRTS_HCN relation.
<i>Dump =</i>	rRTS_HCN buffer.

Table 18 Error Code: STABLE_LINK

Error Code:	STABLE_LINK
Error Description:	An rITSDB tuple in a stable state has bad linkage.
Possible Error Effect:	The bad linkage may cause an ASSERT before the audit finds the error.
Corrective Action Taken:	The audit idles the rITSDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rITSDB tuple address.
<i>Bad Data =</i>	Contents of linkage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Physical key of the rITSDB tuple.
<i>Dump =</i>	rITSDB tuple.

Table 19 Error Code: STATUS

Error Code:	STATUS
Error Description:	The time slot is idle or associated with a stable call but the time slot job map associated with immediate audible does not indicate that this time slot is idle.
Possible Error Effect:	This time slot could be connected to audible ring.
Corrective Action Taken:	PCcnaud() is called to idle AR job.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	The state field from the PCTsdbjb[] table.
<i>Good Data =</i>	The idle state value (PCARIDLE).
<i>Logical Index =</i>	Key used to access the RLTSDB relation.
<i>Dump =</i>	rITSDB tuple.

Table 20 Error Code: STUCK

Error Code:	STUCK
Error Description:	The rITSDB tuple has been in the transient state for more than 5 minutes.
Possible Error Effect:	A call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rITSDB tuple is zeroed. The call is terminated.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	Valid values are: 0 = If peripheral control is not scanning for E-bit. 1 = If peripheral control is scanning for E-bit. 2 = Not applicable.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Logical key of rITSDB tuple.
<i>Dump 1 =</i>	rITSDB tuple.
<i>Dump 2 =</i>	PHDB tuple.

Table 21 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	A rITSDB tuple's associated time slot is stuck in some invalid state. The tuple will be corrected and returned to the appropriate idle list.
Possible Error Effect:	A time slot may not be used for call processing.
Corrective Action Taken:	The audit initializes the rITSDB tuple and places it on the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	0.
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of rITSDB tuple and time slot.
<i>Dump 1 =</i>	rITSDB tuple.

<i>Dump 2</i> = rIPHDB tuple.

Table 22 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The peripheral rTSDB tuple has been in the transient state for more than five minutes.
Possible Error Effect:	A peripheral to peripheral call in progress is stuck in a transient state and cannot disconnect.
Corrective Action Taken:	The rTSDB tuple is zeroed. The call is terminated. The rTSDB tuple is placed in rIPHDB headcell for idle peripheral rTSDB tuples.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTSDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key of the rTSDB tuple.
<i>Dump</i> =	The rTSDB tuple.

Table 23 Error Code: TSMAP1

Error Code:	TSMAP1
Error Description:	The rtsstat field of an RSM rTSDB tuple and its linkage or the time slot allocation maps do not agree.
Possible Error Effect:	Time slots will not be correctly allocated.
Corrective Action Taken:	The audit initializes the rTSDB tuple. As part of the initialization the rtsstat field is set correctly and the allocation maps marked accordingly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rTSDB tuple.
<i>Bad Data</i> =	Bit 0 is state of inter/out-of-service map (idle = 1; busy = 0). Bit 1 is state of inter/in-service map. Bit 2 is state of intra map.
<i>Good Data</i> =	rtsstat field of rTSDB tuple.
<i>Logical Index</i> =	Logical key of rTSDB tuple.
<i>Dump 1</i> =	rTSDB tuple.
<i>Dump 2</i> =	PHDB tuple.

Table 24 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	For network rTSDBs, the index number in this description corresponds to the <i>Logical Index</i> value. So based on the <i>Logical Index</i> , one of the following apply: 1 = The time slot interchanger (TSI) time slot attribute (rTSDB.tsits) of the rTSDB tuple does not contain the proper TSI time slot number. 2 = For peripheral rTSDBs, the TSI time slot attribute (rTSDB.tsits) is not equal to value PCIDLETN for a virtual path (rCCBCOM.lptype = PCDPN_VIRT). 3 = For peripheral rTSDBs, the TSI time slot attribute (rTSDB.tsits) is not equal to the logical key of rCHDB (derived from rCCBCOM.ccrgm.st_link).
Possible Error Effect:	Corresponds to the Error Description as follows: 1 = This can cause peripheral control (PC) to write to the wrong TSI control RAMs, causing paths through the TSI to be opened erroneously. 2 = May cause the rTSDB tuple to be not usable.

	3 = This can cause PC to write to the wrong TSI control RAMs, causing paths through the TSI to be opened erroneously.
Corrective Action Taken:	Corresponds to the Error Description as follows: 1 = The rTSDB has been reinitialized and linked to the idle list. The TSI control RAMs associated with this rTSDB tuple have been reinitialized. 3 = The rTSDB has been reinitialized and linked to the idle list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rTSDB tuple.
<i>Bad Data =</i>	Contents of the TSI time slot attribute (rTSDB.tsits).
<i>Good Data =</i>	Correct value for rTSDB.tsits.
<i>Logical Index =</i>	1, 2, or 3.
<i>Dump 1 =</i>	rTSDB tuple.
<i>Dump 2 =</i>	rPHDB tuple (if <i>Logical Index</i> equals 2 or 3).
<i>Dump 3 =</i>	rCCBCOM tuple (if <i>Logical Index</i> equals 2 or 3).

Table 25 Error Code: TYPE

Error Code:	TYPE
Error Description:	The rtstype field of the rTSDB tuple disagrees with the actual type of the RSM time slot (INTRA, INTER/IN SERVICE, or INTER/OUT OF SERVICE).
Possible Error Effect:	The improper type of RSM time slot could be allocated to a call.
Corrective Action Taken:	The audit initializes the rTSDB tuple. As part of the initialization the rtstype field is set correctly.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rTSDB tuple.
<i>Bad Data =</i>	rtstype field of rTSDB tuple.
<i>Good Data =</i>	The correct time slot type.
<i>Logical Index =</i>	Logical key of rTSDB tuple.
<i>Dump 1 =</i>	rTSDB tuple.
<i>Dump 2 =</i>	PHDB tuple.

Table 26 Error Code: ZERO_LINK

Error Code:	ZERO_LINK
Error Description:	An idle rTSDB tuple has nonzero linkage in its one-to-one linkage block ("to" pointer).
Possible Error Effect:	The time slot is unusable.
Corrective Action Taken:	The audit initializes the rTSDB tuple and places it on the idle list. As part of initialization, the audit zeros the linkage block "to" pointer.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rTSDB tuple.
<i>Bad Data =</i>	Linkage contents ("to" pointer).
<i>Good Data =</i>	0.
<i>Logical Index =</i>	Logical key of the rTSDB tuple.
<i>Dump 1 =</i>	rTSDB tuple.
<i>Dump 2 =</i>	rPHDB tuple.

TSI

Software Release: 5E10 and later

1. TIME SLOT INTERCHANGER (TSI) AUDIT

The time slot interchanger provides connections between incoming peripheral time slots (channels) and network time slots. The interchanging of the peripheral and network time slots necessary for completing calls is controlled by control RAMs. Refer to Figure 1. Five control RAMs (A through E) represent the paths of the time slots and the states of the hardware of the time slot interchanger. Another control RAM, the alternate data RAM, performs other control functions. Unlike control RAMs A through E, the alternate data RAM does not allocate its 512 locations for time slots. Rather, the alternate data RAM has locations for the tones sent to and returned from the local digital service unit, module processor constants and work space, the dual link interface E-bit buffer, and locations for the data interface error buffer. The system uses control RAM C to connect the network and peripheral time slots to the alternate data RAM.

During system initializations, the configurations of the time slot interchanger and the control RAMs are set to idle. The channel data block (CHDB) and the time slot data block (TSDB) audits ensure that the system has a consistent view of idle peripheral and network time slots. The time slot interchanger (TSI) audit verifies that the connections for idle time slots to the alternate data RAM are valid. It also ensures that the idle code constant in the module processor constants area is valid.

The ADR contains, in addition to other areas, an area for constant values which may be transmitted as the PCM on a circuit. Four constant values exist as part of this arrangement: the idle code, MUX (trouble word for DDS MUX out of service), the TRB (trouble word for analog series 2000/3000 quality of transmission), and the UMC (DDS unassigned multiplex code).

The TSI audit is part of the peripheral control audits subgroup. It operates in the operating system for distributed switching-switching module (OSDS-SM) environment in the segmented modes.

2. DATA STRUCTURES

The primary data structures representing the time slot interchanger are the time slot data block relation (RLTSDB), the channel data block relation (RLCHDB), and tsADRCNFIG. The time slot data blocks represent the network side time slots. The channel data blocks represent the peripheral side time slots. The paths between the channel and peripheral time slots are stored in the path data block relation (RLPHDB). The system stores views of the customer's ports in the channel control block common area relation (RLCCBCOM).

The following data structures are used for SM-2000 processors only:

- **ldsf_ts** - a field in the rLCHDB relation which represents the LDSF TSI time slot associated with the CHDB.
- **rLDSC3PIDB** - a relation that defines the PIDBs assigned to a Digital Service Circuit 3.
- **picb** - a field in the RLDSC3PIDB relation that represents the key to the RLDSC3PIDB relation.
- **pidb0** - a field in the RLDSC3PIDB relation that represents the 1st PIDB number assigned to the DSC3.
- **tsTSICONFG** - a structure used when reading the TSIU4 control RAMs for a given TSI time slot or a given TSI link.
- **dsc3ts[]** - a field in the pcDSC3CKTS structure that represents the TSI time slot number.
- The peripheral link interface to link (RLPLI_LNK) relation is used for peripheral link interface (PLI) time slots. This relation maps from the TSI end of the PCT to the physical link position with respect to the peripheral unit.

- The virtual peripheral interface data bus (RLVPIDB_F) relation is used to map a virtual peripheral interface data bus (VPIDB) to its corresponding SONET facility Number.

Figure 2 shows these relations.

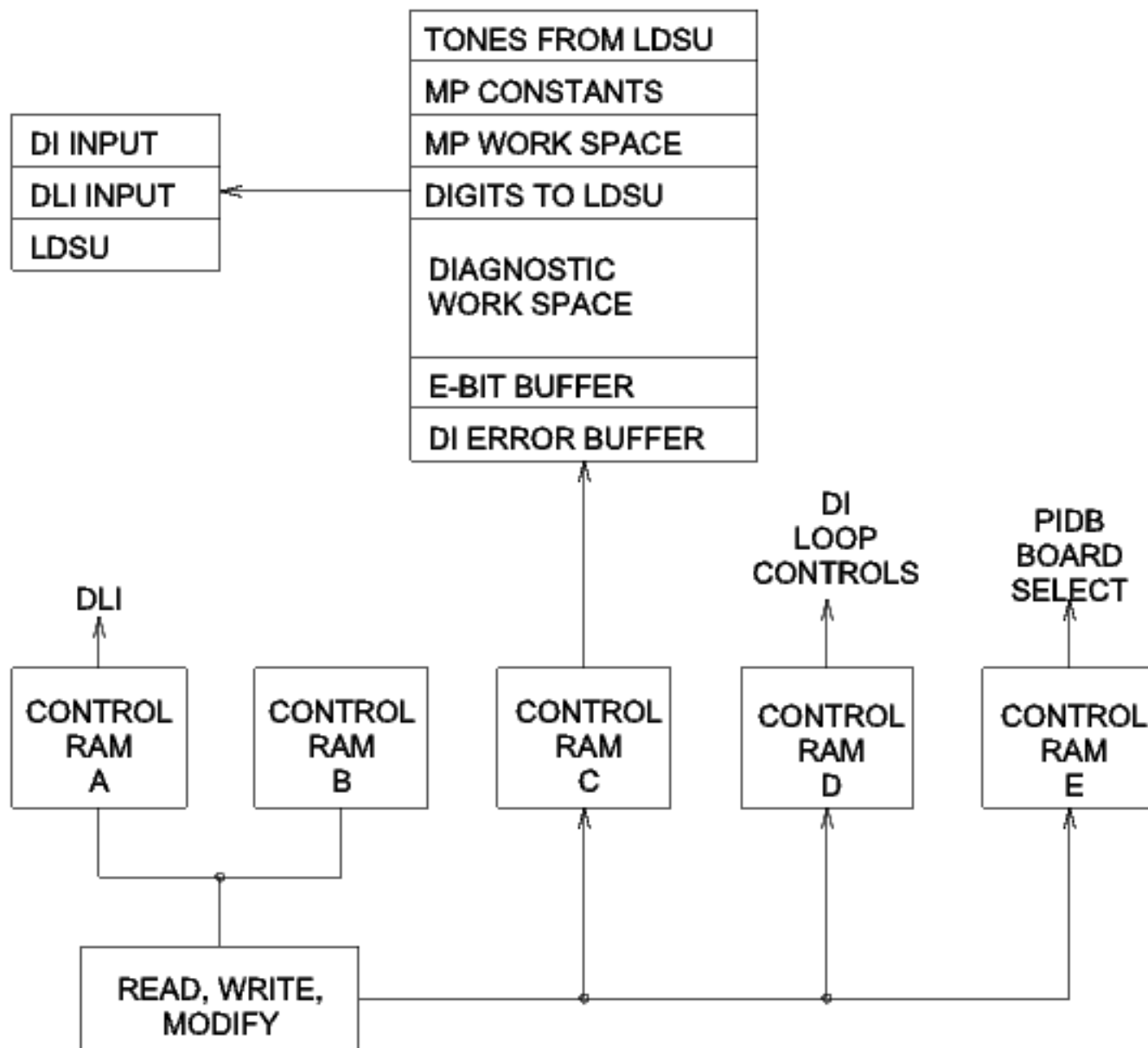


Figure 1 (TSI-1) TSI Hardware Configuration

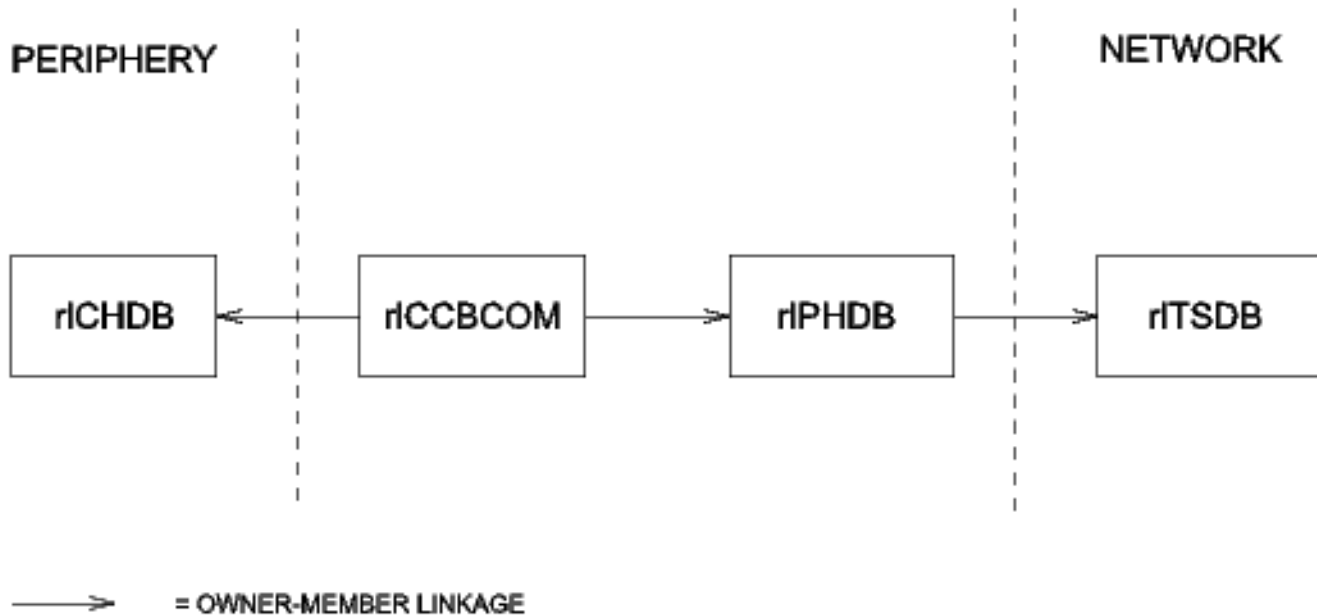


Figure 2 (TSI-2) TSI Structures of the TSI Audit

3. AUDIT CHECKS

The audit performs two checks: a verification of the connections between the alternate data RAM and the idle channels, and a check of the constant area for the module processor within the alternate data RAM. The audit invokes the dynamic driver (DRDYN) to perform both these checks. DRDYN steps through the rICHDB relation and invokes the semantic check function for each tuple.

To verify the connections between the alternate data RAM and the idle channels, the audit performs these activities:

- The audit checks the IN_PROGRESS (**inproc**) field of rICHDB to determine whether the channel is idle.
- If the channel is idle, the audit determines whether the corresponding network time slot is idle by checking the appropriate tuple of rITSDB.
- The audit then checks the paths to and from the alternate data RAM and ensures that they are idle.
- If the rICHDB tuple is equal to the audible ringing channel assignment global variable (Glaudring), the control RAM C alternate data RAM read address value is checked against the rITONECH tuple. If an error is found, the control RAM information is overwritten with the correct value.
- All eight channels in a column (dbchan) are checked to see that there are no duplicates among the eight PIDB assignments.

If the rICHDB tuple matches the AUTOVON precedence audible channel assignment Glpaudring, the control RAM C (ADR) real address is checked against the ADR port assignment (ldsucInn field) in the corresponding rITONECH tuple. An error will cause the ADR information to be overwritten with the correct value.

The audit then checks the module processor constant area of the alternate data RAM. It ensures that values are in range and that there are no extraneous constants.

Previously the TSI audit section which validated the ADR constant area would check the idle code value once for each time slot in the SM (512 times). Since the audit checks the four constants, the audit checks the constant area

only once when the audit is run. In other words, instead of checking the idle code constant 512 times, this section of the audit now checks all four constant values only one time. The values are checked by reading the ADR constant area and comparing the data against hard coded values. If the data is inconsistent with the correct values, the audit prints a message and corrects the RAM.

The TSI hardware is checked for all peripheral time slots associated with PLI links. For PLI links, the equipage bits for data group 1 and the associated TMUXs in the PCT Auxiliary Control Source Register are audited.

4. ERROR HANDLING

When the TSI audit detects an error, it calls a peripheral control primitive to correct the time slot interchanger hardware.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: ACBIT

Error Code:	ACBIT
Error Description:	Each CD in a DNU_S has its own copy of the peripheral configuration table (PCT) auxiliary control source (AUX) register. Contained in the copy of this register is an application message inhibit bit (appl_msginh). The bit is used to inhibit application messages from being sent from a DNU_S for applications that are not active yet during a full or selective initialization of an SM. It should be 0 outside of the full or selective initialization and 1 during these initializations. This error indicates that this bit has been corrupted.
Possible Error Effect:	Application messages from the DNU_S may be lost if this bit is set outside of SM initializations.
Corrective Action Taken:	The application message inhibit bit is set to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The address of the buffer used to read the rIPLI_LNK tuple.
<i>Bad Data =</i>	The value of the appl_msginh bit before correction.
<i>Good Data =</i>	The value of the appl_msginh bit after correction.
<i>Logical Index =</i>	The CD index (0 or 1) whose AUX register copy is in error.
<i>Dump =</i>	rIPLI_LNK tuple buffer.

Table 2 Error Code: ASSOC_TUP

Error Code:	ASSOC_TUP
Error Description:	Upon trying to verify the TSI control RAMs state is idle for an idle rITSDB tuple, the TSI audit uncovers a corrupted rIPHDB tuple which owns the rITSDB tuple.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The TSDB audit will be run to detect and recover this error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to the rITSDB tuple.
<i>Bad Data =</i>	The rIPHDB tuple's PHDB number.
<i>Good Data =</i>	The rITSDB tuple's pointer to the rIPHDB tuple.
<i>Logical Index =</i>	TSI time slot number.
<i>Dump =</i>	rITSDB tuple.

Table 3 Error Code: AUD_RING

Error Code:	AUD_RING
Error Description:	The alternate data RAM LDSU port assignment for audible ringing is wrong according to the value in the PCTAUD rITONECH tuple.
Possible Error Effect:	Audible ringing tone is not working in this SM.

Corrective Action Taken:	The correct value has been written. Audible has been restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Value found for audible ringing.
<i>Good Data =</i>	Correct value.
<i>Logical Index =</i>	Key of CHDB.
<i>Dump 1 =</i>	ADR information structure of the type tsADRCNFG for the audible slot.
<i>Dump 2 =</i>	Audible ringing rITONECH tuple.

Table 4 Error Code: BADSPIGRAM

Error Code:	BADSPIGRAM
Error Description:	The AUTISS reporting is not disabled or the SP scanning is on for this unassigned/unequipped time slot.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	AUTISS reporting status is corrected and SP scanning is turned off.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	Upper word: SP scanning inhibited value. Lower word: Read AUTISS reporting status.
<i>Good Data =</i>	Upper word: Read SP scanning status. Lower word: AUTISS reporting disabled value.
<i>Logical Index =</i>	TSI time slot number.
<i>Dump =</i>	NA.

Table 5 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The rICDBCOM tuple, associated with the specified TSI time slot number, has an invalid circuit type.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	Related audits, TONECH and CDBCOM, will be run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The rPICB tuple buffer's unit number (unitnum field).
<i>Bad Data =</i>	The rICDBCOM tuple's circuit type (type field).
<i>Good Data =</i>	Circuit name from the specified TSI time slot number's PCdsc3ckts[] entry.
<i>Logical Index =</i>	TSI time slot number.
<i>Dump =</i>	rICDBCOM tuple.

Table 6 Error Code: BADUSE

Error Code:	BADUSE
Error Description:	A TSI link usage was added without informing audits.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	TSI link usage.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	TSI link number.
<i>Dump =</i>	tsPCMDATA

Table 7 Error Code: BDLINK

Error Code:	BDLINK
Error Description:	The TSI IF control and IF diagnostic control registers for this audited TSI link were found to be invalid.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.

Corrective Action Taken:	The TSI registers will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	TSI link use.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	TSI link number.
<i>Dump =</i>	tsLNKCONFIG buffer. TSI RAM information.

Table 8 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The circuit name field of the rICDBCOM, associated with the specified TSI time slot number, was not equal to peripheral control (PC) PCdsc3ckts[] circuit name.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	Related CKTDATA audit will be run.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rPICB tuple buffer's unit number (unitnum field).
<i>Bad Data =</i>	rICDBCOM tuple's circuit name.
<i>Good Data =</i>	PCdsc3ckts[] entry circuit name.
<i>Logical Index =</i>	TSI time slot number.
<i>Dump =</i>	NA.

Table 9 Error Code: CRAM_E_UPPERBITS

Error Code:	CRAM_E_UPPERBITS
Error Description:	The value in the upper 8 bits of control RAM E (CRAM-E) is invalid for the current set of options enabled for this TSIU model 3.
Possible Error Effect:	Unpredictable effects can occur when an option is enabled and the associated (and needed) hardware is not equipped.
Corrective Action Taken:	The correct value of the options covered in this audit has been written into the upper portion of CRAM-E.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the faulty CHDB tuple.
<i>Bad Data =</i>	A composite of the errors found during the check of the upper 8 bits of CRAM-E. Any bit=1 is in error.
<i>Good Data =</i>	The value of the global TSIU3 option indicator, Gltsiopt. NOTE: This register uses negative logic: 0 = Option ENABLED. 1 = Option DISABLED.
<i>Logical Index =</i>	The key of the faulty CHDB tuple.
<i>Dump 1 =</i>	The faulty CHDB tuple.
<i>Dump 2 =</i>	The entire structure of type tsADRCONFIG. NOTE: See file hdr/smim/SMts_emctl.h for definition of CRAM-E. NOTE: See file hdr/smim/SMts_cmd3.h for definition of Gltsiopt.

Table 10 Error Code: DATA_CON_ERR

Error Code:	DATA_CON_ERR
Error Description:	Could not find a rIDSC3PIDB tuple associated with the specified LDSF PIDB.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The related CKTDATA audit will be run.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	RLDSC3PIDB relation number.
<i>Good Data</i> =	LDSF TSI time slot PIDB.
<i>Logical Index</i> =	Logical key of the rPICB buffer (picb field).
<i>Dump 1</i> =	rPICB tuple buffer.
<i>Dump 2</i> =	rPIDB tuple buffer.

Table 11 Error Code: DFLG

Error Code:	DFLG
Error Description:	Each CD in a DNU_S has its own copy of the PCT AUX register. Contained in the copy of this register is a data group 1 equipage bit (dg1_value). This bit should be 1 if any of the CD's in data group 1 are operational (that is, SMOPER). This error indicates that this bit has been corrupted.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing. It will also cause peripheral faults to occur in the SM.
Corrective Action Taken:	The data group 1 equipage bit is corrected in the CD's copy of the AUX register.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the buffer used to read the rPLI_LNK tuple.
<i>Bad Data</i> =	The value of the Data Group 1 equipage bit before correction.
<i>Good Data</i> =	The value of the Data Group 1 equipage bit after correction.
<i>Logical Index</i> =	The CD index (0 or 1) whose AUX register copy is in error.
<i>Dump</i> =	rPLI_LNK tuple buffer.

Table 12 Error Code: E_BIT

Error Code:	E_BIT
Error Description:	Each CD in a DNU_S has its own copy of the PCT AUX register. Contained in the copy of this register is an fault recovery message inhibit bit (frcvy_msginh). The bit is used to inhibit fault recovery messages from being sent from a DNU_S for facilities that are not active yet during a full or selective initialization of an SM. It should be 0 outside of the full or selective initialization and 1 during these initializations. This error indicates that this bit has been corrupted.
Possible Error Effect:	Fault recovery messages from the DNU_S may be lost if this bit is set outside of SM initializations.
Corrective Action Taken:	The fault recovery message inhibit bit is set to the correct value.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the buffer used to read the rPLI_LNK tuple.
<i>Bad Data</i> =	The value of the frcvy_msginh bit before correction.
<i>Good Data</i> =	The value of the frcvy_msginh bit after correction.
<i>Logical Index</i> =	The CD index (0 or 1) whose AUX register copy is in error.
<i>Dump</i> =	rPLI_LNK tuple buffer.

Table 13 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The ADR input source selection has been corrupted. It is set to take input from the network side which is illegal.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle value is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB.
<i>Bad Data</i> =	Bad idle value.
<i>Good Data</i> =	Good idle value.
<i>Logical Index</i> =	Idle time slot number.
<i>Dump 1</i> =	rCHDB.
<i>Dump 2</i> =	rITSDB.
<i>Dump 3</i> =	ADR information structure of the type tsADRCONFIG.

Table 14 Error Code: IDLE_BUSY

Error Code:	IDLE_BUSY
Error Description:	The ADR input source selection has been corrupted. It is set to take input from the network side which is illegal.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle value is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rICHDB.
<i>Bad Data</i> =	ADR input source select.
<i>Good Data</i> =	0.
<i>Logical Index</i> =	Key of CHDB.
<i>Dump</i> =	ADR information structure of the type tsADRCNFG.

Table 15 Error Code: IDLE_CHDB

Error Code:	IDLE_CHDB
Error Description:	TSI hardware checks on an idle time slot indicate a fault.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The time slot will be recovered.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rICHDB tuple.
<i>Bad Data</i> =	Channel type.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TSI time slot number.
<i>Dump 1</i> =	rICHDB tuple.
<i>Dump 2</i> =	tsTSICONFG buffer.

Table 16 Error Code: IDLE_CODE

Error Code:	IDLE_CODE
Error Description:	The TSI constant RAMs do not contain the correct idle code.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The TSI constant RAMs will be recovered to an idle state.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Bad idle constant.
<i>Good Data</i> =	Good idle constant.
<i>Logical Index</i> =	TSI slice number.
<i>Dump</i> =	tsPCMDATA

Table 17 Error Code: IDLE_LINK

Error Code:	IDLE_LINK
Error Description:	The ADR input source selection has been corrupted.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle value is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of rITSDB.
<i>Bad Data</i> =	Bad idle value.
<i>Good Data</i> =	Good idle value.
<i>Logical Index</i> =	Idle time slot number.
<i>Dump 1</i> =	rICHDB.
<i>Dump 2</i> =	rITSDB.
<i>Dump 3</i> =	ADR information structure of the type tsADRCNFG.

Table 18 Error Code: IDLE_OWNER

Error Code:	IDLE_OWNER
Error Description:	The idle constant in the alternate data RAM (ADR) has been corrupted.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle constant is written to the ADR.
Manual Action Required:	None.

Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	Bad idle constant.
<i>Good Data</i> =	Good idle constant.
<i>Logical Index</i> =	NA.
<i>Dump 1</i> =	rCHDB.
<i>Dump 2</i> =	rTSDB.
<i>Dump 3</i> =	ADR information structure of the type tsADRCONFIG.

Table 19 Error Code: MATELINK

Error Code:	MATELINK
Error Description:	This nailed-up channel's rCHDB tuple is missing a corresponding rCDBCOM tuple.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The CHDB audit will be run to clean up the discrepancy.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rCHDB tuple.
<i>Bad Data</i> =	The rCHDB tuple rCDBCOM link value.
<i>Good Data</i> =	The rCHDB tuple's inproc state.
<i>Logical Index</i> =	The rCHDB tuple channel number.
<i>Dump</i> =	rCHDB tuple.

Table 20 Error Code: MISSTUP

Error Code:	MISSTUP
Error Description:	The RLCHDB relation linkage to the RLCDBCOM relation is corrupt or the rCKTDATA tuple is missing from the database.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	CHDB audit will be triggered to clean up any linkage problem.
Manual Action Required:	If error occurs multiple times, rCKTDATA tuple may be missing and should be inserted into database by using ODBE and the tuple's key which is dumped as the <i>Logical Index</i> portion of the data dumped for this error.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name. Key for reading RLCKTDATA and RLSMEST.
<i>Dump</i> =	NA.

Table 21 Error Code: MUX_WRD

Error Code:	MUX_WRD
Error Description:	Each CD in a DNU_S has its own copy of the PCT AUX register. Contained in the copy of this register is the TMUX equipage bits (tmux_equip). There are a total of 7 bits (6 TMUX and 1 spare TMUX) for each CD. Each bit is set to one if the corresponding TMUX is equipped for that CD. This error indicates that the TMUX bits have been corrupted.
Possible Error Effect:	This error might cause unusual error conditions in call processing. It will also cause peripheral faults to occur in the SM.
Corrective Action Taken:	The TMUX equipage bits in the CD's copy of the AUX register is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The value of the PLI link in error.
<i>Bad Data</i> =	The value of the TMUX equipage bits before correction.
<i>Good Data</i> =	The value of the TMUX equipage bits after correction.
<i>Logical Index</i> =	The CD index (0 or 1) of the CD whose AUX register copy is corrupted.
<i>Dump</i> =	rPLI_LNK tuple buffer.

Table 22 Error Code: MUX_WRK

Error Code:	MUX_WRK
Error Description:	The ADR input source selection has been corrupted. It is set to take input from the network side which is an illegal setting.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle value is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Current DDS MUX out-of-service code.
<i>Good Data =</i>	Good idle value: TS_MUX.
<i>Logical Index =</i>	0.
<i>Dump =</i>	ADR information structure of the type tsADRCONFIG.

Table 23 Error Code: NOTIDLE

Error Code:	NOTIDLE
Error Description:	Time slot 0 in an RSM was being sourced from the TMS instead of from an alternate source.
Possible Error Effect:	If time slot 0 is being used as a control time slot, users will hear a high pitched whistle whenever silence is supposed to be applied to the line.
Corrective Action Taken:	Time slot 0 is forced to be sourced from the alternate source.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rITSDB tuple.
<i>Bad Data =</i>	Value for TMS.
<i>Good Data =</i>	Value for alternate source.
<i>Logical Index =</i>	0.

Table 24 Error Code: NO_CHDB

Error Code:	NO_CHDB
Error Description:	The rCHDB tuple associated with the specified TSI time slot is corrupt.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The CHDB audit will be triggered to recover the rCHDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Pointer to the rCHDB tuple.
<i>Bad Data =</i>	Channel number from the rCHDB tuple.
<i>Good Data =</i>	TSI time slot number.
<i>Logical Index =</i>	TSI time slot number.
<i>Dump =</i>	NA.

Table 25 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	A utility function could not find a valid circuit name or unit identification when reading static data after a real time break.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	None.
Manual Action Required:	Using ODBE, insert any missing rCKTDATA tuple or rSMEST tuple data using the circuit name key which is dumped in the Logical Index of this error report.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	Circuit name.
<i>Dump =</i>	rCKTDATA tuple.

Table 26 Error Code: PCD

Error Code:	PCD
Error Description:	The TSI audit was unable to read the corresponding static rPLI_LNK tuple for a PLI link when auditing the auxiliary control source register equipage bits.

Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	None.
Manual Action Required:	The missing rIPLI_LNK tuple needs to be inserted by recent change. The tsi_link field in the audit dump is the key of the missing rIPLI_LNK tuple. Note that this tuple also has a corresponding rILNK_PLI tuple that may also be missing.
Dump Description:	
<i>Error Address</i> =	The address of the buffer used to read the rIPLI_LNK tuple.
<i>Bad Data</i> =	The database return code from the read.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The tsi_link field containing the TSI link number.
<i>Dump</i> =	The buffer used to read the missing rIPLI_LNK tuple.

Table 27 Error Code: PICB

Error Code:	PICB
Error Description:	Could not find rIPICB tuple associated with a PIDB.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	None.
Manual Action Required:	Use ODBE to insert the rIPICB tuple using the PICB number dumped in the Bad Data of this error report as the key.
Dump Description:	
<i>Error Address</i> =	Address of audit's rIPICB tuple buffer.
<i>Bad Data</i> =	PICB number from rIPIDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Logical key from rIPIDB tuple (pidb field).
<i>Dump</i> =	NA.

Table 28 Error Code: PIDB

Error Code:	PIDB
Error Description:	Two channels within the same group of 8 (as known as 'dbchan') are using the same PIDB at the same time. This is illegal. NOTE: The 'dbchan' attribute (as used in CKTDATA) can be determined by using bits 4-8 of the channel numbers. Refer to Figure 3.
Possible Error Effect:	Will almost always cause PIDB parity errors and possibly other unknown conditions.
Corrective Action Taken:	The two CHDBs, corresponding to the two channels that are selecting the same PIDB, have had their quarantine bits set. They will be recovered when the CHDB audit gets around to them. Expect to see CHDB audit errors of the type QUAR pointing to each of the tuples flagged here. In addition, both channels' slots in control RAM E have been idle using TSwr_ach().
Manual Action Required:	Normally no action will be required for this type of error. It is assumed to be a simple mutilation of data in control RAM E. If it is noticed that these errors roll (continuously occur), however, there could be an ODD problem. Two rICKTDATA tuples have the same 'pidb' and 'dbchan' attributes. There should never be duplicates in these attributes within the same SM.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	PIDB number that the two channels were attempting to use. This is in the same format as it would be stored in a rICKTDATA tuple.
<i>Good Data</i> =	One of the errant channels.
<i>Logical Index</i> =	The other errant channel.
<i>Dump</i> =	There are no dumps. Since the quarantine bits have been set, the CHDB tuples will be dumped when the CHDB audit detects this quarantine status.

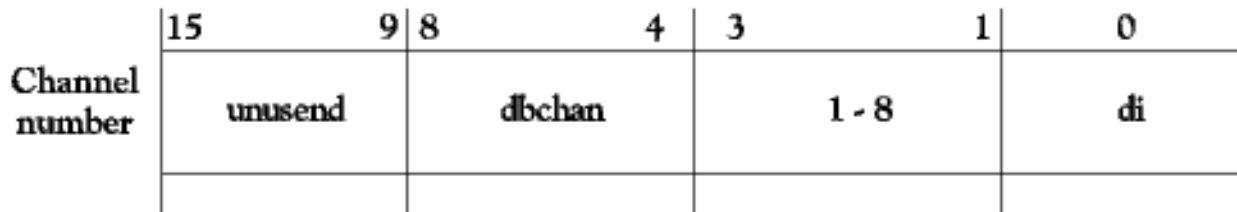


Figure 3 (TSI-3) The dbchan Attribute

Table 29 Error Code: PPAUD_CH

Error Code:	PPAUD_CH
Error Description:	The alternate data ram LDSU port assignment for precedence audible ringing wrong according to the value in the PCTPPAUD rITONECH tuple.
Possible Error Effect:	Precedence audible ringing tone may not be working in this SM.
Corrective Action Taken:	The correct value has been written. Audible has been restored.
Manual Action Required:	None.
Dump Description:	
Error Address =	0.
Bad Data =	LDSU ADR port assignment found for precedence audible ringing.
Good Data =	Correct, calculated port assignment.
Logical Index =	Logical key of CHDB tuple (Glpaudring channel number).
Dump 1 =	ADR information structure of the type tsADRCNCFG for the precedence audible slot.
Dump 2 =	Precedence audible ringing rITONECH tuple.

Table 30 Error Code: RD_XDSU

Error Code:	RD_XDSU
Error Description:	An idle channel has its extended DSU read enable in something but the idle state. This is indicative of a call not cleaning up after itself properly after termination.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The erroneous data has been corrected.
Manual Action Required:	None.
Dump Description:	
Error Address =	Address of the offending CHDB tuple.
Bad Data =	Actual contents of pc adr_cfg.xdsu rd_add.
Good Data =	Correct idle value.
Logical Index =	Key of the offending CHDB tuple.
Dump 1 =	rICHDB.
Dump 2 =	ADR information structure of the type tsADRCNCFG.

Table 31 Error Code: READ

Error Code:	READ
Error Description:	The path from the ADR was left in the connected state for this idle time slot.
Possible Error Effect:	The time slot that is now using this path may pick up noise on the line.
Corrective Action Taken:	Idle the connection from the ADR for this time slot.
Manual Action Required:	None.
Dump Description:	
Error Address =	Address of rITSDB.
Bad Data =	Path from the ADR.
Good Data =	ADR address for the idle constant.
Logical Index =	Idle time slot number.
Dump 1 =	rICHDB.
Dump 2 =	rITSDB.
Dump 3 =	ADR information structure of the type tsADRCNCFG.

Table 32 Error Code: READ

Error Code:	READ

Error Description:	The path from the ADR was left in the connected state for this idle time slot.
Possible Error Effect:	The time slot that is now using this path may pick up noise on the line.
Corrective Action Taken:	Idle the connection from the ADR for this time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of local pointer to CHDB.
<i>Bad Data</i> =	ADR XDSU write address.
<i>Good Data</i> =	Correct idle value.
<i>Logical Index</i> =	Key of CHDB.
<i>Dump 1</i> =	rICHDB.
<i>Dump 2</i> =	ADR information structure of the type tsADRCNFG.

Table 33 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	The TSI audit is attempting to read a nonexistent rSMEST tuple.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	CKTDATA audit will be triggered to run as a related audit and will insert the missing rSMEST tuple into the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	Circuit unit identification.
<i>Dump</i> =	rCKTDATA tuple.

Table 34 Error Code: SGNODD

Error Code:	SGNODD
Error Description:	The TSI audit was unable to read the corresponding static rIPLI_LNK tuple for a PLI link.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	None.
Manual Action Required:	The missing rIPLI_LNK tuple needs to be inserted by recent change. The tsi_link field in the audit dump is the key of the missing rIPLI_LNK tuple.
Dump Description:	
<i>Error Address</i> =	The address of the buffer used to read the rIPLI_LNK tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The tsi_link field containing the TSI link number.
<i>Dump</i> =	The buffer used to read the rIPLI_LNK tuple.

Table 35 Error Code: STATE_VAR

Error Code:	STATE_VAR
Error Description:	The inproc state of the specified rICHDB tuple is not recognized by audits.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to the rICHDB tuple.
<i>Bad Data</i> =	The inproc state of the rICHDB tuple.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	TSI time slot number.
<i>Dump</i> =	rICHDB tuple.

Table 36 Error Code: TRB_WRD

Error Code:	TRB_WRD
Error Description:	The ADR input source selection has been corrupted. It is set to take input from the network side, which is an illegal setting.

Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle value is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Trouble word for series 2000/3000.
<i>Good Data =</i>	Good idle value: TS_TRB.
<i>Logical Index =</i>	0.
<i>Dump =</i>	ADR information structure of the type tsADRCONFIG.

Table 37 Error Code: TS13_OPTS

Error Code:	TS13_OPTS
Error Description:	The value of command register 3 (CMD3) in the TSI or the value of fixed-address ROM variable SMtsiopt does not match the value of global Gltsiopt. Compare the values of SMtsiopt and CMD3 to the value of Gltsiopt to determine who the culprit(s) is(are).
Possible Error Effect:	The TSI hardware is not configured according to the options that are to be active according to the ODD.
Corrective Action Taken:	The correct value has been written to SMtsiopt and CMD3.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	0.
<i>Bad Data =</i>	Value of CMD3 BEFORE correction.
<i>Good Data =</i>	Gltsiopt. If this value is 0xffff, then the BAD DATA is meaningless.
<i>Logical Index =</i>	Value of SMtsiopt BEFORE correction.
	NOTE: See file hdr/smim/SMts_cmd3.h for a layout of CMD3. The bits in these registers use negative logic: 0 = Option ENABLED. 1 = Option DISABLED.

Table 38 Error Code: TSMUX_TRBWRD

Error Code:	TSMUX_TRBWRD
Error Description:	The TS_MUX trouble word constant in the alternate data RAM (ADR) has been corrupted.
Possible Error Effect:	Invalid PCM Trouble word may be sent on channel.
Corrective Action Taken:	The correct trouble word constant is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rICHDB.
<i>Bad Data =</i>	Bad TS_MUX trouble word constant.
<i>Good Data =</i>	Good TS_MUX trouble word constant.
<i>Logical Index =</i>	Trouble word channel number.
<i>Dump 1 =</i>	rICHDB.
<i>Dump 2 =</i>	ADR information structure of the type tsADRCONFIG.

Table 39 Error Code: TSNO_MATCH

Error Code:	TSNO_MATCH
Error Description:	The TSI RAM is not configured correctly.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The TSI RAM configuration is corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	rPICB buffer's unit number (unitnum).
<i>Bad Data =</i>	AUTRUE for universal tone decoder (UTD) case. .LP AUFALSE for universal tone generator (UTG) case.

<i>Good Data</i> =	Circuit name.
<i>Logical Index</i> =	TSI time slot number.
<i>Dump 1</i> =	tsTSICONFG buffer containing TSI control RAM data.
<i>Dump 2</i> =	riCDBCOM tuple.

Table 40 Error Code: TSTRB_TRBWRD

Error Code:	TSTRB_TRBWRD
Error Description:	The TS_TRB trouble word constant in the ADR has been corrupted.
Possible Error Effect:	Invalid PCM trouble word may be sent on channel.
Corrective Action Taken:	The correct trouble word constant is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of riCHDB.
<i>Bad Data</i> =	Bad TS_TRB trouble word constant.
<i>Good Data</i> =	Good TS_TRB trouble word constant.
<i>Logical Index</i> =	Trouble word channel number.

Table 41 Error Code: TSUMC_TRBWRD

Error Code:	TSUMC_TRBWRD
Error Description:	The TS_UMC trouble word constant in the ADR has been corrupted.
Possible Error Effect:	Invalid PCM Trouble word may be sent on channel.
Corrective Action Taken:	The correct trouble word constant is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of riCHDB.
<i>Bad Data</i> =	Bad TS_UMC trouble word constant.
<i>Good Data</i> =	Good TS_UMC trouble word constant.
<i>Logical Index</i> =	Trouble word channel number.
<i>Dump 1</i> =	riCHDB.
<i>Dump 2</i> =	ADR information structure of the type tsADRCONFG.

Table 42 Error Code: TS_USAGE

Error Code:	TS_USAGE
Error Description:	The TSI control RAM bits associated with the specified idle network time slot is incorrect.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	TSI control RAMs are idled.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Pointer to riTSDB tuple.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	The riTSDB tuple's RSM time slot status.
<i>Logical Index</i> =	TSI time slot number.
<i>Dump 1</i> =	riTSDB tuple.
<i>Dump 2</i> =	tsTSICONFG buffer containing TSI RAM information.

Table 43 Error Code: UNC_TRB

Error Code:	UNC_TRB
Error Description:	The ADR input source selection has been corrupted. It is set to take input from the network side, which is an illegal setting.
Possible Error Effect:	This error might cause unusual error conditions in call processing.
Corrective Action Taken:	The correct idle value is written to the ADR.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0.
<i>Bad Data</i> =	DDS unassigned multiplex code.
<i>Good Data</i> =	Good idle value: TS_UMC.
<i>Logical Index</i> =	0.
<i>Dump</i> =	ADR information structure of the type tsADRCONFG.

Table 44 Error Code: WR_XDSU

Error Code:	WR_XDSU
Error Description:	An idle channel has its extended DSU write enable in something but the idle state. This is indicative of a call not cleaning up after itself properly after termination.
Possible Error Effect:	This error may cause unusual and unpredictable conditions to occur during call processing.
Corrective Action Taken:	The erroneous data has been corrected.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the offending CHDB tuple.
<i>Bad Data =</i>	Actual contents of pc_adr_config.xdsu_wr_add.
<i>Good Data =</i>	Correct idle value.
<i>Logical Index =</i>	Key of the offending CHDB tuple.
<i>Dump 1 =</i>	riCHDB.
<i>Dump 2 =</i>	ADR information structure of the type tsADRCONFIG.

Table 45 Error Code: WRITE

Error Code:	WRITE
Error Description:	The path to the ADR was left in the connected state for this idle time slot.
Possible Error Effect:	The time slot that is now using this path may pick up noise on the line.
Corrective Action Taken:	Idle the connection to the ADR for this time slot.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of CHDB local pointer.
<i>Bad Data =</i>	ADR write address to the ADR.
<i>Good Data =</i>	ADR dump address.
<i>Logical Index =</i>	Key of CHDB.
<i>Dump 1 =</i>	riCHDB tuple.
<i>Dump 2 =</i>	ADR information structure of the type tsADRCONFIG.

Table 46 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit was entered to clean up after taking a reset.
Possible Error Effect:	This error can cause resets and asserts. Refer to the Asserts Manual (235-600-500) for more information on asserts.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

TTF

Software Release: 5E10 and later

1. TRANSMISSION TEST FACILITY (TTF) AUDIT

This audit is part of the peripheral control audits subgroup. It runs in the OSDS-M environment in the segmented mode.

2. DATA STRUCTURES

The TTF audit uses the data in RLCKTDATA to verify the existence of the dynamic transmission test relations: RLCOM_TTF, RLGEN_TTF, RLRSP_TTF, RLTD_TTF, and RLTTT_TTF. The transmission test facility relations contain information such as the circuit name for a given transmission test facility circuit. Figure 1 shows the layout of the transmission test facility relations.

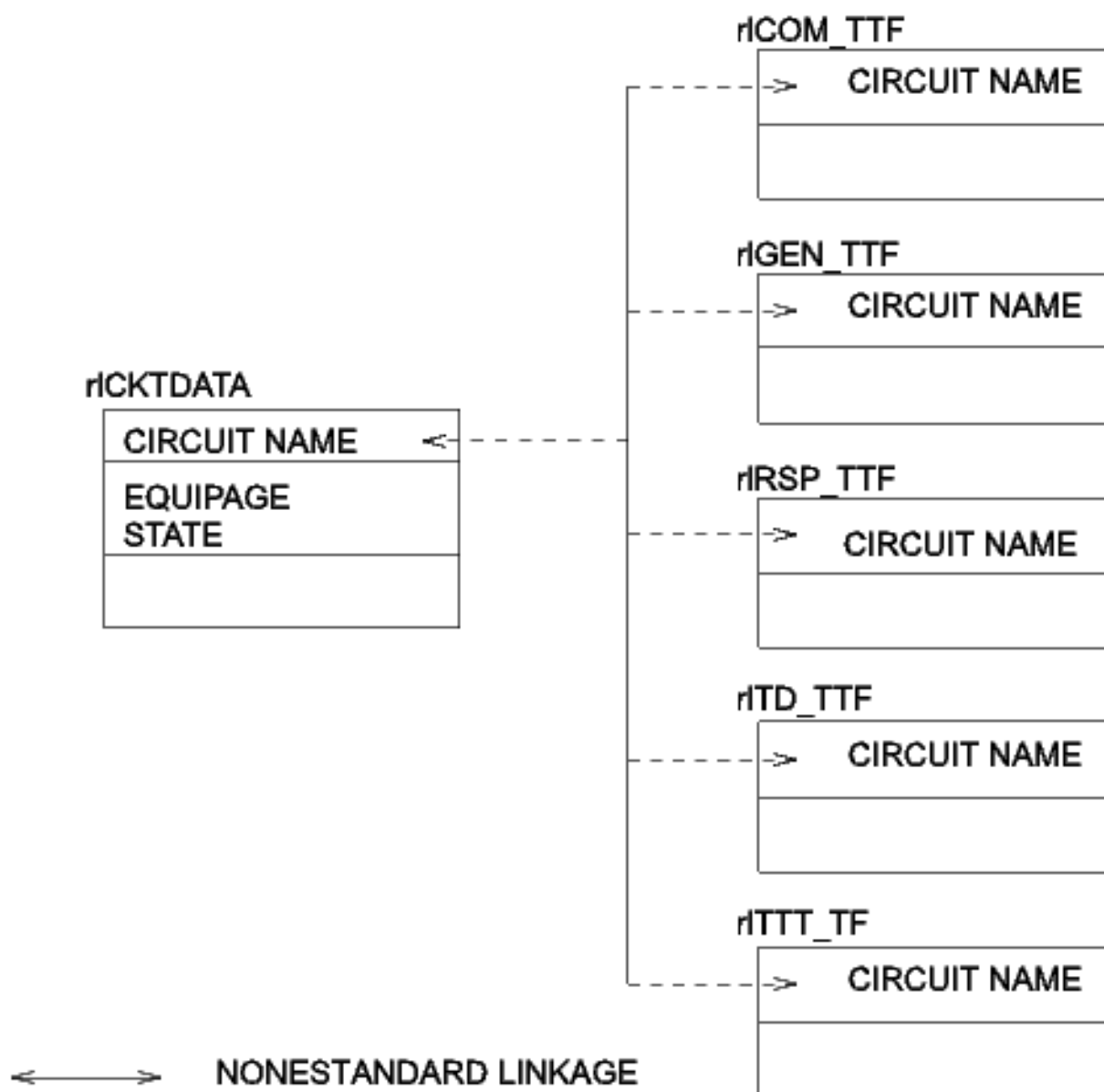


Figure 1 (TTF-1) Data Structures of the TTF Audit**3. AUDIT CHECKS**

The TTF audit uses the static driver (DRSTATIC) to step through each transmission test facility relation. For each tuple in RLCOM_TTF, RLGEN_TTF, RLRSP_TTF, RLTD_TTF, and RLTTT_TTF, the audit performs the following checks:

- (a) Verifies that a corresponding tuple exists in RLCKTDATA.
- (b) Checks the equipage state of RLCKTDATA. If the state is unequipped (UNEQ), no transmission test facility tuple should exist for the corresponding circuit.

4. ERROR HANDLING

Any tuple in a transmission test facility relation that does not correspond to a tuple in RLCKTDATA, or has a corresponding rLCKTDATA tuple in the unequipped state (UNEQ), is deleted and an error is reported.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	A ttf tuple exist but the static relation RLCKTDATA indicates that this rLTTF tuple should not exist.
Possible Error Effect:	The circuit may be erroneously used.
Corrective Action Taken:	The rLTTF tuple is deleted from the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLTTF tuple.
<i>Bad Data =</i>	RLTTF relation identification.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	ttf circuit name.

TXNMGR

Software Release: 5E10 and later

1. TRANSACTION MANAGER (TXNMGR) AUDIT

Transaction capabilities application part (TCAP) processing uses transaction management functionality to manage TCAP transactions between the service switching point (SSP) and the service control point (SCP) for advanced services platform (ASP) calls.

Transaction management functionality provides the ability to associate a TCAP transaction with the process ID controlling the transaction. This association is maintained in a tuple of RLTXN_MGR.

2. DATA STRUCTURES

The RLTXN_MGR dynamic relation exists in the SM and serves as a transaction management table for the situations in which a TCAP transaction is being controlled by a process ID which is different than the process ID which opened the transaction (that is, transaction control has moved from one process ID to another). For each open transaction in this situation, this relation contains a tuple to provide an association from the TCAP transaction ID (hashed key of the tuple, **rITXN_MGR.near_tid**) to the process ID controlling the transaction (**rITXN_MGR.near_pid**).

Tuples are inserted into this hashed relation when this association is needed, and subsequently deleted when the TCAP transaction is closed. The process ID controlling the transaction resides on the same SM as the rITXN_MGR tuple.

The TXNMGR audit references no dynamic relations directly, but confirms that the process ID in **rITXN_MGR.near_pid** identifies an existing process on the same SM.

3. AUDIT CHECKS

The TXNMGR audit uses the DRSTATIC audit driver. For each rITXN_MGR tuple found in the relation, the audit makes the following check:

DEADPID = Check that the **near_pid** in rITXN_MGR is a live process.

4. ERROR HANDLING

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this document, for any error codes not found.

The audit produces the following error code(s).

Table 1 Error Code: DEADPID

Error Code:	DEADPID
Error Description:	The near_pid structure in the rITXN_MGR tuple under audit was null or did not contain the process ID for a live process.
Possible Error Effect:	Further transactions cannot be done against the process in the RLTXN_MGR tuple (rITXN_MGR.near_tid).
Corrective Action Taken:	The rITXN_MGR tuple is deleted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The logical address of the rITXN_MGR tuple.
<i>Bad Data =</i>	The contents of the near_pid structure in the RLTXN_MGR tuple.
<i>Good Data =</i>	N/A.
<i>Logical Index =</i>	The procno portion of the near_tid structure in the RLTXN_MGR tuple.

<i>Dump</i> =	The contents of the rITXN_MGR tuple where the error was found.

UBHC

Software Release: 5E10 and later

1. UNCONCENTRATED BUFFER HEAD CELL (UBHC) AUDIT

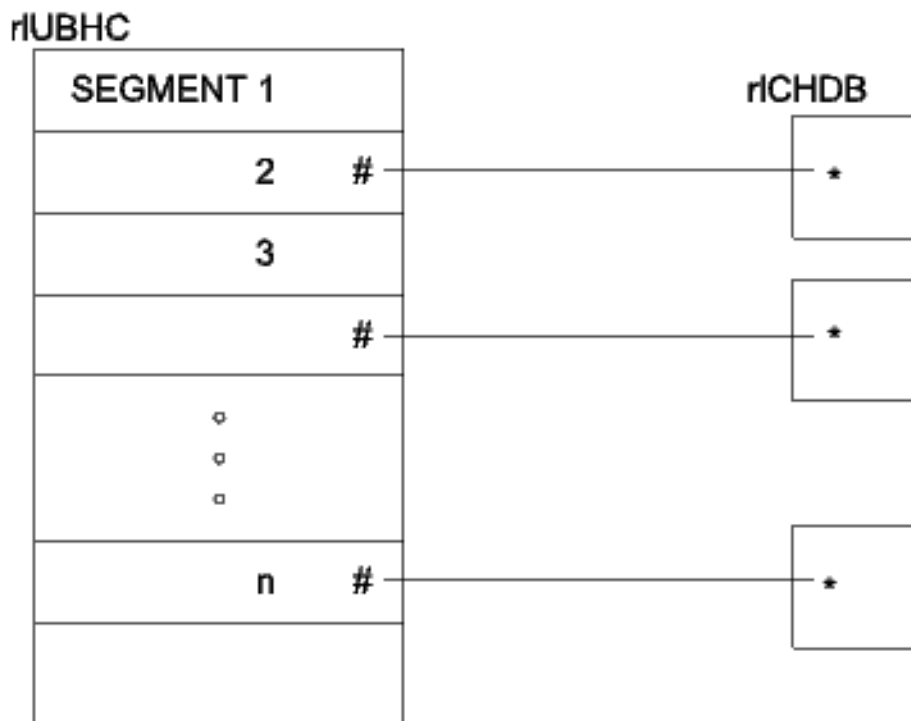
There are two kinds of input to a switch: lines and trunks. Lines are concentrated, which means they share hardware. Trunks are unconcentrated; they use as much hardware as needed to handle all the trunks in a simultaneous busy state. In both cases, there may be more inputs to the switch during a heavy call processing load than can be processed. Excess inputs are buffered onto head cells.

Each buffer head cell represents a one-second segment of time during which an input may have occurred; there are 20 one-second segments. Each unconcentrated buffer head cell owns one or more channel data blocks, which correspond to one channel dedicated to a trunk circuit. If a trunk origination must be buffered, the channel data block is linked to the unconcentrated buffer head cell that represents the time segment in which the origination occurred. The global variable PCubtotal contains the number of channel data blocks that are linked to the unconcentrated buffer head cell.

The unconcentrated buffer head cell audit (UBHC) prevents the loss of queued originations and linked data blocks. This audit is part of the Peripheral Control audits subgroup. It runs in the OSDS-M environment in the segmented mode.

2. DATA STRUCTURES

This audit operates on the unconcentrated buffer head cell relation (RLUBHC). The RLUBHC relation has an owner linkage with the channel data block relation (RLCHDB). Figure 1 illustrates this linkage.



SM

AM

_____ * = OWNER-MEMBER LINKAGE

Figure 1 (UBHC-1) rIUBHC Linkages

3. AUDIT CHECKS

To verify the linkages associated with the unconcentrated buffer head cells, the audit calls the linkage driver (DRLINK) once for each audit segment. The audit checks each head cell in order using the physical key to step through the relation.

After the linkages are verified, the driver invokes the UBHC semantic check function to perform a check on the number of channel data blocks (rICHDBs) linked to an unconcentrated buffer head cell (rIUBHC). The number should be consistent with the number in the global variable PCubtotal.

4. ERROR HANDLING

Any rIUBHC tuples that fail these checks are reinitialized. The logical key field is set to its correct value.

If the number of rICHDB tuples linked to an rIUBHC is inconsistent with the global variable PCubtotal, the audit prints an error message and corrects PCubtotal.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNTS

Error Code:	COUNTS
Error Description:	1) There is not any buffered rCHDB tuples linked on the active rUBHC head cell's, but PCubtotal is not zero. 2) There is buffered rCHDBs linked on the active rUBHC head cell's and PCubtotal is zero. 3) There are more buffered rCHDBs than PCubtotal shows.
Possible Error Effect:	May cause the system to go into overload.
Corrective Action Taken:	Set PCubtotal to zero for error description 1. Set PCubtotal to the value of is linked for error description 2 and error description 3.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of PCubtotal.
<i>Bad Data</i> =	Contents of PCubtotal.
<i>Good Data</i> =	Contents of is linked.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

UNIT2RDT

Software Release: 5E12 and later

1. UNIT NUMBER TO REMOTE DATA TERMINAL (UNIT2RDT) AUDIT

The RLUNIT2RDT relation maps the unit number of a TR-303 RDT to the circuit name of TR-303 RDT (PCRDT303). This static relation is used to initialize several dynamic relations: RLRDTCPCDAT, RLALM_RDT, and RLLLM_RDT. The UNIT2RDT audit ensures that these dynamic relations exist in the database.

2. DATA STRUCTURES

For details on the fields contained in the RLUNIT2RDT relation, refer to the 5ESS® Translations Data Manual (235-600-1xx) and the 5ESS® Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: LOSTTUP

Error Code:	LOSTTUP
Error Description:	A dynamic tuple does not exist for the RDT circuit referenced in RLUNIT2RDT.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	The dynamic tuple is inserted in the database.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of dynamic tuple buffer.
<i>Bad Data =</i>	Relation ID that is missing a tuple: RLRDTCPCDAT, RLALM_RDT, or RLLLM_RDT.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The key of the missing dynamic tuple.

Table 2 Error Code: ODDLOSTTUP

Error Code:	ODDLOSTTUP
Error Description:	A static tuple in RLCKTDATA does not exist for the RDT circuit referenced in RLUNIT2RDT.
Possible Error Effect:	May not be able to properly set up calls.
Corrective Action Taken:	NA.
Manual Action Required:	An rLCKTDATA tuple must exist for each RDT circuit. Refer to the logical key information for the circuit name.
Dump Description:	
<i>Error Address =</i>	RLCKTDATA relation ID value.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	The RDT circuit name as found in RLUNIT2RDT.
<i>Dump =</i>	rUNIT2RDT tuple.

UNITCFG-A

Software Release: 5E11 only

1. UNIT CONFIGURATION (UNITCFG) AUDIT

The UNITCFG audit will step through the RLPICB relation to search for each equipped access interface unit (AIU). For each equipped AIU the UNITCFG audit will insure the integrity of the SMaiuconfig[] array. The audit will also insure the integrity of such AIU hardware registers as the common data and control (COMDAC) configuration register (CX_CONFIG), the firmware configuration register (FW_CONFIG), the COMDAC even application pack (AP) side select register 0 (CX_EAS_0), the COMDAC even AP side select register 1 (CX_EAS_1), the COMDAC odd AP side select register 0 (CX_OAS_0), and the COMDAC odd AP side select register 1 (CX_OAS_1).

2. DATA STRUCTURES

For details on fields contained in the dynamic data audited, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACTBASE

Error Code:	ACTBASE
Error Description:	Either the active major (act_major) bit in the SMaiuconfig[] array is set to 1 when the corresponding rSMEST tuple indicates that the service group (SG) 0 COMDAC is active major (rSMEST.bas_stat), or the active major (act_major) bit in the SMaiuconfig[] array is set to 0 when the corresponding rSMEST tuple indicates that the SG 1 COMDAC is active major.
Possible Error Effect:	Call processing may be trying to set up call paths through an out-of-service (OOS) COMDAC.
Corrective Action Taken:	The active major (act_major) bit in the SMaiuconfig[] array is set to 0 if the rSMEST tuple for the SG 0 COMDAC is active major (rSMEST.bas_stat) or the active major (act_major) bit in the SMaiuconfig[] array is set to 1 if the rSMEST tuple for the SG 1 COMDAC is active major.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMaiuconfig[].
<i>Bad Data =</i>	Contents of the active major (act_major) bit.
<i>Good Data =</i>	Correct value for the active major (act_major) bit.
<i>Logical Index =</i>	AIU unit number.
<i>Dump 1 =</i>	rSMEST tuple of the SG 0 COMDAC.
<i>Dump 2 =</i>	rSMEST tuple of the SG 1 COMDAC.

Table 2 Error Code: ACTIVE_BAD

Error Code:	ACTIVE_BAD
Error Description:	The call processing inhibited (CPI) bit, which shows contention for a circuit, is set for either an active common control (CC) or active mate CC, with a status of idle.

Possible Error Effect:	Call processing is inhibited for this unit.
Corrective Action Taken:	The audit will reset the CPI bit to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value of the CPI bit.
<i>Good Data =</i>	DBNO.
<i>Logical Index =</i>	Line unit number.
<i>Dump 1 =</i>	Active CC rISMEST tuple.
<i>Dump 2 =</i>	Mate CC rISMEST tuple.

Table 3 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The audit has detected a missing rICDBCOM tuple from the database or has detected a rICDBCOM tuple with a corrupt logical key.
Possible Error Effect:	This error may cause asserts to fire from the application.
Corrective Action Taken:	The audit will schedule the CKTDATA audit to insert the missing rICDBCOM tuple or will schedule the CDBCOM audit to recover the corrupt logical key in the rICDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rICDBCOM tuple.
<i>Bad Data =</i>	AIU unit number.
<i>Good Data =</i>	Application pack number.
<i>Logical Index =</i>	Application pack circuit name.
<i>Dump 1 =</i>	NA.

Table 4 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The D-channel configuration attribute (dchan) in the SMaiu_config[] array is not consistent with the D-channel configuration attribute (dchan) in the corresponding static rIAIUPIDB tuple.
Possible Error Effect:	Setup of D-channel paths may only be simplex instead of duplexed.
Corrective Action Taken:	The audit will reset the D-channel configuration attribute (dchan) in the SMaiu_config[] array to the D-channel configuration attribute (dchan) from the static rIAIUPIDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SMaiu_config[] array entry.
<i>Bad Data =</i>	D-channel configuration attribute (dchan) from the SMaiu_config[] array.
<i>Good Data =</i>	D-channel configuration attribute (dchan) from the rIAIUPIDB tuple.
<i>Logical Index =</i>	AIU unit number.
<i>Dump =</i>	NA.

Table 5 Error Code: HWSW_CONFIG

Error Code:	HWSW_CONFIG
Error Description:	This error indicates one of two conditions. (1) The maintenance status (rISMEST.bas_stat) in the rISMEST tuple for the COMDAC indicates that the circuit is in an active major state, but the COMDAC master (CX_MASTER) bit in the COMDAC configuration register (CX_CONFIG) indicates that the COMDAC is receiving clock from the mate COMDAC. (2) The maintenance status (rISMEST.bas_stat) in the rISMEST tuple for the COMDAC indicates that the circuit is in an active minor state but the CX_MASTER bit in the CX_CONFIG register indicates that the COMDAC is receiving clock from peripheral interface data bus (PIDB) 0.

Possible Error Effect:	Synchronization errors may occur over the AIU PIDBs connected to the COMDAC that is in error.
Corrective Action Taken:	The COMDAC is placed into the maintenance request administrator (MRA) recovery to be restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the COMDAC rSMEST tuple.
<i>Bad Data =</i>	Contents of the CX_MASTER bit in the CX_CONFIG register.
<i>Good Data =</i>	Correct value of the CX_MASTER bit in the CX_CONFIG register.
<i>Logical Index =</i>	Circuit name of the COMDAC.
<i>Dump =</i>	rSMEST tuple.

Table 6 Error Code: INHBOTH

Error Code:	INHBOTH
Error Description:	Both COMDACs associated with the AIU are OOS, but the call processing inhibit bit (cp_inh) or the routine maintenance inhibit bit (rm_inh) is not set to inhibit in the SMaiuconfig[] array.
Possible Error Effect:	Call processing or maintenance activities may try to use this AIU causing peripheral interrupts.
Corrective Action Taken:	The call processing inhibit bit (cp_inh) and the routine maintenance inhibit bit (rm_inh) in the SMaiuconfig[] array is set to inhibit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMaiuconfig[].
<i>Bad Data =</i>	Contents of the routine maintenance inhibit bit (rm_inh).
<i>Good Data =</i>	Contents of the call processing inhibit bit (cp_inh).
<i>Logical Index =</i>	AIU unit number.
<i>Dump 1 =</i>	rSMEST tuple of the service group (SG) 0 COMDAC.
<i>Dump 2 =</i>	rSMEST tuple of the SG 1 COMDAC.

Table 7 Error Code: INVTSTMP

Error Code:	INVTSTMP
Error Description:	Add invalid time was derived from operating system (OS): time appears to be moving backward compared to global data.
Possible Error Effect:	Verification of routine maintenance inhibit (RMI) can not be performed.
Corrective Action Taken:	The audit will reset our global time management data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of AUice_time.
<i>Bad Data =</i>	The value of AUice_time.lo_time.
<i>Good Data =</i>	The value of AUice_time.hi_time.
<i>Logical Index =</i>	The difference with OSnow.
<i>Dump =</i>	The current time according to OSnow.

Table 8 Error Code: LAMP

Error Code:	LAMP
Error Description:	The maintenance status (rSMEST.bas_stat) in the rSMEST tuple for the COMDAC indicates that the circuit is in an active major or active minor state, but the light emitting diode (LED) lamp bit (CX_LED OFF) in the COMDAC configuration register (CX_CONFIG) indicates that the COMDAC is OOS.
Possible Error Effect:	The LED lamp bit (CX_LED OFF) in the COMDAC configuration register (CX_CONFIG) is set to indicate that the COMDAC pack is IS (in-service).
Corrective Action Taken:	The light emitting diode report (CX_LED) bit in the COMDAC firmware configuration register (CX_FWCONF) is set to indicate that LED reports are allowed to be forwarded from the COMDAC to the SMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the COMDAC rSMEST tuple.
<i>Bad Data =</i>	Contents of the CX_LED OFF bit in the CX_CONFIG register.
<i>Good Data =</i>	Correct value of the CX_LED OFF bit in the CX_CONFIG register.
<i>Logical Index =</i>	Circuit name of the COMDAC.

<i>Dump</i> =	rISMEST tuple.

Table 9 Error Code: RA_ENABLE

Error Code:	RA_ENABLE
Error Description:	The broadband enable bit in the SMaiufig[] array is inconsistent with the change level indicator (CLI) in the corresponding rUNITDATA tuple.
Possible Error Effect:	Interrupts may occur due to software operations on the wrong type of AIU backplane.
Corrective Action Taken:	The broadband enable bit in the SMaiufig[] array is made consistent with the CLI value in the corresponding rUNITDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	Contents of the broadband enable bit.
<i>Good Data</i> =	Correct value for the broadband enable bit.
<i>Logical Index</i> =	AIU unit number.
<i>Dump</i> =	NA.

Table 10 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	Access of the rISMEST tuple for an AP, SG 0 COMDAC, or SG 1 COMDACs associated with the AIU under audit has failed.
Possible Error Effect:	The UNITCFG audit will skip auditing the SMaiufig[] array associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rISMEST tuple.
<i>Bad Data</i> =	AIU unit number.
<i>Good Data</i> =	AP number or COMDAC SG number.
<i>Logical Index</i> =	Circuit name of the AP or COMDAC.
<i>Dump</i> =	rPICB tuple.

Table 11 Error Code: SALW

Error Code:	SALW
Error Description:	Both COMDACs associated with the AIU are in an active state, but the call processing inhibit bit (cp_inh) or the routine maintenance inhibit bit (rm_inh) is set to inhibit in the SMaiufig[] array.
Possible Error Effect:	Call processing or maintenance activities are blocked on the AIU unit.
Corrective Action Taken:	The call processing inhibit bit (cp_inh) and the routine maintenance inhibit bit (rm_inh) in the SMaiufig[] array is set to allow.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	Contents of the routine maintenance inhibit bit (rm_inh).
<i>Good Data</i> =	Contents of the call processing inhibit bit (cp_inh).
<i>Logical Index</i> =	AIU unit number.
<i>Dump</i> =	rISMEST tuple of the SG 0 COMDAC.
<i>Dump</i> =	rISMEST tuple of the SG 1 COMDAC.

Table 12 Error Code: SCNMSK

Error Code:	SCNMSK
Error Description:	The maintenance state (rISMEST.bas_stat) in the rISMEST tuple for the COMDAC indicates that the circuit is in an active major or active minor state, but the allow scan report (CX_ALW_SCAN) bit in the COMDAC firmware configuration register (CX_FWCONF) indicates that scan reports are inhibited from being forwarded to the switching module processor (SMP) from the COMDAC.
Possible Error Effect:	AIU port will not be able to originate a call.
Corrective Action Taken:	The allow scan report (CX_ALW_SCAN) bit in the COMDAC firmware

	configuration register (CX_FWCONF) is set to indicate that scan reports are allowed to be forwarded from the COMDAC to the SMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the COMDAC rSMEST tuple.
<i>Bad Data</i> =	Contents of the CX_ALW_SCAN bit in the CX_CONFIG register.
<i>Good Data</i> =	Correct value of the CX_ALW_SCAN bit in the CX_CONFIG register.
<i>Logical Index</i> =	Circuit name of the COMDAC.
<i>Dump</i> =	rSMEST tuple.

Table 13 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The static rPICB tuple exists in the database but the corresponding rIAUIPDB tuple is missing from the database.
Possible Error Effect:	All call processing and maintenance activities on this AIU may assert.
Corrective Action Taken:	None.
Manual Action Required:	The missing rIAUIPDB tuple must be insert into the database using the PICB attribute from the static rPICB tuple, which is dumped as the <i>Logical Index</i> .
Dump Description:	
<i>Error Address</i> =	Address of the SMaiu_config[] array entry.
<i>Bad Data</i> =	Relation identification of the RLAIUIPDB relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	System PICB number.
<i>Dump</i> =	rPICB tuple.

Table 14 Error Code: SELECT_BIT_BAD

Error Code:	SELECT_BIT_BAD
Error Description:	The AP side select (ap_side_select) bits in the SMaiuconfig[] array does not match the status of the AP side select bits in either the even AP side select register 0 (CX_EAS_O), even AP side select register 1 (CX_EAS_1), odd AP side select register 0 (CX_OAS_0), or odd AP side select register 1 (CX_OAS_1).
Possible Error Effect:	The AP is listening to the wrong COMDAC, causing AP hardware jobs to fail.
Corrective Action Taken:	The COMDAC is placed into the MRA recovery to be restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiuconfig[].
<i>Bad Data</i> =	Contents of the AP side select bits from SMaiuconfig[].
<i>Good Data</i> =	Correct value of the AP side select bits.
<i>Logical Index</i> =	AIU unit number in the upper 16 bits and AP number in the lower 16 bits.
<i>Dump 1</i> =	SMaiuconfig[] array entry.
<i>Dump 2</i> =	Contents of the AP side select bits from the service group (SG) 0 COMDAC.
<i>Dump 3</i> =	Contents of the AP side select bits from the SG 1 COMDAC.

Table 15 Error Code: SMEST

Error Code:	SMEST
Error Description:	Access of the rSMEST tuple for either the SG 0 COMDAC, or the SG 1 COMDAC associated with the AIU under audit has failed.
Possible Error Effect:	The UNITCFG audit will skip auditing the SMaiuconfig[] array associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple for SG 0 COMDAC.
<i>Bad Data</i> =	Address of the rSMEST tuple for SG 1 COMDAC.
<i>Good Data</i> =	Circuit name of the SG 0 COMDAC.
<i>Logical Index</i> =	Circuit name of the SG 1 COMDAC.
<i>Dump</i> =	rPICB tuple.

Table 16 Error Code: STUCK

Error Code:	STUCK

Error Description:	The COMDAC block job bit (cd_blk_jobs) or the digital loop back inhibit bit (dlb_inh) in the SMaiufig[] array is stuck set (DBYES). The <i>Bad Data</i> field contains information on which of these bits were found to be corrupt.
Possible Error Effect:	If the COMDAC block job bit is in error, then the error effect is that all COMDAC hardware jobs are inhibited from being sent from the switching module processor (SMP) to the AIU, blocking all call processing and maintenance activities associated with that COMDAC. If the digital loop back inhibit bit is in error, then the error effect is that all digital loop back failures are inhibited from being reported.
Corrective Action Taken:	The COMDAC block job (cd_blk_jobs) and digital loop back inhibit (dlb_inh) bits in the SMaiufig[] array are cleared.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	0 if the COMDAC block job bit is corrupt or 1 if the digital loop back inhibit bit is corrupt.
<i>Good Data</i> =	If <i>Bad Data</i> is 0, then this is the COMDAC SG number. Otherwise, <i>Good Data</i> is NA.
<i>Logical Index</i> =	AIU unit number.
<i>Dump</i> =	SMaiufig[] array entry found to be in error.

Table 17 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	The AP block job bit (ap_blk_jobs) in the SMaiufig[] array is stuck set for one of the AP circuits.
Possible Error Effect:	All AP hardware jobs are inhibited from being sent from the SMP to the AIU, blocking all call processing and maintenance activities associated with that AP circuit.
Corrective Action Taken:	The AP block job bit (ap_blk_jobs) in the SMaiufig[] array is cleared for the AP circuit allowing AP hardware jobs to be sent from the SMP to the AIU.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	AP number.
<i>Logical Index</i> =	AIU unit number.
<i>Dump 1</i> =	SMaiufig[] array entry found to be in error.
<i>Dump 2</i> =	rSMEST tuple associated with the AP circuit.

Table 18 Error Code: STUCKBUSY

Error Code:	STUCKBUSY
Error Description:	The false ring trip inhibit bit (frt_inh) in the SMaiufig[] array is stuck set for one of the ring generator (RG) circuits.
Possible Error Effect:	All false ring trip reports are inhibit causing faults in the ring generators to not be reported.
Corrective Action Taken:	The false ring trip inhibit bit (frt_inh) in the SMaiufig[] array is cleared for the RG circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	RG number.
<i>Good Data</i> =	AP number.
<i>Logical Index</i> =	AIU number.
<i>Dump 1</i> =	SMaiufig[] array entry found to be in error.
<i>Dump 2</i> =	rCDBCOM tuple associated with the RG circuit.

UNITCFG-B

Software Release: 5E12 and later

1. UNIT CONFIGURATION (UNITCFG) AUDIT

The UNITCFG audit will step through the RLPICB relation to search for each equipped access interface unit (AIU). For each equipped AIU the UNITCFG audit will insure the integrity of the SMaiuconfig[] array. The audit will also insure the integrity of such AIU hardware registers as the common data and control (COMDAC) configuration register (CX_CONFIG), the firmware configuration register (FW_CONFIG), the COMDAC even application pack (AP) side select register 0 (CX_EAS_0), the COMDAC even AP side select register 1 (CX_EAS_1), the COMDAC odd AP side select register 0 (CX_OAS_0), and the COMDAC odd AP side select register 1 (CX_OAS_1).

2. DATA STRUCTURES

For details on fields contained in the dynamic data audited, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACTBASE

Error Code:	ACTBASE
Error Description:	Either the active major (act_major) bit in the SMaiuconfig[] array is set to 1 when the corresponding rSMEST tuple indicates that the service group (SG) 0 COMDAC is active major (rSMEST.bas_stat), or the active major (act_major) bit in the SMaiuconfig[] array is set to 0 when the corresponding rSMEST tuple indicates that the SG 1 COMDAC is active major.
Possible Error Effect:	Call processing may be trying to set up call paths through an out-of-service (OOS) COMDAC.
Corrective Action Taken:	The active major (act_major) bit in the SMaiuconfig[] array is set to 0 if the rSMEST tuple for the SG 0 COMDAC is active major (rSMEST.bas_stat) or the active major (act_major) bit in the SMaiuconfig[] array is set to 1 if the rSMEST tuple for the SG 1 COMDAC is active major.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMaiuconfig[].
<i>Bad Data =</i>	Contents of the active major (act_major) bit.
<i>Good Data =</i>	Correct value for the active major (act_major) bit.
<i>Logical Index =</i>	AIU unit number.
<i>Dump 1 =</i>	rSMEST tuple of the SG 0 COMDAC.
<i>Dump 2 =</i>	rSMEST tuple of the SG 1 COMDAC.

Table 2 Error Code: ACTIVE_BAD

Error Code:	ACTIVE_BAD
Error Description:	The call processing inhibited (CPI) bit, which shows contention for a circuit, is set for either an active common control (CC) or active mate CC, with a status of idle.

Possible Error Effect:	Call processing is inhibited for this unit.
Corrective Action Taken:	The audit will reset the CPI bit to DBNO.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The value of the CPI bit.
<i>Good Data =</i>	DBNO.
<i>Logical Index =</i>	Line unit number.
<i>Dump 1 =</i>	Active CC rSMEST tuple.
<i>Dump 2 =</i>	Mate CC rSMEST tuple.

Table 3 Error Code: CDBCOM

Error Code:	CDBCOM
Error Description:	The audit has detected a missing rCDBCOM tuple from the database or has detected a rCDBCOM tuple with a corrupt logical key.
Possible Error Effect:	This error may cause asserts to fire from the application.
Corrective Action Taken:	The audit will schedule the CKTDATA audit to insert the missing rCDBCOM tuple or will schedule the CDBCOM audit to recover the corrupt logical key in the rCDBCOM tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rCDBCOM tuple.
<i>Bad Data =</i>	AIU unit number.
<i>Good Data =</i>	Application pack number.
<i>Logical Index =</i>	Application pack circuit name.
<i>Dump 1 =</i>	NA.

Table 4 Error Code: DCHAN

Error Code:	DCHAN
Error Description:	The D-channel configuration attribute (dchan) in the SMaiu_config[] array is not consistent with the D-channel configuration attribute (dchan) in the corresponding static rAIUPIDB tuple.
Possible Error Effect:	Setup of D-channel paths may only be simplex instead of duplexed.
Corrective Action Taken:	The audit will reset the D-channel configuration attribute (dchan) in the SMaiu_config[] array to the D-channel configuration attribute (dchan) from the static rAIUPIDB tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the SMaiu_config[] array entry.
<i>Bad Data =</i>	D-channel configuration attribute (dchan) from the SMaiu_config[] array.
<i>Good Data =</i>	D-channel configuration attribute (dchan) from the rAIUPIDB tuple.
<i>Logical Index =</i>	AIU unit number.
<i>Dump =</i>	NA.

Table 5 Error Code: HWSW_CONFIG

Error Code:	HWSW_CONFIG
Error Description:	This error indicates one of two conditions. 1 = The maintenance status (rSMEST.bas_stat) in the rSMEST tuple for the COMDAC indicates that the circuit is in an active major state, but the COMDAC master (CX_MASTER) bit in the COMDAC configuration register (CX_CONFIG) indicates that the COMDAC is receiving clock from the mate COMDAC. 2 = The maintenance status (rSMEST.bas_stat) in the rSMEST tuple for the COMDAC indicates that the circuit is in an active minor state but the CX_MASTER bit in the CX_CONFIG register indicates that the COMDAC is receiving clock from peripheral interface data bus (PIDB) 0.
Possible Error Effect:	Synchronization errors may occur over the AIU PIDBs connected to the

	COMDAC that is in error.
Corrective Action Taken:	The COMDAC is placed into the maintenance request administrator (MRA) recovery to be restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the COMDAC rSMEST tuple.
<i>Bad Data =</i>	Contents of the CX_MASTER bit in the CX_CONFIG register.
<i>Good Data =</i>	Correct value of the CX_MASTER bit in the CX_CONFIG register.
<i>Logical Index =</i>	Circuit name of the COMDAC.
<i>Dump =</i>	rSMEST tuple.

Table 6 Error Code: INHBOTH

Error Code:	INHBOTH
Error Description:	Both COMDACs associated with the AIU are OOS, but the call processing inhibit bit (cp_inh) or the routine maintenance inhibit bit (rm_inh) is not set to inhibit in the SMaiuconfig[] array.
Possible Error Effect:	Call processing or maintenance activities may try to use this AIU causing peripheral interrupts.
Corrective Action Taken:	The call processing inhibit bit (cp_inh) and the routine maintenance inhibit bit (rm_inh) in the SMaiuconfig[] array is set to inhibit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMaiuconfig[].
<i>Bad Data =</i>	Contents of the routine maintenance inhibit bit (rm_inh).
<i>Good Data =</i>	Contents of the call processing inhibit bit (cp_inh).
<i>Logical Index =</i>	AIU unit number.
<i>Dump 1 =</i>	rSMEST tuple of the service group (SG) 0 COMDAC.
<i>Dump 2 =</i>	rSMEST tuple of the SG 1 COMDAC.

Table 7 Error Code: INVTSTMP

Error Code:	INVTSTMP
Error Description:	Add invalid time was derived from operating system (OS): time appears to be moving backward compared to global data.
Possible Error Effect:	Verification of routine maintenance inhibit (RMI) can not be performed.
Corrective Action Taken:	The audit will reset our global time management data.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of AUice_time.
<i>Bad Data =</i>	The value of AUice_time.lo_time.
<i>Good Data =</i>	The value of AUice_time.hi_time.
<i>Logical Index =</i>	The difference with OSnow.
<i>Dump =</i>	The current time according to OSnow.

Table 8 Error Code: LAMP

Error Code:	LAMP
Error Description:	The maintenance status (rSMEST.bas_stat) in the rSMEST tuple for the COMDAC indicates that the circuit is in an active major or active minor state, but the light emitting diode (LED) lamp bit (CX_LED OFF) in the COMDAC configuration register (CX_CONFIG) indicates that the COMDAC is OOS.
Possible Error Effect:	The LED lamp bit (CX_LED OFF) in the COMDAC configuration register (CX_CONFIG) is set to indicate that the COMDAC pack is in-service (IS).
Corrective Action Taken:	The light emitting diode report (CX_LED) bit in the COMDAC firmware configuration register (CX_FWCONF) is set to indicate that LED reports are allowed to be forwarded from the COMDAC to the SMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the COMDAC rSMEST tuple.
<i>Bad Data =</i>	Contents of the CX_LED OFF bit in the CX_CONFIG register.
<i>Good Data =</i>	Correct value of the CX_LED OFF bit in the CX_CONFIG register.
<i>Logical Index =</i>	Circuit name of the COMDAC.
<i>Dump =</i>	rSMEST tuple.

Table 9 Error Code: RA_ENABLE

Error Code:	RA_ENABLE
Error Description:	The broadband enable bit in the SMainuconfig[] array is inconsistent with the change level indicator (CLI) in the corresponding rUNITDATA tuple.
Possible Error Effect:	Interrupts may occur due to software operations on the wrong type of AIU backplane.
Corrective Action Taken:	The broadband enable bit in the SMainuconfig[] array is made consistent with the CLI value in the corresponding rUNITDATA tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMainuconfig[].
<i>Bad Data =</i>	Contents of the broadband enable bit.
<i>Good Data =</i>	Correct value for the broadband enable bit.
<i>Logical Index =</i>	AIU unit number.
<i>Dump =</i>	NA.

Table 10 Error Code: READSMEST

Error Code:	READSMEST
Error Description:	Access of the rSMEST tuple for an AP, SG 0 COMDAC, or SG 1 COMDACs associated with the AIU under audit has failed.
Possible Error Effect:	The UNITCFG audit will skip auditing the SMainuconfig[] array associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rSMEST tuple.
<i>Bad Data =</i>	AIU unit number.
<i>Good Data =</i>	AP number or COMDAC SG number.
<i>Logical Index =</i>	Circuit name of the AP or COMDAC.
<i>Dump =</i>	rPICB tuple.

Table 11 Error Code: SALW

Error Code:	SALW
Error Description:	Both COMDACs associated with the AIU are in an active state, but the call processing inhibit bit (cp_inh) or the routine maintenance inhibit bit (rm_inh) is set to inhibit in the SMainuconfig[] array.
Possible Error Effect:	Call processing or maintenance activities are blocked on the AIU unit.
Corrective Action Taken:	The call processing inhibit bit (cp_inh) and the routine maintenance inhibit bit (rm_inh) in the SMainuconfig[] array is set to allow.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of SMainuconfig[].
<i>Bad Data =</i>	Contents of the routine maintenance inhibit bit (rm_inh).
<i>Good Data =</i>	Contents of the call processing inhibit bit (cp_inh).
<i>Logical Index =</i>	AIU unit number.
<i>Dump =</i>	rSMEST tuple of the SG 0 COMDAC.
<i>Dump =</i>	rSMEST tuple of the SG 1 COMDAC.

Table 12 Error Code: SCNMSK

Error Code:	SCNMSK
Error Description:	The maintenance state (rSMEST.bas_stat) in the rSMEST tuple for the COMDAC indicates that the circuit is in an active major or active minor state, but the allow scan report (CX_ALW_SCAN) bit in the COMDAC firmware configuration register (CX_FWCONF) indicates that scan reports are inhibited from being forwarded to the switching module processor (SMP) from the COMDAC.
Possible Error Effect:	AIU port will not be able to originate a call.
Corrective Action Taken:	The allow scan report (CX_ALW_SCAN) bit in the COMDAC firmware configuration register (CX_FWCONF) is set to indicate that scan reports are

	allowed to be forwarded from the COMDAC to the SMP.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the COMDAC rSMEST tuple.
<i>Bad Data</i> =	Contents of the CX_ALW_SCAN bit in the CX_CONFIG register.
<i>Good Data</i> =	Correct value of the CX_ALW_SCAN bit in the CX_CONFIG register.
<i>Logical Index</i> =	Circuit name of the COMDAC.
<i>Dump</i> =	rSMEST tuple.

Table 13 Error Code: SDATA_CON_ERR

Error Code:	SDATA_CON_ERR
Error Description:	The static rPICB tuple exists in the database but the corresponding rIAIUPIDB tuple is missing from the database.
Possible Error Effect:	All call processing and maintenance activities on this AIU may assert.
Corrective Action Taken:	None.
Manual Action Required:	The missing rIAIUPIDB tuple must be insert into the database using the PICB attribute from the static rPICB tuple, which is dumped as the <i>Logical Index</i> .
Dump Description:	
<i>Error Address</i> =	Address of the SMaiu_config[] array entry.
<i>Bad Data</i> =	Relation identification of the RLAIUPIDB relation.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	System PICB number.
<i>Dump</i> =	rPICB tuple.

Table 14 Error Code: SELECT_BIT_BAD

Error Code:	SELECT_BIT_BAD
Error Description:	The AP side select (ap_side_select) bits in the SMaiuconfig[] array does not match the status of the AP side select bits in either the even AP side select register 0 (CX_EAS_O), even AP side select register 1 (CX_EAS_1), odd AP side select register 0 (CX_OAS_0), or odd AP side select register 1 (CX_OAS_1).
Possible Error Effect:	The AP is listening to the wrong COMDAC, causing AP hardware jobs to fail.
Corrective Action Taken:	The COMDAC is placed into the MRA recovery to be restored.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiuconfig[].
<i>Bad Data</i> =	Contents of the AP side select bits from SMaiuconfig[].
<i>Good Data</i> =	Correct value of the AP side select bits.
<i>Logical Index</i> =	AIU unit number in the upper 16 bits and AP number in the lower 16 bits.
<i>Dump 1</i> =	SMaiuconfig[] array entry.
<i>Dump 2</i> =	Contents of the AP side select bits from the SG (service group) 0 COMDAC.
<i>Dump 3</i> =	Contents of the AP side select bits from the SG 1 COMDAC.

Table 15 Error Code: SMEST

Error Code:	SMEST
Error Description:	Access of the rSMEST tuple for either the SG 0 COMDAC, or the SG 1 COMDAC associated with the AIU under audit has failed.
Possible Error Effect:	The UNITCFG audit will skip auditing the SMaiuconfig[] array associated with the missing circuit(s).
Corrective Action Taken:	Related audits are scheduled to correct error.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of the rSMEST tuple for SG 0 COMDAC.
<i>Bad Data</i> =	Address of the rSMEST tuple for SG 1 COMDAC.
<i>Good Data</i> =	Circuit name of the SG 0 COMDAC.
<i>Logical Index</i> =	Circuit name of the SG 1 COMDAC.
<i>Dump</i> =	rPICB tuple.

Table 16 Error Code: STUCK

Error Code:	STUCK
Error Description:	The COMDAC block job bit (cd_blk_jobs) or the digital loop back inhibit bit

	(dlb_inh) in the SMaiufig[] array is stuck set (DBYES). Additionally this error is used for the COMDAC blocking reply bit (cd_blk_replies) being set incorrectly based upon the state of the COMDAC. The bit should be set whenever the COMDAC is out of service or in the blocked condition otherwise the bit should be cleared. The <i>Bad Data</i> field contains information on which of these bits were found to be corrupt.
Possible Error Effect:	If the COMDAC block job bit is in error, then the error effect is that all COMDAC hardware jobs are inhibited from being sent from the SMP to the AIU, blocking all call processing and maintenance activities associated with that COMDAC. If the digital loop back inhibit bit is in error, then the error effect is that all digital loop back failures are inhibited from being reported. If the COMDAC block reply bit is in error, then call processing and maintenance activities will be effected. These activities will wait for replies from the COMDAC that that will not come because the COMDAC is out of service.
Corrective Action Taken:	The COMDAC block job (cd_blk_jobs) and digital loop back inhibit (dlb_inh) bits in the SMaiufig[] array are cleared. For errors involving the block reply bit (cd_blk_replies) in the SMaiufig[] array, set the block reply bit if the COMDAC (common data and control circuit) is out of service or blocked. Otherwise the block reply bit is cleared if the COMDAC is active and not blocked.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	Valid values are: 0 = The COMDAC block job bit is corrupt. 1 = The digital loop back inhibit bit is corrupt. 2 = The block reply bit is corrupt.
<i>Good Data</i> =	If <i>Bad Data</i> is 0, then this is the COMDAC SG number. Otherwise, <i>Good Data</i> is NA.
<i>Logical Index</i> =	AIU unit number.
<i>Dump 1</i> =	SMaiufig[] array entry found to be in error.
<i>Dump 2</i> =	For errors involving the block reply bit the rSMEST tuple of the SG 0 COMDAC.
<i>Dump 3</i> =	For errors involving the block reply bit and for duplex COMDAC configurations the rSMEST tuple of the SG 1 COMDAC.

Table 17 Error Code: STUCK2

Error Code:	STUCK2
Error Description:	The AP block job bit (ap_blk_jobs) in the SMaiufig[] array is stuck set for one of the AP circuits.
Possible Error Effect:	All AP hardware jobs are inhibited from being sent from the SMP to the AIU, blocking all call processing and maintenance activities associated with that AP circuit.
Corrective Action Taken:	The AP block job bit (ap_blk_jobs) in the SMaiufig[] array is cleared for the AP circuit allowing AP hardware jobs to be sent from the SMP to the AIU.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	AP number.
<i>Logical Index</i> =	AIU unit number.
<i>Dump 1</i> =	SMaiufig[] array entry found to be in error.
<i>Dump 2</i> =	rSMEST tuple associated with the AP circuit.

Table 18 Error Code: STUCKBUSY

Error Code:	STUCKBUSY

Error Description:	The false ring trip inhibit bit (frt_inh) in the SMaiufig[] array is stuck set for one of the ring generator (RG) circuits.
Possible Error Effect:	All false ring trip reports are inhibit causing faults in the ring generators to not be reported.
Corrective Action Taken:	The false ring trip inhibit bit (frt_inh) in the SMaiufig[] array is cleared for the RG circuit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Address of SMaiufig[].
<i>Bad Data</i> =	RG number.
<i>Good Data</i> =	AP number.
<i>Logical Index</i> =	AIU number.
<i>Dump 1</i> =	SMaiufig[] array entry found to be in error.
<i>Dump 2</i> =	riCDBCOM tuple associated with the RG circuit.

UTD

Software Release: 5E10 and later

1. UNIVERSAL TONE DECODER (UTD) AUDIT

The universal tone decoder circuit, located on the local digital service unit (LDSU) or model 2 digital services unit (DSU2), performs tone decoding functions required for call processing. The circuit is capable of decoding tones such as audible ringing, busy tone, dial tone, call-waiting tone, and receiver off-hook tone.

The universal tone decoder audit (UTD) verifies that every nonvacant tuple in the universal tone decoder relation (RLUTD) has a corresponding tuple in the static circuit data relation (RLCKTDATA). This audit identifies inconsistencies such as extra tuples, lost rCDBCOM tuples, and tuples with bad keys.

This audit is part of the Peripheral Control audits subgroup. It runs in the OSDS-M environment in the initialization and segmented modes.

2. DATA STRUCTURES

The UTD audit uses the information in the static circuit data relation (RLCKTDATA) to verify the dynamic universal tone decoder relation (RLUTD). The name field in the RLUTD relation is compared with the name field in the RLCKTDATA relation. Figure 1 shows the field in rLUTD checked by the audit.

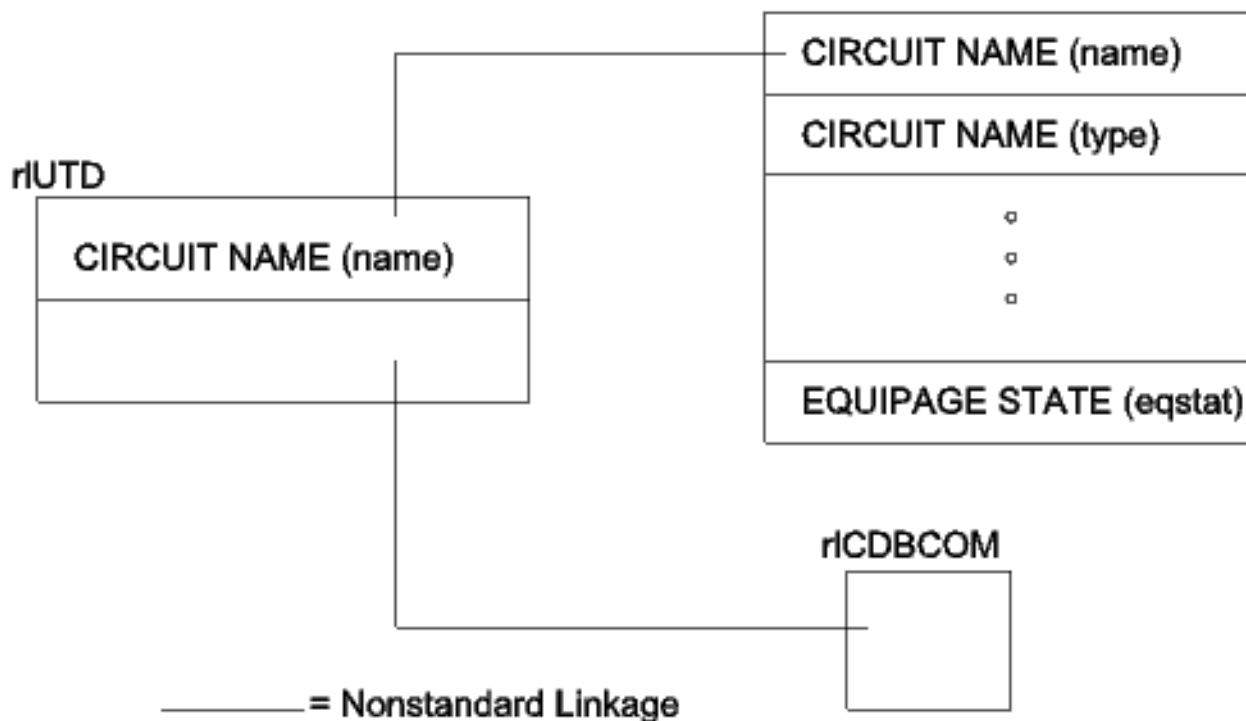


Figure 1 (UTD-1) Universal Tone Decoder Relation

3. AUDIT CHECKS

The universal tone decoder audit uses the dynamic driver (DRDYN) to retrieve tuples sequentially. The DRDYN driver calls a semantic check function to perform the following checks:

- The audit calls the existence check function to translate the physical key of the rLUTD tuple into a circuit name. For each nonvacant (or nonzero) rLUTD tuple retrieved, the audit verifies that a corresponding tuple

with the same circuit name exists in the static RLCKTDATA relation.

- (b) If the corresponding rICKTDATA tuple is in the unequipped (UNEQ) or growth (GROW) state and is the correct type of circuit, the audit verifies that its associated rCDBCOM tuple exists.
- (c) This audit also verifies the idle UTD count to be the same as the global variable used by PC and MIA.

4. ERROR HANDLING

Any nonvacant tuple in RLUTD that does not correspond to a tuple in RLCKTDATA is deleted and an error is reported. If a valid rLUTD tuple does not have an associated rCDBCOM tuple, an error is reported.

See Driver Error Codes (DRERRCDS), located in the Audit Drivers section of this manual, for any error codes not found below.

The audit produces the following error code(s).

Table 1 Error Code: COUNT_CDB

Error Code:	COUNT_CDB
Error Description:	The DB primitive can not traverse the idle list. The DB internal error or the set relation is wrong.
Possible Error Effect:	The idle UTD will not be available for call processing.
Corrective Action Taken:	The CDBCOM audit will be scheduled to recover the list.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	NA.
<i>Bad Data =</i>	The error return code.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Dump =</i>	NA.

Table 2 Error Code: EXTRATUP

Error Code:	EXTRATUP
Error Description:	The rLUTD tuple is not vacant as it should be, since the rICKTDATA tuple does not exist.
Possible Error Effect:	The rLUTD tuple may be accidentally taken to be valid, causing problems for other operational and maintenance software.
Corrective Action Taken:	Zero the rLUTD tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of the rLUTD tuple.
<i>Bad Data =</i>	NA.
<i>Good Data =</i>	NA.
<i>Logical Index =</i>	NA.
<i>Logical Index =</i>	Key of rLUTD tuple.
<i>Dump =</i>	rLUTD tuple.

Table 3 Error Code: KEY

Error Code:	KEY
Error Description:	The circuit name in the rLUTD tuple is not equal to the calculated name.
Possible Error Effect:	This could lead to accessing the wrong rICKTDATA tuple and cause operational and maintenance software to find errors.
Corrective Action Taken:	Correct the circuit name in the rLUTD tuple.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	Address of rLUTD tuple.
<i>Bad Data =</i>	Circuit name in the rLUTD tuple.
<i>Good Data =</i>	Address of rLUTD tuple.
<i>Logical Index =</i>	Key of rLUTD tuple.

<i>Dump</i> =	rlUTD tuple.

Table 4 Error Code: LOSTCDB

Error Code:	LOSTCDB
Error Description:	The rICDBCOM tuple with the same circuit name from rICKTDATA could not be found.
Possible Error Effect:	A valid rlUTD tuple has no busy/idle/OOS list information.
Corrective Action Taken:	None. The CDBCOM audit is scheduled as a related audit.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	Circuit name of rlUTD tuple.
<i>Dump</i> =	rlUTD tuple.

Table 5 Error Code: UTDCOUNT

Error Code:	UTDCOUNT
Error Description:	The actual idle UTDs in the system are different from that defined in the global variable.
Possible Error Effect:	The PC and FC will misinterpret the available UTDs in the system and the overload control probably will not throttle UTD count.
Corrective Action Taken:	Reset the global variable OCrcvcnt to reflect the actual UTDs.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	NA.
<i>Bad Data</i> =	OCrcvcnt.
<i>Good Data</i> =	Actual idle UTDs.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

VCCB-A

Software Release: 5E10 - 5E11

1. VOCODER CONTROL BLOCK (VCCB) AUDIT

The vocoder connection control block is located in a PHV for CDMA calls.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: ACT_LNK

Error Code:	ACT_LNK
Error Description:	The audit verifies the active links counter in the VCCB by counting the number of members of the xfer_voice array that are set to DBYES. This error indicates that the counter was incorrect.
Possible Error Effect:	
Corrective Action Taken:	The active links counter, act_links, is set to match the link count that the audit found.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	vccb_ptr→act_links
<i>Bad Data =</i>	link_map
<i>Good Data =</i>	link_count, audits counter
<i>Logical Index =</i>	vccb_idx, vccb index
<i>Dump 1 =</i>	Contents of the forward signaling structure, fsigg
<i>Dump 2 =</i>	Contents of the CDMA frame descriptor structure, cfd

Table 2 Error Code: PTR1

Error Code:	PTR1
Error Description:	The audit verifies the integrity of the forward and reverse signaling queue head and tail pointers, and the best voice pointer. The error is due to one of the following: - the forward signaling head or tail pointers is out of range, - the reverse signaling head or tail pointers is out of range, - the best voice pointer is out of range or is NULL.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will restore all of the pointers to a known valid state.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	vccb_ptr→fsigg.head_ptr; forward head pointer
<i>Bad Data</i> =	vccb_ptr→fsigg.tail_ptr; forward tail pointer
<i>Good Data</i> =	vccb_ptr→best.cfd_ptr; best voice pointer
<i>Logical Index</i> =	vccb index
<i>Dump 1</i> =	Contents of the forward signaling structure, fsigg.
<i>Dump 2</i> =	Contents of the CDMA frame descriptor structure, cfd.

Table 3 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit will take this error as the result of a write protection violation, an out of range read or invalid operation by foreground or any audit system process.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit will take appropriate recovery action depending on where it was when the error occurred. The assumption is made that the data structure was left in an inconsistent state when the VCCB audit was interrupted.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	0.
<i>Bad Data</i> =	NA.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	AUcount, the current element number.
<i>Dump</i> =	None.

VCCB-B

Software Release: 5E12 and later

1. VOCODER CONTROL BLOCK (VCCB) AUDIT

The vocoder connection control block is located in a PHV for CDMA calls.

2. DATA STRUCTURES

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here. The audit produces the following error code(s).

Table 1 Error Code: BADLINK

Error Code:	BADLINK
Error Description:	The VCCB contains the index of an IFD which is not owned by the VCCB (IFD.client != PSICCESH2 or IFD.clientidx != the VCCB layer 2 index).
Possible Error Effect:	Frames may be dropped or their contents corrupted.
Corrective Action Taken:	The audit clears (sets = PSIFDNULL) all IFD linkage fields associated with this list in the VCCB layer 2 structure being checked (VCCB.l2cb).
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data =</i>	The top 16 bits contain PSICCESH2. The bottom 16 bits contain the VCCB layer 2 index.
<i>Good Data =</i>	The number of IFD lists owned by the structure under audit.
<i>Logical Index =</i>	The top 16 bits contain PSICCESH2. The bottom 16 bits contain the VCCB index.
<i>Dump =</i>	IFD linkage information for this list.

Table 2 Error Code: BADTYPE

Error Code:	BADTYPE
Error Description:	The audit was unable to retrieve IFD linkage information for the IFD list being audited.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None (assert). This error indicates a problem which should be escalated to the next level of support. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	The top 16 bits contain the number of IFD lists owned by this structure. The lower 16 bits contain an index indicating which list is currently being audited.
<i>Bad Data =</i>	A value used internally for debugging.
<i>Good Data =</i>	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Logical Index =</i>	The top 16 bits contain PSICCESH2. The bottom 16 bits contain the VCCB layer 2 index.

<i>Dump</i> =	NA.

Table 3 Error Code: HEAD

Error Code:	HEAD
Error Description:	An IFD list owned by this VCCB is invalid. Either the ready or blocked indexes are not null (PSIFDNULL) and not valid ($\geq$ PSRIFDS).
Possible Error Effect:	IFDs may be owned when they should not be, possibly contributing to resource overloads which may in turn cause frames to be dropped.
Corrective Action Taken:	The audit clears (sets = PSIFDNULL) all IFD linkage fields associated with this list in the VCCB layer 2 structure.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICCESH2. The bottom 16 bits contain the VCCB layer 2 index.
<i>Dump</i> =	IFD linkage information for this list.

Table 4 Error Code: NONNULL

Error Code:	NONNULL
Error Description:	An IFD list owned by this VCCB is invalid. Although the "first" and "last" IFD indexes are null, either the ready or blocked indexes are not null (PSIFDNULL).
Possible Error Effect:	IFDs may be owned when they should not be, possibly contributing to resource overloads which may in turn cause frames to be dropped.
Corrective Action Taken:	The audit clears (sets = PSIFDNULL) all IFD linkage fields associated with this list in the VCCB.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An index indicating which IFD list owned by the structure under audit was in error.
<i>Bad Data</i> =	The top 16 bits contain the IFD list owner type. The bottom 16 bits contain the IFD list owner index.
<i>Good Data</i> =	The number of IFD lists owned by the structure under audit.
<i>Logical Index</i> =	The top 16 bits contain PSICCESH2. The bottom 16 bits contain the VCCB layer 2 index.
<i>Dump</i> =	IFD linkage information for this list.

Table 5 Error Code: READ

Error Code:	READ
Error Description:	The audit was unable to retrieve IFD list information for the VCCB.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	None (assert). This error indicates a problem that should be escalated to the next level of support. Refer to the TECHNICAL ASSISTANCE portion of the INTRODUCTION section of the Audits manual.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The number of IFD lists owned by the VCCB.
<i>Bad Data</i> =	A value used internally for debugging.
<i>Good Data</i> =	The maximum number of IFD lists which can be owned.
<i>Logical Index</i> =	The top 16 bits contain PSICCESH2. The lower 16 bits contain the VCCB layer 2 index.
<i>Dump</i> =	NA.

Table 6 Error Code: TIMERLNK

Error Code:	TIMERLNK
Error Description:	A timer index in a VCCB is out of range or indexes a timer which is not owned by

	the VCCB or is of an unexpected type.
Possible Error Effect:	Varies according to conditions.
Corrective Action Taken:	The audit sets the timer index field in the VCCB to the null value (PSNULLTE) or takes other special recovery based on the timer type.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	An internal value used to indicate which timer index in the VCCB was being checked.
<i>Bad Data</i> =	An internal value used to indicate the type of error.
<i>Good Data</i> =	An internal value indicating which entry in an array of timers (as indicated by the <i>Error Address</i> value) was being checked.
<i>Logical Index</i> =	The timer element index.
<i>Dump</i> =	The VCCB substructure which contained the timer index.

VTTMRS

Software Release: 5E13 and later

1. VERIFICATION TEST TIMER TABLES (VTTMRS) AUDIT

The VT_TMRS audit is for the SS7 operations, maintenance and administration part (OMAP) feature. The *administrable* MTP routing data and SCCP routing data are stored in the CCMRV_TEST[] and CCSRV_TEST[] tables respectively. Data in these tables is audited by the VTTMRS audit which runs in all SMs.

2. DATA STRUCTURES

CCmrsv_test[] = MTP routing verification test entry.

CCsrv_test[] = SCCP routing verification test entry.

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

The audit produces the following error code(s).

Table 1 Error Code: BAD_COUNTS

Error Code:	BAD_COUNTS
Error Description:	The error is one of the following: - The value contained in the global counter CCommloc_cntr does not equal the count of local <i>test_busy</i> fields set to DBYES in CCmrsv_test[]. This occurs if the error address is 1 and the logical key is 1. - The value contained in the global counter CComsloc_cntr does not equal the count of local <i>test_busy</i> fields set to DBYES in CCSrv_test[]. This occurs if the error address is 2 and the logical key is 1. - The value contained in the global counter CCommint_cntr does not equal the count of intermediate <i>test_busy</i> fields set to DBYES in CCmrsv_test[]. This occurs if the error address is 1 and the logical key is 2. - The value contained in the global counter CComsint_cntr does not equal the count of intermediate <i>test_busy</i> fields set to DBYES in CCSrv_test[]. This occurs if the error address is 2 and the logical key is 2.
Possible Error Effect:	The possible error effect is one of the following: CCommloc_cntr controls the number of local message transfer part routing verification tests (MRVTs) that can be scheduled. An incorrect value in CCommloc_cntr would result in an incorrect number of local

	MRVTs that can be scheduled. - CComsloc_cntr controls the number of local signaling connection control part routing verification tests (SRVTs) that can be scheduled. An incorrect value in CComsloc_cntr would result in an incorrect number of local SRVTs that can be scheduled. - CCommint_cntr controls the number of intermediate MRVTs that can be scheduled. An incorrect value in CCommint_cntr would result in an incorrect number of intermediate MRVTs that can be scheduled. - CComsint_cntr controls the number of intermediate SRVTs that can be scheduled. An incorrect value in CComsint_cntr would result in an incorrect number of intermediate SRVTs that can be scheduled.
Corrective Action Taken:	The corrective action is one of the following: - The audit sets CCommloc_cntr equal to the count of local <i>test_busy</i> fields set to DBYES in CCmrsv_test[]. - The audit sets CComsloc_cntr equal to the count of local <i>test_busy</i> fields set to DBYES in CCsrsv_test[]. - The audit sets CCommint_cntr equal to the count of intermediate <i>test_busy</i> fields set to DBYES in CCmrsv_test[]. - The audit sets CComsint_cntr equal to the count of intermediate <i>test_busy</i> fields set to DBYES in CCsrsv_test[].
Manual Action Required:	None.
Dump Description:	
<i>Error Address =</i>	1 if test array is CCmrsv_test[] OR 2 if test array is CCsrsv_test[].
<i>Bad Data =</i>	It could be one of the following: - CCommloc_cntr, if error address is 1 and logical key is 1. - CComsloc_cntr, if error address is 2 and logical key is 1. - CCommint_cntr, if error address is 1 and logical key is 2. - CComsint_cntr, if error address is 2 and logical key is 2.
<i>Good Data =</i>	Correct value to which the global counter should be set.
<i>Logical Index =</i>	It is one of the following: 1 = The global counter containing the count of local <i>test_busy</i> fields set to DBYES in the test array is incorrect. 2 = The global counter containing the count of intermediate <i>test_busy</i> fields set to DBYES in the test array is incorrect.
<i>Dump =</i>	None.

Table 2 Error Code: BIT1MAP

Error Code:	BIT1MAP
Error Description:	The error is one of the following: In the current entry of CCmrsv_test[], <i>test_busy</i> is set to DBYES, and

	<i>parent</i> is within range (between 0 and CCTTBLNUM), but the corresponding bit in CCtmrtbl[<i>parent</i>].mrvt_bitmap is not set. This occurs if the error address is 1. In the current entry of CCsrv_test[], <i>test_busy</i> is set to DBYES, and <i>parent</i> is within range (between 0 and CCTTBLNUM), but the corresponding bit in CCtmrtbl[<i>parent</i>].srvt_bitmap is not set. This occurs if the error address is 2.
Possible Error Effect:	The possible error effect is one of the following: - This error indicates that an MRVT is running without a corresponding timer, or that a previous MRVT has not terminated normally. In either case, the currently running test might never complete. This affects the total number of MRVTs that can be scheduled. - This error indicates that an SRVT is running without a corresponding timer, or that a previous SRVT has not terminated normally. In either case, the currently running test might never complete. This affects the total number of SRVTs that can be scheduled.
Corrective Action Taken:	The corrective action is one of the following: - The audit sets the current entry of CCmrv_test[] to an idle state. Specifically, it sets all fields to 0 and the <i>parent</i> field to its idle state of CCWRVAL. - The audit sets the current entry of CCsrv_test[] to an idle state. Specifically, it sets all fields to 0 and the <i>parent</i> field to its idle state of CCWRVAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1 if test array is CCmrv_test[] OR 2 if test array is CCsrv_test[].
<i>Bad Data</i> =	<i>parent</i> field in the current entry of test array.
<i>Good Data</i> =	<i>test_busy</i> field in the current entry of test array.
<i>Logical Index</i> =	The current index into the test array.
<i>Dump</i> =	Contents of the current entry of the test array.

Table 3 Error Code: PARENT

Error Code:	PARENT
Error Description:	One of the following conditions has occurred: - In the current entry of CCmrv_test[], <i>test_busy</i> is set to DBYES, but <i>parent</i> is not within range; that is, it is not between 0 and CCTTBLNUM. This occurs if the error address is 1. - In the current entry of CCsrv_test[], <i>test_busy</i> is set to DBYES, but <i>parent</i> is not within range; that is, it is not between 0 and CCTTBLNUM. This occurs if the error address is 2.
Possible Error Effect:	The possible error effect is one of the following: This error indicates that an MRVT is running without a corresponding timer, or that a previous MRVT has not terminated normally. In either case, the currently running test might never complete. This affects the total number of MRVTs that can be scheduled.

	This error indicates that an SRVT is running without a corresponding timer, or that a previous SRVT has not terminated normally. In either case, the currently running test might never complete. This affects the total number of SRVTs that can be scheduled.
Corrective Action Taken:	The corrective action is one of the following: - The audit sets the current entry of CCmrv_test[] to an idle state. It sets all fields to 0 and the <i>parent</i> field to its idle state of CCWRVAL. - The audit sets the current entry of CCsrv_test[] to an idle state. It sets all fields to 0 and the <i>parent</i> field to its idle state of CCWRVAL.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	1 if test array is CCmrv_test[] OR 2 if test array is CCsrv_test[].
<i>Bad Data</i> =	<i>parent</i> field in the current entry of test array.
<i>Good Data</i> =	<i>test_busy</i> field in the current entry of test array.
<i>Logical Index</i> =	The current index into the test array.
<i>Dump</i> =	Contents of the current entry of the test array.

Table 4 Error Code: ZAP

Error Code:	ZAP
Error Description:	The audit takes this error as a result of a write protect violation, out of range read, or invalid operation by foreground or any audit system process.
Possible Error Effect:	Neither CCmrv_test[] nor CCsrv_test[] is verified for this iteration of the audit.
Corrective Action Taken:	None.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	Value of Glmrvtact which indicates if MRVT feature is active.
<i>Bad Data</i> =	Value of Glsrvtact which indicates if SRVT feature is active.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	NA.
<i>Dump</i> =	NA.

WSFCORRAL

Software Release: 5E10 and later

1. WIDE BAND SIMULATED FACILITY GROUP CALL INFORMATION STATUS GLOBAL ARRAY (WSFCORRAL) AUDIT

The fractional DS1 switching using PRI (NxDS0) feature gives customers the ability to establish calls with customers at rates between 128 Kbps and 1536 Kbps through the 5ESS[®] switch module fabric by switching multiple rates of 64 Kbps. This bandwidth may be represented as $N(2 \leq N \leq 24)$ members of an SFG, and are stored in the RTwsfcorral[] (Wide Band SFG CORRAL) global array. The WSFCORRAL audit runs in all SMs and verifies the consistency of data contained in the Wide Band SFG CORRAL array.

2. DATA STRUCTURES

When a wide band SFG call is set up, a list of wide band SFG CORRAL array entries is created and linked to an owning tuple in the RLSFGCOR (SFG CORRAL) relation. When a list is created, the following fields in the wide band SFG CORRAL array are used:

- *prev_idx*: Contains the index of the *previous* (or predecessor) entry in the list. If this is the first entry on the list, this field is marked WSFSTART.
- *next_idx*: Contains the *next* (or successor) entry on the list. If this is the first entry on the last entry on the list, this field is marked WSFEND.
- *cid*: Contains the call ID (key) of the owning SFG CORRAL (RLSFGCOR) relation tuple.
- *used*: Indicates if the wide band SFG CORRAL array entry is being used (DBYES), or is idle (DBNO).
- *mbr[WSFMAXMBR][WSFMAXMBR]*: Contains the members being used for the call for a given entry.
- *quar*: Indicates if a given wide band SFG CORRAL array entry has been quarantined for recovery (DBYES) by some previous recovery action of the SFGCOR audit.

The maximum number of entries for a given list is AUWBLISTSZ.

In the RLSFGCOR relation, for members associated with wide band SFG calls, an implied linkage exists between the rlsfgcor tuple used for wide band and the of RTwsfcorral[] array entries. For these linkages, *wsf_head* in RLSFGCOR is the index into the RTwsfcorral[] entry which is serving as the head of the wide band list. The *wsf_tail* field in RLSFGCOR is used as the index into the RTwsfcorral[] array entry which is serving as the tail of the wide band list. For wide band lists of one RTwsfcorral[] entry, rlsfgcor.wsf_tail is always marked WSFSAMEIDX, where rlsfgcor.cnum (key) == RTwsfcorral[].cid.

Figure 1 and Figure 2 demonstrate the data structure linkages for single and multiple list entry scenarios.

rISFGCOR

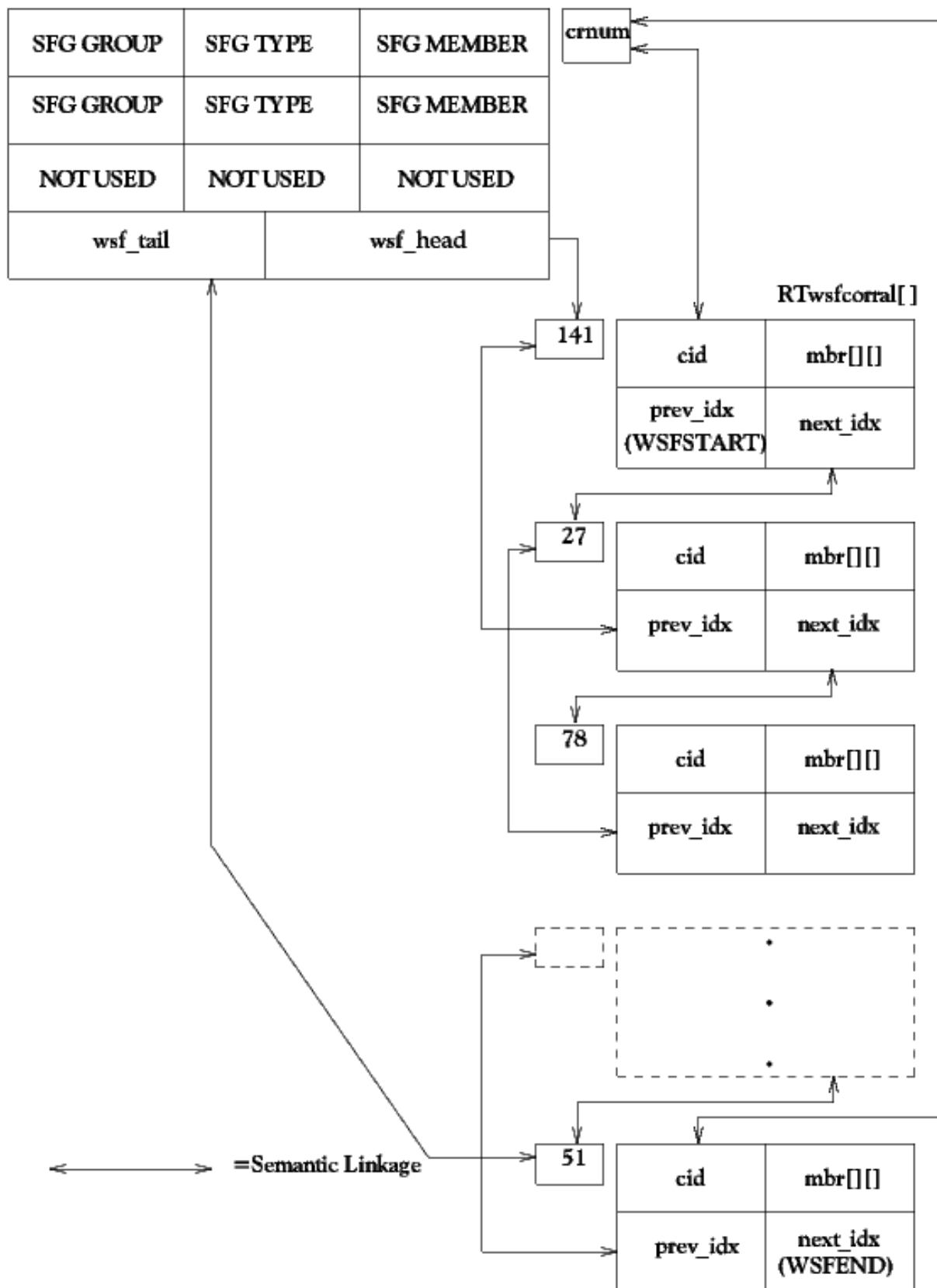
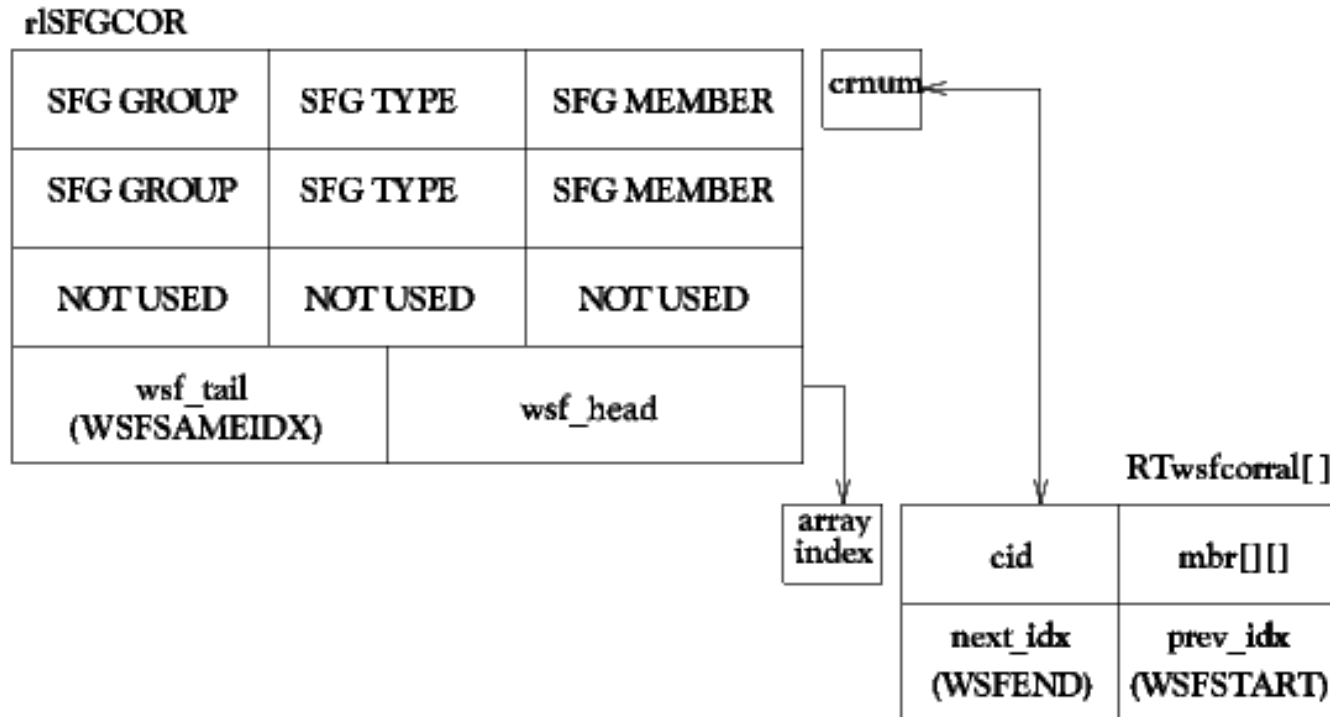


Figure 1 (WSFCORRAL-1) Wide Band SFG Scenario



↔ = Semantic Linkage

Figure 2 (WSFCORRAL-2) Example of list with single entry

3. AUDIT CHECKS

For each entry in the wide band SFG CORRAL array, the WSFCORRAL audit makes the following checks:

If the WSFCORRAL audit is entered:

- QUAR_INV = Check if entry has been quarantined for recovery.
- NOTZERO = Check if *used* bit is correctly marked for an idle entry.
- INVMEM = Check if *mbr[[]]* data is correctly marked for a busy head or middle list entry.
- MEMBAD = Check if *mbr[][]* data is correctly marked for a busy tail list entry.
- INV_STATE = Check if *mbr[[]]* data is in range.
- INVLIST
- AUWPRANGE = Check if *prev_idx* linkage is in range.
- AUWNRANGE = Check if *next_idx* linkage is in range.

OUTOFRANGE = Check if at least one linkage is in range.

CALLID_BAD = Check if the *cid* field in RTwsfcorral[] is in range.

INVLIST

AUWHEAD = Check if rISFGCOR tuple head linkage points to the head of the RTwsfcorral[] wide band list.

AUWNLINK = Check if the next entry on the wide band list points back to the one currently under audit.

AUWTAIL = Check if rISFGCOR tuple tail linkage points to the tail of the RTwsfcorral[] wide band list.

AUWPLINK = Check if the previous entry on the wide band list points back to the one currently under audit.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

The audit produces the following error code(s).

Table 1 Error Code: CALLID_BAD

Error Code:	CALLID_BAD
Error Description:	The call ID, <i>cid</i> , in the Wide Band SFG CORRAL array is not in range. The <i>cid</i> should be greater than or equal to CRZERO, and less than CRMAX.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The wide band SFG CORRAL array entry is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the wide band SFG CORRAL array entry under audit.
<i>Bad Data</i> =	The <i>cid</i> field in the wide band SFG CORRAL array entry.
<i>Good Data</i> =	CRMAX, which is the maximum call record number plus 1.
<i>Logical Index</i> =	The index of the wide band SFG CORRAL array entry under audit.
<i>Dump</i> =	The contents of the wide band SFG CORRAL array entry.

Table 2 Error Code: INVLIST

Error Code:	INVLIST
Error Description:	A list of wide band SFG CORRAL array entries linked off of the rISFGCOR tuple is invalid. For every rISFGCOR tuple marked for wide band (rISFGCOR.wsf_tail != 0) a list of one to AUWBLISTSZ RTwsfcorral[] wide band array entries should be linked off of the rISFGCOR tuple. This error indicates that a problem with the integrity of the list was encountered. For details on determining the specific nature of the problem encountered, please refer to the <i>Good Data</i> section of this audit error description.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The Wide Band SFG CORRAL array list which was found corrupt is recovered to an idle condition, and the SFGCOR audit is scheduled as a related audit to recover the associated SFG CORRAL (rISFGCOR) tuple.
Manual Action Required:	None.
Dump Description:	

<i>Error Address</i> =	The position in the list where the problem was found (0 to AUWBLISTSZ - 1).
<i>Bad Data</i> =	If the high order word dumped as <i>Good Data</i> is WSFSTART, the <i>prev_idx</i> field in the entry where the problem was found is dumped here. Otherwise, the <i>next_idx</i> field in the entry where the problem was found is dumped.
<i>Good Data</i> =	The direction of the recovery bitwise OR'd with the error type. The following directions apply: WSFSTART = Recovery is from point of corruption toward tail. WSFEND = Recovery is from point of corruption toward head. The following error types apply: AUWHEAD = The head entry of the wide band SFG CORRAL array (RTwsfcorral[]) list could not be accessed or was not found linked to the SFG CORRAL (rISFGCOR) tuple. For RTwsfcorral[rISFGCOR.wsf_head].cid, <i>cid</i> in RTwsfcorral[] must equal rISFGCOR.cnum. AUWNLINK = A wide band SFG CORRAL array entry did not point back to its predecessor (<i>next_idx</i> was invalid). AUWNRange = The <i>next_idx</i> of a wide band SFG CORRAL array entry which is not the tail of a list was found to be out of range. This field must be zero or greater, but less than WSFMAXCOR. AUWPLINK = A wide band SFG CORRAL array entry did not point back to its successor (<i>prev_idx</i> was invalid). AUWPRANGE = The <i>prev_idx</i> of a wide band SFG CORRAL array entry was found to be out of range. This field must be zero or greater, but less than WSFMAXCOR. AUWTAIL = The tail entry of a wide band SFG CORRAL array list was not found linked to the SFG CORRAL (rISFGCOR) tuple. For RTwsfcorral[rISFGCOR.wsf_head].cid, <i>cid</i> in RTwsfcorral[] must equal rISFGCOR.cnum for cases where the wide band list contains more than one entry.
<i>Logical Index</i> =	The index of the wide band SFG CORRAL array entry under audit.
<i>Dump</i> =	The contents of the wide band SFG CORRAL array entry.

Table 3 Error Code: INVMEM

Error Code:	INVMEM
Error Description:	The member data in the <i>mbr[][]</i> array is in range, but not populated correctly. Each set of members in the two-dimensional <i>mbr[][]</i> array is associated with an SF group. The first set will ALWAYS BE USED for populated entries. This was checked by verifying that both of the members in <i>mbr[0][X]</i> were not set to 0. If they were found both set to 0, this means that there are two member numbers of 0 for the same group. If <i>mbr[0][0]</i> is marked WSFNULMBR, this means no members are populated for a "used" entry (RTwsfcorral[].used = DBYES). For either case, the data in the entry is considered corrupt. Refer to the contents of the <i>mbr[][]</i> array, which is dumped as the second dump for this error message, for a display of the data found corrupted.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The wide band SFG CORRAL array entry is zeroed out.

Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the wide band SFG CORRAL array entry under audit.
<i>Bad Data</i> =	The member number contained in <i>mbr[0][0]</i> .
<i>Good Data</i> =	The member number contained in <i>mbr[0][1]</i> .
<i>Logical Index</i> =	The index of the wide band SFG CORRAL array entry under audit.
<i>Dump 1</i> =	The contents of the wide band SFG CORRAL array entry.
<i>Dump 2</i> =	The contents of the <i>mbr[][]</i> two-dimensional array where member data is stored.

Table 4 Error Code: INV_STATE

Error Code:	INV_STATE
Error Description:	The member data in the <i>mbr[][]</i> array is out of range. SFG members should be zero or greater, but less than RTLGMAXSFGMEM.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The wide band SFG CORRAL array entry is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the wide band SFG CORRAL array entry under audit.
<i>Bad Data</i> =	WSFNULLMBR, which is the value specified for a "NULL" SFG member.
<i>Good Data</i> =	RTLGMAXSFGMEM, which is the maximum allowable SFG member number + 1.
<i>Logical Index</i> =	The index of the wide band SFG CORRAL array entry under audit.
<i>Dump 1</i> =	The contents of the wide band SFG CORRAL array entry.
<i>Dump 2</i> =	The contents of the <i>mbr[][]</i> two-dimensional array where member data is stored.

Table 5 Error Code: MEMBAD

Error Code:	MEMBAD
Error Description:	The member data in the <i>mbr[][]</i> array is in range, but not populated correctly for a tail entry in a Wide Band SFG CORRAL list. If <i>mbr[0][1]</i> is marked NULL (WSFNULLMBR), then <i>mbr[1][1]</i> should also be marked WSFNULLMBR, and this entry should be marked as being the last entry in the list (RTwsfcorral[].wsf_tail = WSFEND).
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The wide band SFG CORRAL array entry is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The index of the wide band SFG CORRAL entry under audit.
<i>Bad Data</i> =	The member number contained in <i>mbr[0][1]</i> .
<i>Good Data</i> =	The member number contained in <i>mbr[1][1]</i> .
<i>Logical Index</i> =	The index of the next wide band SFG CORRAL array entry on the list.
<i>Dump 1</i> =	The contents of the wide band SFG CORRAL array entry.
<i>Dump 2</i> =	The contents of the <i>mbr[][]</i> two-dimensional array where member data is stored.

Table 6 Error Code: NOTZERO

Error Code:	NOTZERO
Error Description:	A wide band SFG CORRAL entry had its used bit marked DBNO, but is not an idle entry. If RTwsfcorral[].used is DBNO, then all other fields in the array entry should be zero.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The wide band SFG CORRAL array entry is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the wide band SFG CORRAL entry under audit.
<i>Bad Data</i> =	The contents of the used bit.
<i>Good Data</i> =	NA.

<i>Logical Index</i> =	The index of the next wide band SFG CORRAL array entry on the list.
<i>Dump</i> =	The contents of the wide band SFG CORRAL array entry.

Table 7 Error Code: QUAR_INV

Error Code:	QUAR_INV
Error Description:	A wide band SFG CORRAL entry had its quarantine bit set (quar = DBYES). The entry was quarantined as part of the recovery action by the SFGCOR audit which schedules the WSFCORRAL audit as a related audit. NOTE: This audit error will never appear with the message class AUDTFST. Refer to the SFGCOR audit message which precedes this error message for the specific error that was found with SFG data.
Possible Error Effect:	Possible incorrect measurements and customer billing of SFG traffic for the related group(s).
Corrective Action Taken:	The wide band SFG CORRAL array entry is zeroed out.
Manual Action Required:	None.
Dump Description:	
<i>Error Address</i> =	The address of the wide band SFG CORRAL entry under audit.
<i>Bad Data</i> =	The contents of the quar bit.
<i>Good Data</i> =	NA.
<i>Logical Index</i> =	The index of the wide band SFG CORRAL array entry.
<i>Dump</i> =	The contents of the wide band SFG CORRAL array entry.

XPTMR

Software Release: 5E11 and later

1. TRANSFER PROHIBIT TIMER (XPTMR) AUDIT

XPTMR audit steps throughs the RLXPTMR (transfer prohibit timer) relation. For each tuple, the audit makes sure that the linkage between the RLXPTMR and RLS7RDB (route data block) relations is correct.

2. DATA STRUCTURES

For details on the fields contained in the RLXPTMR relation, refer to the Translations Data Manual (235-600-1xx) and the Dynamic Data Manual (235-600-2xx).

For details on the specific data fields checked by this audit, refer to the error code explanations in the **ERROR HANDLING** section of this audit description.

3. AUDIT CHECKS

For details on the specific data fields checked by this audit, refer to the specific error code explanation in the **ERROR HANDLING** section of this audit description.

4. ERROR HANDLING

See driver error codes (DRERRCDS), located in the Audit Drivers section of this document, for additional error code(s) not shown here.

34. GLOSSARY

Glossary

Software Release: 5E10 and later

1. GLOSSARY

Access dictionary An array of access blocks, each of which contains access information for a given version of a relation.

Access methods package The data base software that furnishes data base access by mapping tuple keys to logical addresses.

Administrative module (AM) The central processing unit of the 5ESS[®] switch.

Administrative Services subsystem The subsystem that collects and processes information about switching activities.

Application integrity monitor (ASM) The subsystem environment in which the detection and correction of insanity in application software is overseen.

Area A number of contiguous items in a tuple.

Array A number of elements of the same type and size arranged in rows and columns.

Assert A statement in an application program that checks the ranges, redundancy, linkage, and/or consistency of program variables.

Assert message brevity The administrative module intervention feature that limits the number of audit output messages printed.

Attribute A column of a relation.

Audit Software that resides in long-lived and transient processes that are used to verify the validity of the UNIX[®] RTR Operating System data structures.

Audit assert An assert embedded in an audit program.

Audit Control program (AC) The audit program that schedules and controls audits; it resides in each processor and runs under the control of OSDS.

Audit control record Information that SIM needs to schedule and dispatch the audit, such as a unique name and member number for each audit. There are three audit record forms (**audrec**, **auderr**, and **audinst**).

Audit driver A common program that is called by specific audits to control segment breaks, perform syntactic checks, and handle errors. The 5ESS[®] Switch software has five drivers: the bit mask driver, the traverse driver, the linkage driver, the static driver, and the dynamic driver.

Audit error recovery form (auderr) RC/V form used to enter information about one of six error categories in an audit. It is identified by the key fields audit name, member number, and category number. One of these forms must be entered for each error category that applies.

Audit instance form (audinst) RC/V form used to enter information about one of four possible instances of an audit. It is identified by the key fields audit name, member number, and instance name (which may be NULL for an audit that will never have more than one instance). One of these forms must be entered for each instance of an audit. At least one **audinst** form must be entered for each **audrec** form that is entered.

Audit manager	The program unit that controls the scheduling and execution of all <i>UNIX</i> ® audits.
Audit record form (audrec) RC/V form	used to enter instance-independent information about an audit. It is identified by the key fields audit name and member number.
Audit requests	Requests from other processes to run audits.
Audit system	The software system that detects, isolates, and corrects errors in the 5ESS Switch data structures.
Automatic voice network (AUTOVON)	A four-wire telecommunications network used by various authorized government agencies that forms the basic framework of a world-wide telecommunications system.
Base priority input queue	A queue that holds messages (from an SM, for example); it is filled in by the Message Switch Kernel Process (MSKP).
Base priority output queue	A queue that holds messages (to the SMs, for example); it is filled in by system processes, such as the message broadcast system process.
Binary tree	A tree in which each node has one or two daughter nodes.
Bitmap	Records the status of each block in the file system, using one bit per block, excluding the blocks used for the super block and inode list.
Block	1) A tuple. 2) A contiguous area of memory.
Breakpoint	A programming entity that enables users to halt a process running in the switching module, execute a predefined set of utility functions, and resume execution of the process.
Brevity	The feature that limits the number of audit output messages logged and printed.
Buddies	An original process and its associated remoted process are referred to as buddies. When referring to one of the pair, the other is the buddy.
Business and Residence Custom Services (BRCS)	The custom calling features and centrex services included in generics 5E2(1) and later that can be tailored to the needs of each 5ESS Switch.
C	The high-level programming language in which most of the 5ESS Switch software is written.
C-language structure	A collection of variables, possibly of different types, grouped together under a single name for convenient handling.
Call processing	The activities involved in transmitting voice and data calls from inception, through the network, to their final destination. Three subsystems handle call processing: Feature Control, Routing and Terminal Allocation, and Peripheral Control.
Call record	A fixed-size block of memory associated with each call processed by the switching system.
Central processor (AM)	The centralized processor in the 5ESS Switch system that performs the processing, administration, and maintenance that cannot be economically distributed to switching modules; the 3B20D processor.
Channel	A peripheral time slot.
Circular list	A linked list in which the last node contains a pointer to the first node. A separate pointer is kept for the current node.
Clock diagnostic monitor	Clock daemon program that executes commands at specified dates and times according

to the instructions in file */usr/lib/crontab*.

Communications module (CM) The hardware interface between the switching modules and the administrative module consisting of the message switch (MSGS) and the time-multiplexed switch (TMS).

Compound key A set of attributes of a relation that determines one tuple of that relation (for example, the switching module number, digital service unit, and service group fields form the key of rIGCKTI).

Concurrent activities Two activities that have both begun but not yet terminated.

Contiguous files Contains data stored in single multiple extent files.

Coupler A special data structure used to extend owner-member linkage to many-many associations between tuples.

Customer-originated recent change (CORC) The feature of the Business and Residence Custom Services that allows business and residential customers to enter recent change transactions without operating company assistance.

Data Base Manager (DBM) The program that provides facilities for accessing and updating all global data.

Data delivery A mechanism that delivers status bits from the administrative module to the switching modules to inhibit and allow routine audits.

Data tuple One of the three types of tuples in a relation. Data tuples are the tuples requested and manipulated by call processes.

Defensive check A means to guard against run-time errors in the operating system or in application processes. Also a synonym for assert.

Diagnostics The programs that locate faults and verify repairs.

Dictionary A data structure that contains information for locating other data structures.

Domain The set of possible values for an attribute.

Doubly linked list A linked list in which each node contains pointers to both the predecessor and successor nodes.

Driver See "Audit driver."

Dummy tuple One of the three types of tuples in a relation. Dummy tuples are the first tuples in each relation; they normally contain all zeros and are used to detect sporadic system abnormalities such as wild writes.

Dynamic In the context of data in memory, dynamic means not write-protected. In the context of data structures, dynamic means the data structure uses linkage.

Dynamic driver An audit driver that steps through relations with tuples that have sequential keys.

Dynamic relation dictionary An array of description blocks, each of which defines the current (most recent) version of the associated relation.

Element A component of an array.

Elevated segmented mode The mode of operation in which audits run at priority 1 and take real-time breaks.

Equipment configuration data base Data base that contains all the attributes and current status of all processor and peripheral hardware known to the operating system. Software integrity records in the ECD contain

information concerning execution of audits and audit error recovery.

Error	A type of fault in which an audit encounters client data. The audit reports the error and continues.
Escalation	A recovery action taken when a critical resource is in error and when taking segment breaks before the error is corrected would further degrade processing.
Event	1) A system fault. Events are assigned event numbers, which are printed in the output messages. 2) A two-bit piece of information sent by a process to notify another process about an impending message. 3) Interprocess communication mechanism for user processes used to synchronize processes.
Extent	A range of contiguous blocks in a file system, specified by starting address and size.
Failure	A type of fault that occurs when the system wants to schedule an escalation or additional audits but can not.
Feature Control subsystem	The call processing subsystem responsible for sequencing call processing actions.
FIFO (first-in, first-out) list	A synonym for queue.
Full initialization	The form of initialization in which stable calls (those in a stable talking state) are disconnected.
Generic Access Program (GRASP)	A utility program residing in the AM that helps verify application programs and localize and resolve problems.
Generic Access Program for the Switching Module (GRASPSM)	A utility program residing in each switching module that helps verify application programs and localize and resolve problems.
Global data	The data in memory that is not allocated on a generic basis or at compile time to individual subsystems.
Global port key (gport)	The key that identifies the ports in rIPRTSTATUS. Each gport contains 24 bits of information: the 8-bit processor name and the 16-bit local or internal port name.
Global system status table (GSS)	A table that maintains status bits for each hardware unit in the system.
Head cell	One of the three types of tuples in a relation. Head cells are used as owners for idle resources.
Head table	A table containing pointers to pages of memory.
Heap	A data structure that stores data in consecutive memory locations, much like a stack. However, the heap pointers remain pointing to the first locations of the heap, not the next available stack locations.
Hunting	The process of selecting one of a set of available ports to satisfy a calling customer.
Idle list	1) A linked list of available nodes. 2) A list of idle resources linked to a head cell.
Incore data base	Data base read off of disk that is resident (incore) in the machine.
Index	A number that specifies an element in an array.
Indirect block	There are three types: a block that contains pointers to data blocks is a single indirect block ; a block that contains pointers to single indirect blocks is a double indirect block ; a block that contains pointers to double indirect blocks is a triple indirect block .

Initialization mode The mode of operation in which audits are run during system initializations.

Inode Defines a file in a file system. It contains the size, block addresses, and mode information about the file.

Insanity The condition of the system in which abnormal processing occurs.

Integrated SLC[®]-96 The call processing system consisting of remote terminals and hardware in the switching modules that provides low-cost service to remote customers.

Interject level See OSDS interject level.

Interprocessor data synchronization A means of synchronizing interprocessor redundant dynamic data by using interprocessor messages.

Item A word of memory within a tuple, a portion of a word of memory within a tuple, or a number of consecutive words of memory within a tuple.

Joining The process of combining relations with a common attribute; the new relation is a concatenation of the relations to be joined, with one of the resulting duplicate common attributes removed.

Kernel process A logical collection of software running at the level closest to the machine.

Key An attribute (or set of attributes) of a relation that determines one tuple of that relation (for example, the circuit name field is the key of rHLSC). If more than one attribute defines the key, it is a compound key (see "Compound key").

Killable process Executable process that can terminate by itself or be terminated without disrupting service. All user-level processes are considered killable.

LIFO (last-in, first-out) list A synonym for stack.

Linear list A synonym for linked list.

Link A connection between tuples implemented by placing within one tuple the address (or physical key) of another tuple.

Linkage area The link items in a tuple.

Linkage driver An audit driver that verifies the one-to-one and owner-member linkages among relations for certain audits.

Linked list A dynamic list in which nodes contain pointers to other nodes.

Link-sharing A type of linkage that is used in special cases such as custom calling features where one process owns many ports.

List pointer A type of physical linkage that indicates the first node of a list.

Logical key A number used to specify a tuple. It is equal to the physical key minus the number of head cells plus one.

Maintenance The activities involved in detecting internal errors and recovering faults. Four subsystems handle audit-related maintenance: Switch Maintenance, Terminal Maintenance, Utility, and Diagnostic Supervisor.

Maintenance request administrator (MRA) A program in each switching module that handles hardware maintenance

requests and updates system status tables.

Manner	An indication to audit control about where in a data structure an audit should run. Audit manners include starting (START), resuming after a segment break (RESUME), and terminating (ZAP).
Manual input commands	PDS and MML format input commands that allow the user to query and request system services.
Mapping	A synonym for linkage.
Master audit	An audit that runs in one processor and calls a slave audit to run in another processor.
Master control program	The part of the audit control program that understands partition boundaries and that updates slave data in the AM from the master data on the SM.
Master/slave linkage	The type of linkage that occurs when one tuple is a member in more than one owner-member linkage. One linkage is called the master linkage; the others are called slave linkages.
Member	In the context of owner-member linkage, a member is a tuple in a linked list owned by another tuple. In the context of a C-language structure, a member is a variable of a structure.
Member page	A page of memory containing tuples accessed via multilevel keys.
Message	1) A mechanism by which processes communicate with each other. 2) A mechanism that transfers data from one process to another.
Message broadcast mechanism	See "operating system message broadcast and reply collection mechanism".
Min-mode	A processing configuration that inhibits most audit activity but allows critical processing to continue.
Mode	The way an audit runs depending on its real-time requirements. Audit modes include segmented, nonsegmented (directed), initialization, and postinitialization.
Module integrity control (MICO)	The system integrity program that controls the audit, initialization, assert, and overload control functions in the switching modules.
Multiextent file	Contains contiguous data stored in up to six separate extents. Each extent is specified by its starting address and size.
Multilevel key	A key consisting of a high-order part used to look up the address of a member page in a head table, and a low-order part used to look up a tuple on the member page.
NCT links (network, control, and timing links)	High-speed data links that connect the time-multiplexed switch to the switching modules and to the message switch.
Network Management subsystem (NM)	The part of the Administrative Services subsystem that maintains traffic and routing in the switching system.
Node	A component of a linked list consisting of data and one or more pointers to other nodes.
Nonkillable process	Executable process that cannot be terminated without a disruption in service.
Nonprocess error	An error not associated with a specific process.
Nonreturning assert	An assert that does not return to program control and also purges the current process.
Nonsegmented audits	Audits that run to completion and report the amount of CPU time used. All UNIX [®] system

and supervisor level audits are nonsegmented.

Nonsegmented mode The mode of operation in which audits run to completion without real-time breaks.

Nonstandard linkage An association between data structures in which the data structures have the same value for (usually) the key. Unlike a real link (one-to-one or owner-member), neither relation in a nonstandard linkage contains the address of another relation in a linkage attribute.

One-dimensional array An array having only one row.

One-to-one linkage The linkage between two tuples in which each tuple has a single link to the other tuple.

Open relation dictionary An array of relation description blocks, one of which is assigned for each relation associated with a transaction.

Operating System for Distributed Switching (OSDS) The operating system that controls the functions of the switching system. The 5ESS Switch contains several instances of OSDS: one in the SM, one in the SMKP, and one in OKP.

Operating system message broadcast and reply collection mechanism The mechanism that allows user processes to send, receive, and react to SM messages. The broadcast mechanism works at the OSDS interject level. The message broadcast mechanism uses the base priority input queue for receiving messages from the SMs; it uses the message broadcast system process, which fills up the base priority output queue, to send messages.

Operational kernel process (OKP) An environment in the administrative module in which application-oriented call processing occurs.

OSDS interject level The processing level that handles short-duration jobs that are controlled by neither system processes nor terminal process. The level is made up of gaps in the call processing stream that occur at regular intervals. The interject intervals are controlled by OSDS by means of the base priority input queue. In generic 5E2(2), interject level processing is used only by the RTA system and the message broadcast mechanism.

Owner A tuple that is linked in an owner-member linkage to a list of member tuples.

Owner-member linkage A linkage between an owner tuple and a number of member tuples. The member tuples are arranged in a doubly linked list with each member tuple having an additional link to the owner tuple. The owner tuple has links to the first and last member tuples.

Packed array An array with no gaps between elements (that is, an array with no null elements).

Page descriptor table Contains administrative information about each page of physical memory. There is a one-to-one correspondence between a physical page and a PDE.

Page table List of physical pages in main memory that belong to a segment. Each member of the list is referred to as a PGE. The MMGR administers an array of PGTs and allocates a PGT to every segment that resides in main memory.

Partition mounting kernel process (PMKP) The kernel process that mounts new 5ESS Switch test and ODD partitions and creates the application integrity monitor (ASM) on full configuration AM boots.

Peripheral Control subsystem The call processing subsystem that acts upon customer requests collected by the Feature Control subsystem and insulates the Feature Control and Terminal Maintenance subsystems from changes in hardware configuration.

Peripheral Diagnostics Language (PDL) The language in which diagnostics are written.

Per-mapping information Information specifying where linkages are stored in each tuple of the audited relation.

Per-process list A list of timing control blocks used by a process needing more than one timer.

Per-relation information Generic information about a relation that includes the number and size of the tuples in the relation, the number of head cells, and the number and type of linkages found in each tuple.

Physical key The array index of a tuple; the address (link) of a tuple.

Physical linkage The type of linkage occurring between records both in network and in hierarchical data bases. Physical linkages use pointers within each record to lead to other records.

Plain old telephone service (POTS) Local call service between residential customers.

Plant measurements system data base Data base that stores system performance information. Counts of audit attempts and failures are stored in the PMS data base.

Pointer A variable or item that contains a value giving the location of some entity such as another node, item, or variable.

Point-to/point-back check A verification process that determines whether a relation pointed to from a source relation points back to the source relation.

Popping The operation of removing a node from a stack.

Postinitialization mode The mode in which audits restore noncritical data structures after system initialization.

Predecessor The node before a given node in a linked list.

Process A programming entity that implements an activity. Each process consists of a program, a process control block, and a stack.

Process control block The segment of a process that contains its controlling parameters and virtual memory administration structures.

Process errors The errors associated with a single identifiable process.

Process ID (PID) The unique identifier assigned to each process as it is created.

Projection The selection of specific attributes from a relation, followed by the deletion of all resulting duplicate tuples.

Pushing The operation of adding a node to a stack.

Quarantine field A field that lets client programs know whether a tuple may be audited. If the quarantine bit is set, the audit recovers the tuple without checking it. If an audit finds an invalid tuple in an associated relation, it sets the quarantine field so the audit that is checking the relation does not have to audit the tuple.

Quasikillable process Process that is monitored and automatically regenerated after being terminated.

Queue A linked list in which additions can only be made at one end and deletions can only be made at the other end. A synonym for FIFO list.

Recent change and verify Procedures used for creating or modifying records in the ECD.

Regular file	Collection of individual blocks which may be scattered throughout a file system. Each block is specified only by its address.
Related audits	The audits scheduled by an audit when it finds an error.
Relation	An array of tuples.
Relational data base	A type of data base that organizes data in relations so that users have a simple view of it.
Remote process	A process invoked by an existing process that resides in a processor other than the one in which the original process resides.
Routine audit intervention	The administrative module intervention feature that limits the number of routine audits run.
Routine segmented mode	The mode of operation in which audits run at priority 0 and take real-time breaks.
Routing and Terminal Allocation subsystem	The call processing subsystem that routes calls to available terminals.
Sanity	The normal processing state of the system.
Sanity timer	The system integrity measure that detects malfunctions by checking the time required to perform system processes.
Segment	1) An interval of time during which an audit runs. 2) The basic memory unit in <i>UNIX</i> [®] , composed of from 1 to 64 pages. Each segment is 512 32-bit words in length. 3) Logically related portion of main memory with consecutive virtual addresses.
Segment descriptor table	Contains the information that is used to administer segments of memory. Each entry in the table is referred to as an SDE. An SDE is allocated upon the creation of a segment and remains allocated until it is no longer referenced by any process. An SDE exists for every segment regardless of whether the segment is in main memory or swapped out.
Segmented audits	Audits that relinquish the CPU after a predetermined amount of time. Most kernel level audits execute in a segmented mode.
Selective incoming load control (SILC)	The feature of the Network Management subsystem that places controls on a switch's incoming traffic during periods of congestion.
Selective initialization	The form of initialization in which transient calls (those not in a stable talking state) are disconnected.
Semantic check	The verification an audit makes on semantic data.
Semantic data	The data items in a tuple.
Sequential list	A synonym for one-dimensional array.
Set-member linkage	The linkage within the member tuples of an owner-member linkage.
Set-owner linkage	The linkage within the owner tuples of an owner-member linkage.
Shared linkage	A type of linkage in which the linkage area of a tuple contains either one-to-one or owner-member linkage, depending on how the tuple is used.
Single process purge (SPP)	The form of initialization in which a single process (such as the routine audit process) is terminated.

- Singleton** A special case of owner-member linkage consisting of a linkage between only two tuples.
- Singly linked list** A linked list in which each node contains a pointer to the successor node.
- Slave audit** An audit in one processor that is called by a master audit in another processor.
- Slave data** The data on the AM to be updated to agree with the master data on the SM.
- Slave linkage** See "Master/slave linkage."
- Software demand paging package** The data base software that maps logical addresses to physical addresses and handles buffer management when data is stored in disk files.
- Software integrity subsystem** A system feature designed to handle all aspects of software reliability. The software integrity subsystem is composed of SIM, the ECD, audits, the system integrity output formatter, manual input commands, audit requests, and a PMS interface.
- Stack** A linked list in which addition and deletions can only be made at one end. A synonym for LIFO list.
- Static** In the context of data in memory, static means write-protected. In the context of data structures, static means the data structure does not use linkage.
- Static driver** The audit driver that reads successive tuples of static relations in the data base to verify the tuples of other relations.
- Structure** See "C-language structure."
- Subgroup, audit** The group of audits related to a subsystem.
- Subsystem** The 5ESS Switch software for a specific set of functions.
- Successor** The node following a given node in a linked list.
- Switching module (SM)** The generic name for remote switching modules (RSMs), host switching modules (HSMs), and local switching modules (LSMs).
- Switch Maintenance Kernel Process (SMKP)** The program that controls the execution of the audits related to the 5ESS Switch call processing that are not controlled by OSDS.
- Switch Maintenance subsystem** The maintenance subsystem that protects the switching system from problems in the hardware in the switching modules and peripheral units.
- Syntactic check** The verification an audit makes on syntactic data.
- Syntactic data** The logical key and linkage items in a tuple.
- System environments** OKP (Operation Kernel Process), SMKP (Switch Maintenance Kernel Process), SIM (system integrity monitor), AIM (application integrity monitor), OSDS-C (Operating System for Distributed Switching--Central Processor), and OSDS-M (Operating System for Distributed Switching--Switching Module).
- System integrity control (SICO)** The system integrity program in the administrative module that provides global integrity control for switching functions residing in OSDS-C and coordinated MICO integrity actions.
- System integrity monitor (SSM)** 1) The system environment in which the detection and correction of insanity in *UNIX*[®] is overseen. 2) Software integrity and audit control process. SIM is responsible for scheduling all audits.

- System integrity monitor control (simcntl) record Information stored in the ECD and used by SIM to regulate the percentage of system time to be spent running audits and the scheduling of manually requested audits. A master inhibit state in this record is used to turn off the scheduling of all routine audits.
- System integrity output formatter Program that provides a consistent output message format when reporting audit results to the user.
- System process A process that provides services on a system-wide basis. System processes are created and terminated independent of call processing and do not share their programs with other processes.
- Terminal maintenance processes The processes that periodically check hardware terminals, remove faulty terminals from service, and generate reports on output devices.
- Terminal maintenance subsystem The maintenance subsystem that protects the switching system from problems in trunks, lines, service circuits, and signaling paths.
- Terminal process A process that controls the associated physical terminal.
- Tight loop A software error that occurs when a program does not give up control within a specified time interval (usually 250-400 milliseconds).
- Tight loop detector The sanity timer in the application integrity monitor that purges processes that do not give up control in a specified time interval.
- Transaction The means used by the initialization and maintenance code to read port assignments, service status, etc., and to manually add recent change information to the data base.
- Transaction dictionary An array of transaction blocks, one of which is assigned for each active transaction.
- Traverse To scan a linked data structure by tracing along its pointers.
- Tree A set of linked nodes that contains a unique root node and does not contain loops. Each node may contain one or more child nodes.
- Trigger An event that causes the occurrence of another event.
- Tuple A row in a relation. In the 5ESS Switch data base, a tuple is a fixed-length Record that consists of a number of items.
- Tuple level access package The data base software that manages transactions, controls concurrent access, and handles data distribution.
- Two-dimensional array An array having more than one row. A two-dimensional array is implemented as an array of arrays.
- Union A type of structure that saves memory space by overwriting portions of it.
- UNIX[®] RTR operating system The main operating system in the 3B administrative module.
- Wild write Data mutilation that occurs when a program writes meaningless data into an incorrect data area. Wild writes are usually caused by program bugs.
- Word A discrete area of memory, usually four bytes long.

List of Figures

Figure 2-1 : 5ESS® Switch Software Organization

Figure 2-2 : 5ESS® Switch Relation Structure

Figure 2-3 : One-Dimensional Array

Figure 2-4 : Two-Dimensional Array

Figure 2-5 : Multiple Arrays Used to Store Table with Different Data Types

Figure 2-6 : Singly Linked List

Figure 2-7 : Linkage Reconstruction by Audits

Figure 2-8 : Types of Linked List

Figure 2-9 : Stack (LIFO List)

Figure 2-10 : Queue (FIFO List)

Figure 2-11 : Heap

Figure 2-12 : Structure and Union

Figure 2-13 : Message Layout

Figure 2-14 : Linking a Tuple from the Idle List

Figure 2-15 : Multilevel Key

Figure 2-16 : One-to-One Linkage

Figure 2-17 : Owner-Member Linkage

Figure 2-18 : Many-to-Many Association

Figure 2-19 : Use of Couplers

Figure 2-20 : Process Implementation

List of Tables

Table 2-1 : Example of Table Requiring Two-Dimensional Array

Table 2-2 : Example of Table Requiring Multiple Arrays

Table 3-1 : Summary of Audit Scheduling Mechanisms

Table 3-2 : Summary of Audit Scheduling Effects

Table 3-3 : Audit Input Messages Summary

Table 3-4 : SODD Audit Responses to Inputs