

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers
***UNIX*[®] RTR Operating System**

Recent Change and Verify Manual

254-303-104
Issue 1
May 1998

Copyright © 1998 Lucent Technologies. All Rights Reserved.

This Information Product (IP) is protected by the copyright laws of the United States and other countries. It may not be reproduced, distributed, or altered in any fashion by any entity, including other Lucent Technologies Business Units or Divisions, without express written consent. For permission to reproduce or distribute, call 1-888-LTINFO6 (1-888-584-6366).

Notice

Lucent Technologies is the successor to the business and assets of AT&T Network Systems Business Unit. Every effort was made to ensure that the information contained in this IP was complete and accurate at the time of printing. However, information is subject to change.

This IP describes certain hardware, software, features, and capabilities of Lucent Technologies products. This IP is for information purposes, and you are cautioned that the hardware, software, features, and capabilities described herein may not be the same as the product you have.

Trademarks

5ESS is a registered trademark of Lucent Technologies.

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

All other trademarks are identified as they appear in the text.

Ordering Information

This IP is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana. Orders for additional copies may be placed from the continental United States by calling 1-888-LUCENT8 (1-888-582-3688) or faxing to 1-317-322-6484. Customers outside the continental United States may place orders by contacting the in-country Lucent Technologies market manager. Lucent Technologies personnel outside the continental United States can order additional copies by faxing to 1-317-322-6699.

Information Product Support Telephone Number

Call 1-888-LTINFO6 (1-888-584-6366) to report errors or to ask questions about this IP.

Acknowledgment

Developed by Lucent Technologies Network Systems Customer Training and Information Products.

How Are We Doing?

Document Title: AT&T 3B20D and 3B21D Computers *UNIX*® RTR Operating System
Recent Change and Verify Manual

Document No.: 254-303-104

Issue 1, Revised: Date: June 1999

Lucent Technologies welcomes your feedback on this document. Your comments can be of great value in helping us improve our documentation.

1. Please rate the effectiveness of this document in the following areas:

	Excellent	Good	Fair	Poor	Not Applicable
Ease of Use					////////////////////
Clarity					////////////////////
Completeness					////////////////////
Accuracy					////////////////////
Organization					////////////////////
Appearance					////////////////////
Examples					
Illustrations					
Overall Satisfaction					////////////////////

2. Please check the ways you feel we could improve this document:

- | | |
|--|---|
| <input type="checkbox"/> Improve the overview/introduction | <input type="checkbox"/> Make it more concise/brief |
| <input type="checkbox"/> Improve the table of contents | <input type="checkbox"/> Add more step-by-step procedures/tutorials |
| <input type="checkbox"/> Improve the organization | <input type="checkbox"/> Add more troubleshooting information |
| <input type="checkbox"/> Include more figures | <input type="checkbox"/> Make it less technical |
| <input type="checkbox"/> Add more examples | <input type="checkbox"/> Add more/better quick reference aids |
| <input type="checkbox"/> Add more detail | <input type="checkbox"/> Improve the index |

Please provide details for the suggested improvement. _____

3. What did you like most about this document?

4. Feel free to write any comments below or on an attached sheet.

If we may contact you concerning your comments, please complete the following:

Name: _____ Telephone Number: _____

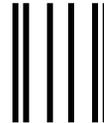
Company/Organization: _____ Date: _____

Address: _____

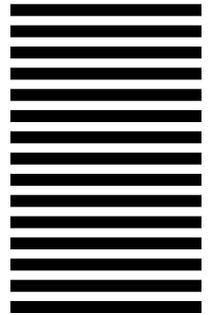
When you have completed this form, please fold, tape, and return to address on back or Fax to: 910-727-3043.

-----Do Not Cut—Fold Here And Tape-----

Lucent Technologies
Bell Labs Innovations



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1999 GREENSBORO, N.C.

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENTATION SERVICES
2400 Reynolda Road
Winston-Salem, NC 27199-2029



Contents

About This Information Product	<u>xi</u>
■ Purpose	<u>xi</u>
■ Audience	<u>xi</u>
■ Reason for Issue	<u>xii</u>
■ How to Use This Information Product	<u>xii</u>
■ Conventions Used	<u>xiii</u>
■ Safety Labels	<u>xv</u>
■ Related Information Products	<u>xvi</u>
■ How to Comment on This Information Product	<u>xvii</u>
■ Mandatory Customer Information	<u>xvii</u>

1	Database Structure	<u>1-1</u>
	■ Introduction	<u>1-1</u>
	■ Low-Level Access Databases — File Conventions	<u>1-2</u>
	■ ECD Structure	<u>1-2</u>
	■ SG Database Structure	<u>1-10</u>
	■ Database Location Conventions	<u>1-11</u>
	■ Database Life Cycle	<u>1-13</u>

2	Logical-to-Physical Mapping	<u>2-1</u>
	■ Introduction	<u>2-1</u>
	■ Special Device Files	<u>2-1</u>
	■ Driver View	<u>2-2</u>

3	Representing Devices	<u>3-1</u>
	■ Introduction	<u>3-1</u>
	■ Control Unit Complex	<u>3-4</u>

Contents

- Disk File Controller Complex [3-9](#)
- Input/Output Processor Complex [3-12](#)

4	Database Tools	4-1
	■ Introduction	4-1
	■ Conventions	4-1
	■ CREATEECD and CREATESG Commands	4-2
	■ RCVECD and RCVSG Commands	4-3
	■ LOADF3B Command	4-28

5	Using Database Tools	5-1
	■ Creating a Database	5-1
	■ Verifying a Database	5-1
	■ Modifying Recent Change Database Items (trbegin and trend)	5-1
	■ Aborting a Transaction (trabort)	5-2
	■ Listing Keys (klist)	5-3
	■ Printing Forms (prtlist)	5-3
	■ Reclaiming Fragmented Database Space (dbdump and dbload)	5-3
	■ Finding All UCBs Under a Specific UCB	5-4
	■ Listing UCBs Under a Specific IOP	5-4
	■ Finding All Links to a Specific Form	5-5
	■ Finding the RID of a Specified Form	5-5
	■ Applying a Batch Script	5-5
	■ Making Incore Changes Permanent (activate)	5-5
	■ Placing an ECD in Flat-File Format	5-5
	■ Moving Databases	5-6
	■ Performing Routine Maintenance	5-6
	■ Color RC/V	5-6
	■ Example of an RC/V Session	5-8

Contents

6	Database Form Guide	6-1
	■ General Form Information	6-1
	■ ECD Forms	6-5
	■ SG Forms	6-333
	■ ECD/SG Forms	6-359

7	High-Level RC/V	7-1
	■ Introduction	7-1
	■ High-Level RC/V Tools	7-1
	■ Using High-Level RC/V Tools	7-4
	■ Default Files	7-7
	■ High-Level RC/V Database Forms Guide	7-8

8	RC/V Error Code Explanations	8-1
	■ Low-Level Access Error Codes	8-1

	Acronyms	AC-1
--	-----------------	----------------------

	Glossary	GL-1
--	-----------------	----------------------

	Index	IN-1
--	--------------	----------------------

Figures

1 Database Structure

- 1-1. ECD Database Linkages [1-3](#)
- 1-2. ECD Logical-to-Physical Device Translation [1-4](#)
- 1-3. ECD Device Layout Information [1-5](#)
- 1-4. ECD Device Specific Information [1-6](#)
- 1-5. Hardware Fault Information [1-8](#)
- 1-6. SG Database Disk Layout Hierarchy [1-11](#)

3 Representing Devices

- 3-1. ECD Unit Control Block Linkages for the 3B20D Computer [3-2](#)
- 3-2. ECD Unit Control Block Linkages for the 3B21D Computer [3-3](#)

4 Database Tools

- 4-1. PF Key Functions [4-7](#)

6 Database Form Guide

- 6-1. SCSI Tape Device (Partition Number Descriptor) — 3B21D Computer Only [6-224](#)

Tables

About This Information Product

- 1. IPs Supporting 3B21D Computer [xvi](#)

1 Database Structure

- 1-1. ECD Option Blocks [1-7](#)
- 1-2. ECD and SG Database Locations [1-12](#)

4 Database Tools

- 4-1. Form Navigation Commands [4-13](#)

5 Using Database Tools

- 5-1. Terminal Displays [5-7](#)

6 Database Form Guide

- 6-1. ECD and SG Forms [6-2](#)
- 6-2. SCSI Tape Device logdev Values [6-225](#)
- 6-3. u_model Values Top Units **ucb**
Forms [6-314](#)

7 High-Level RC/V

- 7-1. High-Level RC/V Forms [7-8](#)

About This Information Product

Purpose

The Recent Change and Verify Manual is part of a documentation set used to support the *UNIX*® Real-Time Reliable (RTR) operating system running on the Lucent Technologies 3B20D and 3B21D computers.

This Information Product (IP) is a guide for creating and maintaining the *UNIX* RTR operating system 3B20D and 3B21D computers Equipment Configuration Database (ECD) and System Generation (SG) database via the Recent Change and Verify (RC/V) system and other database tools. It does not discuss equipment growth or database evolution.

Audience

This IP provides information for a variety of audiences. Anyone interested in the *UNIX* RTR operating system and 3B20D/3B21D computers will find it useful. The primary audience for this IP includes the following:

- Software developers
- Hardware developers
- Training course developers
- Craft or maintenance personnel
- System administrators
- 3B20D and 3B21D computer documentation users.

*

Reason for Issue

This IP is being issued to incorporate new information for the 3B21D computer and to bring the IP into compliance with new Lucent Technologies documentation standards. The major changes are as follows:

- Change the six-digit Select Code IP number for the *Recent Change and Verify Manual* from 303-010 to the new nine-digit IP number 254-303-104.
- Add the following new forms to Chapter 6:
 - *btparm* [R21.1 & R21.3]
 - *btparm* [21.5L & Later]
 - *dciopt*.
- Delete *btparm* [21.1L & Later] form.
- Combine *logdev* [R21.1 & Later] form with *logdev* form in Chapter 6.
- Add *dciadd* and *dcidel* forms to Chapter 7.
- Update various RC/V forms in Chapters 6 and 7.
- Add and update error messages in Chapter 8.

Change bars on the far right-hand side of the text are used to denote changes made to the current issue. Change bars will help you understand at a glance what has changed from the previous issue. The position of deleted text is shown by a single asterisk (*). Not all editorial changes will be marked with change bars. Occasionally, change bars do not work in tables or figures; therefore, those changes will be shown by adding a change bar to only the table or figure title.

Whenever this IP is reissued, the reason(s) for reissue will be listed in this paragraph.

How to Use This Information Product

This IP is composed of independent chapters and reference information (Table of Contents, Acronyms, Glossary, and Index). Specific topics are referenced in the Table of Contents and Index with page number locations. The Acronyms and Glossary define the terminology used in this IP. A brief description of each chapter is as follows:

- Chapter 1 — Database Structure. Describes the structure of the ECD and the SG database, gives the location of each of the databases within the file system, and describes database life cycle.

- Chapter 2 — Logical-to-Physical Mapping. Describes how the *UNIX* RTR operating system maps from logical to physical devices via special device files. The user and driver views are included.
- Chapter 3 — Representing Devices. Shows the hierarchical structure of the ECD and describes how to represent devices in the ECD.
- Chapter 4 — Database Tools. Describes the tools used to create and modify the ECD and the SG database.
- Chapter 5 — Using Database Tools. Describes how to use database tools to change and verify a database, list database keys, print database forms, reclaim unused database space, build and apply batch scripts, make incore changes permanent, and move databases on the 3B20D and 3B21D computers. In addition, this chapter gives guidelines for routine maintenance.
- Chapter 6 — Database Form Guide. Illustrates each of the database forms used to review or modify the ECD and the SG database and gives a guide for filling out the forms.
- Chapter 7 — High-Level RC/V. Documents the high-level RC/V feature [also known as flexible Input/Output Processor (IOP)] capabilities. This chapter provides supplemental information to Chapters 4, 5, and 6 specific to high-level recent change.
- Chapter 8 — RC/V Error Code Explanations. Lists the RC/V error codes and their associated text.

Conventions Used

Command, Filename, and Display Notations

The following notations are used to show commands and filenames in the text and displays.

⇒ NOTE:

System filenames and command names are case-sensitive, so you must enter them exactly as they are shown.

- Command names in text appear in **bold** type; for example, the **/usr/bin/lis** command. Command names in headings appear in **bold** type.
- Filenames, forms, and records in text appear in *italic* type; for example, the *usr/lib/uucp/System* file, or *scsdbody* form or the *ucb* record. Forms and records in headings appear in ***bold-italics*** type.

- Text that you enter, such as a command or response to a prompt, appears in **bold** type; for example, the **ls -la** command. The convention of making text that you enter **bold** overrides other conventions; for example, enter **scsdbody** for the form name.
- Variables that appear in a command line or file appear in *italic* type; for example, **grep *username* /etc/passwd**. In this example, *username* is a variable indicating a user's name is required.
- Screen displays and system messages appear in `constant width` type; for example, `Please enter your password`. Program code listings and file listings are also shown in `constant width` type. Input messages are shown in `constant-width bold` type.
- Comments and explanations within a display are indented and shown in *italic* type. These are for information only and will not appear on your screen.
- A line in a file or on the computer screen that is too long to be shown as it actually appears in this IP will be shown with a backslash (\) at the end of the first line. This indicates the next line should be read as a continuation of the current line.
- Square brackets around an argument on a command line indicate that the argument is optional; for example, the **lpstat [-t]** command. In this example, the **-t** argument is optional and can be omitted.
- A vertical bar (|) between words in an argument on a command line indicates that one of the arguments is to be selected.
- The key identified on your keyboard as Return, Enter, or a bent arrow (↵) is referred to as the Return key. Occasionally, representations of this Return key will be boxed; for example, Return.
- There is an implied Return at the end of each command and menu response that you enter. Some examples do not explicitly show the Return. Where you may be expected to enter a Return (as in the case where you are accepting a menu default), the symbol <CR> is shown to indicate that you are to press the Return key.
- Key combinations appear in a hyphenated format; for example, Ctrl-d. Press and hold down the first key of a key combination while pressing the second key.
- Ellipses (three dots) on a command line indicate that the previous argument can be repeated; for example, **ls [*file* ...]**. In this example, multiple files can be listed after the command.
- References to manual pages are followed by their manual page location number in parentheses; for example, **mount(1M)**.

Hexadecimal Notation

Hexadecimal (base 16) numbers are denoted with a **0x** prefix; for example, 0x00A is decimal 10.

Signal Designations

The term “asserted” is used in the descriptions to mean that a signal is driven to its active state. The term “negated” is used in the descriptions to mean that a signal is driven to its inactive state.

Signal names used in the diagrams and descriptions that end in a 0 or 1 indicate the active state of the signal. Names ending in “0” are “active low” signals; names ending in “1” are “active high” signals. For example, CCIOD(31-00)1 describes the Central Control Input/Output Data bits 31 through 00 which are “active high” signals.

Equipment Locations

A coordinate numbering system is used to identify Equipment Locations (EQLs) in units and cabinets. The origin is the lower left front of the cabinet or unit. Vertical increments are measured in inches. Horizontal increments are measured in eighths of an inch. The coordinate location of a circuit pack is expressed as the horizontal and vertical location of the center lines of the connector into which the circuit pack is inserted. The location of a unit in a cabinet is identified by the placement of the lower left corner of the unit in the cabinet.

For example, a connector at EQL 004-080 is located 4 inches above the origin and 10 inches ($80 \times 0.125 = 10.0$) to the right of the origin.

Safety Labels

Safety labels are reminders used to assure the safety of personnel and to minimize service interruptions, loss of data, and damage to equipment, products, and software.

Three types of safety labels are used in Lucent Technologies documentation. The three types, in descending order of priority, are as follows:

1. **DANGER** indicates the presence of a hazard that **will** cause death or severe personal injury if the hazard is not avoided.
2. **WARNING** indicates the presence of a hazard that **can** cause death or severe personal injury if the hazard is not avoided.

3. **CAUTION** indicates the presence of a hazard that **will** or **can** cause minor personal injury or property damage if the hazard is not avoided.

This IP contains safety labels in the form of CAUTION statements.

Related Information Products

Table 1 lists by IP number and Select Code the Lucent Technologies IPs supporting the 3B21D computer. IPs with Select Codes will be converted to nine-digit IP numbers as they are reissued.

Table 1. IPs Supporting 3B21D Computer

NEW IP NUMBER	OLD SELECT CODE	TITLE
254-303-100	-	3B21D Computer Growth/Retrofit Tasks
254-303-101	-	3B21D Computer Routine Maintenance Tasks
254-303-102	-	3B21D Computer Trouble Clearing Tasks
254-303-103	303-007	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Processor Recovery Messages Guide
254-303-104	303-010	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Recent Change and Verify Manual
254-303-105	304-045	3B21D Computer Hardware Reference Manual
254-303-106	304-046	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System System Maintenance Manual
254-303-107	303-072	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Software Troubleshooting Guide
254-303-110	303-080	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Input Messages Manual
254-303-111	303-081	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Output Messages Manual
254-303-112	303-082	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Input Messages Manual
254-303-113	303-083	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Output Messages Manual

IP references will not include full titles; for example, “Refer to 254-303-105, *Hardware Reference Manual*, for more information.”

How to Comment on This Information Product

Lucent Technologies welcomes your comments on this IP. Your comments will aid us in improving the quality and usefulness of Lucent Technologies documentation. Please use the Feedback Form provided at the front of this IP. If the Feedback Form is missing, mail your comments to the following address:

Lucent Technologies
Customer Training and Information Products (CTIP)
2400 Reynolda Road
Winston-Salem, NC 27106-4606

Mandatory Customer Information

The 3B21D computer is used as the Administrative Module (AM) in various switching system applications and is not provided as a stand-alone product. Therefore, the application documentation is responsible for providing this information.

Refer to the applicable application documentation for “Mandatory Customer Information.”

Database Structure

1

Contents

Introduction	1-1
Low-Level Access Databases — File Conventions	1-2
ECD Structure	1-2
■ Logical-to-Physical Device Translation Information	1-4
■ Device Layout Information	1-5
■ Device Specific Information	1-6
■ Hardware Fault Information	1-8
■ Audit Control Information	1-9
■ Craft Interface Information	1-9
■ ULARP Information	1-9
■ Boot Time Parameters	1-10
SG Database Structure	1-10
■ Operating System Generation Information	1-10
■ Disk Layout Information	1-10
Database Location Conventions	1-11
Database Life Cycle	1-13
■ Booting	1-13
■ Making ECD Changes Permanent	1-13
■ Reclaiming Space in the Database	1-13
■ Building a New Boot Image	1-14

Database Structure

1

Introduction

The *UNIX*® Real-Time Reliable (RTR) operating system has two major databases.

- Equipment Configuration Database (ECD)
- System Generation (SG) database

The ECD describes the physical and logical configurations of the 3B20D and 3B21D computers and peripheral hardware. It also contains configuration and control information for the audit and craft interface software. The ECD is an integral part of the system. Both *UNIX* RTR operating system software and application [*5ESS*® switch, Operator Services Position System (OSPS), and so forth] software access the ECD for configuration information.

The SG database contains information needed to build a system disk, make a boot image, and make changes via field update.

The ECD and the SG database are accessed by the Recent Change and Verify (RC/V) system via a form on the terminal. Each form is a template or mask for input information. A guide for filling out each of the database forms is given in Chapter 6, "Database Form Guide."

The ECD and the SG database are Low-Level Access (LLA) databases. The LLA access primitives are used to access the ECD and the SG database.

Low-Level Access Databases — File Conventions

The LLA databases are Software Demand Paging (SDP) address spaces. The SDP allows secondary storage to be treated as main memory. The SDP address spaces consist of at least one file which contains the SDP control data used to manipulate the address space. Any other SDP file holds the data of the address space, and these files are created, as needed. SDP naming conventions use a common root name for all address space files. SDP adds its own suffixes to differentiate among the files composing the address space.

The suffix "<>" is added to the address space name to give the name of the file containing the SDP control data. Suffixes "<>0" through "<>n-1" represent the n files that hold the data portion of the address space. The limit n is determined by the compilation time configuration of SDP.

If the ECD is stored in an address space with the root *dmert.ecd*, the files needed for the ECD are as follows:

- *dmert.ecd<>* — The SDP control information for the data portion.
- *dmert.ecd<>0* — The data portion of the database address space.

There may be *dmert.ecd<>1*, and so forth, if needed. The SG database has similar files. The bootable ECDs on the 3B20D/3B21D computers are in flat-file format. Refer to "Database Location Conventions" in this chapter for further details.

ECD Structure

The information in the ECD is logically divided into eight parts.

- Logical-to-physical device translation information
- Device layout information
- Device specific information
- Hardware fault information
- Audit control information
- Craft interface information
- User-Level Automatic Restart Process (ULARP) information
- Boot time parameters.

The eight different parts of the ECD information are contained on different database forms. Figure 1-1 shows how different database forms are linked together. These linkages provide a path for information in the database to be retrieved.

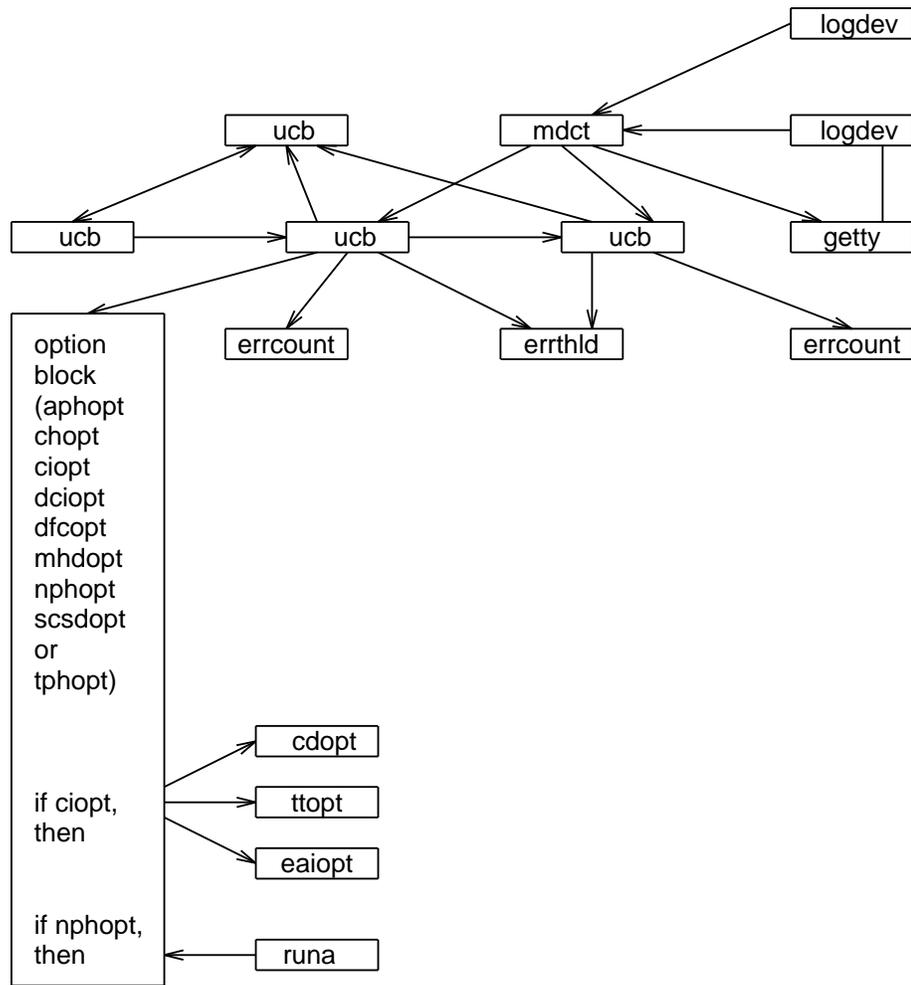


Figure 1-1. ECD Database Linkages

Logical-to-Physical Device Translation Information

Logical-to-physical device translation in the ECD is done via the minor device chain table (*mdct*), logical device information (*logdev*), and unit control block (*ucb*) forms (Figure 1-2).

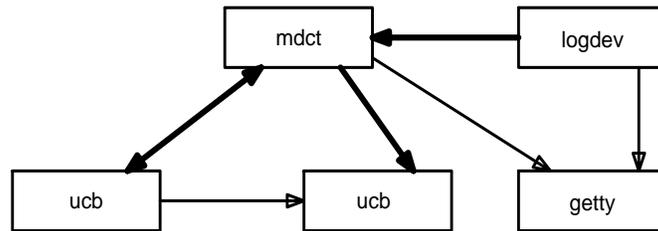


Figure 1-2. ECD Logical-to-Physical Device Translation

The *mdct*, which represents the logical device and maintains a list of active and backup devices, may reference many *ucbs*. For more details on logical-to-physical device translation, refer to Chapter 2, “Logical-to-Physical Mapping.”

The *ucb*, which represents the physical device, contains general information about a device; for example, device name, status, linkage, and so forth. If drivers for some devices require additional information, option blocks are used. The *ucbs* will reference option blocks only when necessary.

The *logdev* represents a special device file. The special device file identifies the device, driver, and type of device for logical devices. There may be many *logdevs* associated with a single *mdct*.

The *getty* form specifies the shell that is brought up for a particular device.

Device Layout Information

The device layout information is indicated via **ucb-to-ucb** connections (Figure 1-3).

⇒ NOTE:

For DCI applications, refer to 254-303-106, *System Maintenance Manual*.

Unit control blocks are connected in a hierarchical manner in the ECD. (Refer to Chapter 3, "Representing Devices.") These connections reflect the physical connections of the hardware. Four pointers are used in this linkage: two up pointers, one down pointer, and one side pointer. The up pointers point to the controlling devices (side 0 and side 1). The Input/Output Processor (IOP) and Disk File Controller (DFC) devices have two up pointers, all other devices have only one. The down pointer points to the first controlled device. A side pointer points to the device(s) at the same level that has the same controlling device.

Figure 1-3 uses the *ucb* to represent a controlling device with three controlled devices. This controlling device has one down pointer pointing to the first controlled device. The first and second controlled devices have side pointers pointing to the next controlled devices in the chain. The last controlled device in the chain does not have a device to point to; thus, it has no side pointer. All three controlled devices have one up pointer to the controlling device.

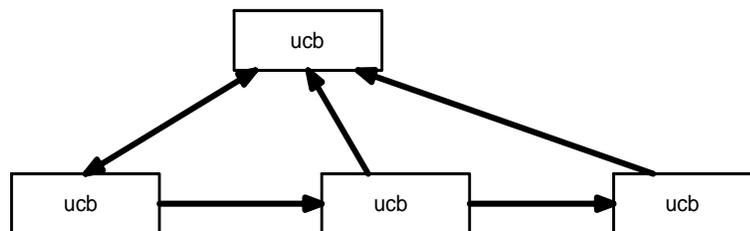


Figure 1-3. ECD Device Layout Information

The information associated with the driver's use of the *ucb* that cannot be modified using the recent change system is represented on the *ucbdyn* form.

Table 1-1 lists these option blocks and describes their contents.

Table 1-1. ECD Option Blocks

Option Block	Contents
<i>aphopt</i>	<i>aphuser</i> options: application type and maximum number of applications and session mapping.
<i>cdopt*</i>	Cathode Ray Tube (CRT) mapping and physical characteristic options. This option block is linked from a <i>ciopt</i> form.
<i>chopt</i>	IOP controller options.
<i>ciopt</i>	Craft-interface peripheral hardware options. This option block may also point to a <i>ttopt</i> , a <i>cdopt</i> , and an <i>eaiopt</i> .
<i>cpblx3</i>	Communication protocol options for a X.25 data link peripheral controller subdevice that handles only level 2 and level 3 protocols.
<i>cpphnone</i>	Communication protocol options for a X.25 data link peripheral controller subdevice that handles only level 2 protocols.
<i>dciopt</i>	Options that control operation of the Dual Serial Channel Computer Interconnect (DCI).
<i>dfcopt</i>	Options which control various aspects of DFC operation. Some <i>dfcopt</i> options are specific to the Small Computer System Interface (SCSI) DFC only.
<i>duopt</i>	Direct user interface options that are recent changeable.
<i>duoptdyn*</i>	Direct user interface options that are not recent changeable. This form is associated with a <i>duopt</i> form.
<i>eaiopt*</i>	Emergency Action Interface (EAI) options for the EAI terminal. This option block is linked from a <i>ciopt</i> form.
<i>mhdopt</i>	Options which control various aspects of Moving Head Disk (MHD) operation. Some <i>mhdopt</i> options are specific to the SCSI-MHD.
<i>nphopt</i>	Communications network protocol handler options for a X.25 data link peripheral controller subdevice that handles level 2 and level 3 protocols.
<i>runa*</i>	X.25 packet layer remote user network address information. This form is associated with <i>nphopt</i> forms.
<i>scsbody*</i>	Single Scanner and Signal Distributor (SCSD) point options. This form is associated with the <i>scsdopt</i> form.
* Other forms associated with option blocks that are not linked directly to a <i>ucb</i> .	

Table 1-1. ECD Option Blocks (Contd)

Option Block	Contents
<i>scsdmap*</i>	Scan point view. This form is associated with the <i>scsdopt</i> form.
<i>scsdopt</i>	SCSD options common to all scan points.
<i>tphopt</i>	Options and maximum number of transport and network connections, multiplex, and other parameters.
<i>ttopt*</i>	Teletypewriter (TTY) mapping and physical characteristic options. This option block is linked from a <i>ciopt</i> form.
* Other forms associated with option blocks that are not linked directly to a <i>ucb</i> .	

Hardware Fault Information

The hardware fault information (Figure 1-5) is contained in the error threshold (*errthld*) and the error counter (*errcount*) forms.

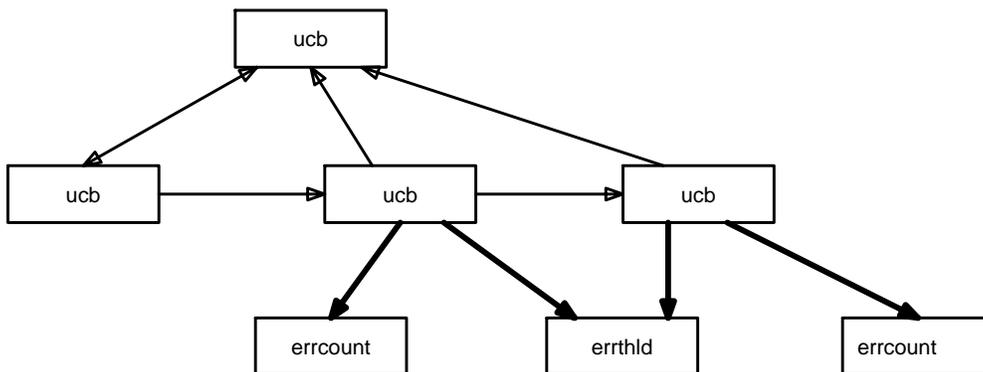


Figure 1-5. Hardware Fault Information

The *errthld* form specifies the number of faults that can occur in a given time frame before a unit is considered faulty. Hardware error numbers are also mapped into counters and thresholds. The *errcount* form tracks the beginning of the fault collection period and the number of faults that occur.

When a fault is detected, the time is noted. If the fault occurred in the previous fault-collection interval, the count is incremented and checked against the threshold. If the threshold is exceeded, maintenance activity is required. If the previous fault-collection interval has expired, the timers are reset, the counters are zeroed, and the particular fault count is incremented and checked.

Audit Control Information

Audit control information defines and controls audits and the associated recovery actions for errors detected. System Integrity Monitor (SIM) control information is specified on the *simcntl* form including the percentage of system resources to be spent on audits, and so forth.

Audit information is specified on the *audrec*, *auderr*, and *audinst* forms. Instance-independent information is specified on *audrec*; information about each possible audit instance is on *audinst*; error recovery for specific errors is on *auderr*. All *auderr* and *audinst* forms are associated with an *audrec* form. Dynamic information of an audit and audit instance can be reviewed on the *audstat* and *audhist* forms, respectively.

Craft Interface Information

The craft interface needs information about how to bring up craft shells and how to configure the craft output spooler. The *splinfo* form specifies whether the craft interface is Program Documentation Standard (PDS) or Man-Machine Language (MML) for the spooler. Spooler output devices are defined on the *device* form, and spooler classes are defined on the *classdef* form.

ULARP Information

The ULARP creates and monitors critical user-level processes or runs specified commands. Information about these processes is defined in the *ularp* form.

Boot Time Parameters

The *btparm* form (Chapter 6, “Database Form Guide”) contains boot time parameters (such as maximum processes concurrently allowed) which govern many aspects of the operating system’s size and performance. The point at which these changes take effect depends on the database containing the form and the bootstrap level. If the *btparm* form in the *incore* database is changed, a level 52 boot is required to have the values of the form take effect. If the disk copy of the database is changed, a level 53 or 54 boot is required. Refer to “Database Life Cycle” in this chapter for more details on booting and its implication on databases.

SG Database Structure

The SG database information is logically divided into two parts.

- Operating system generation information
- Disk layout information

Operating System Generation Information

The operating system generation information is contained in the *sgen2*, *process*, and *share* forms; for example, processes making up the boot image. The SG database can have more than one *process* or *share* form.

Disk Layout Information

The disk data may be stored on a single MHD (simplex MHD) or on paired MHDs (duplexed MHDs). Whether simplex or duplex, this disk data can be thought of as residing on a single virtual disk. There can be several simplex MHDs as well as several duplex MHDs on the computers. Also, there can be many virtual disks. The following information on disk layouts applies to each virtual disk.

The disk layout information is contained in the *pack*, *fs*, *fp*, *fsb*, and *fpb* forms (Figure 1-6).

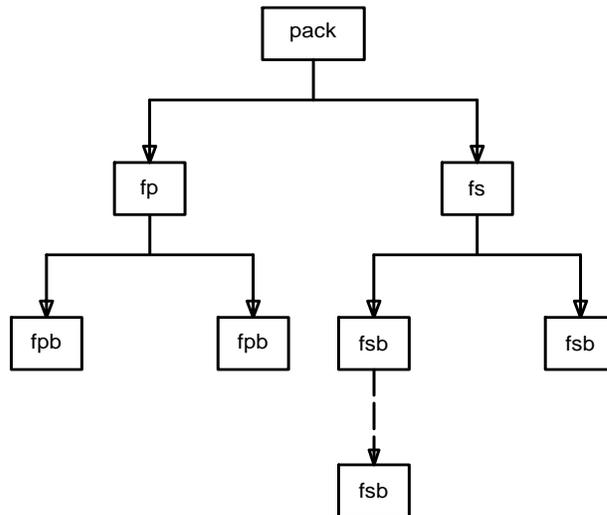


Figure 1-6. SG Database Disk Layout Hierarchy

Each virtual disk is represented in the SG database by a *pack* form. The *pack* form contains disk-specific information such as logical disk size (disk image size) and disk name. Each disk is logically subdivided into several disk partitions, each categorized as either a file system or file partition. A file system is represented by a file system (*fs*) form, and a file partition is represented by a file partition (*fp*) form. Each of these forms have a *pack* name field to relate it to a specific *pack* form.

A file system is composed of file system body (*fsb*) forms which specify directory entries or file entries that are to exist within the file system. An *fsb* form must specify the disk name, the file system to which the *fsb* belongs, and all the directories between the file system and *fsb* entity being described.

A file partition is composed of file partition body (*fpb*) forms which specify the information that is to exist within the file partition. An *fpb* form must specify the disk name and the file partition to which the *fpb* belongs.

Database Location Conventions

Multiple copies of the ECD and the SG databases reside on the computers. They are known by mnemonic names and reside in the locations shown in Table 1-2. These names are *incore*, *root*, *rootdmly*, *broot*, and *brootdmly*. All SDP files associated with an SG database are designated using *.sg**.

All ECDs are in flat-file format (all SDP files combined into a single contiguous file). The *incore* copy of the ECD always reflects the disk copy from which the system was last booted and any recent changes made since the boot. There is no *incore* copy of the SG database.

The mnemonic name will refer to the ECD when executing **rcvecd** and to the SG database when executing **rcvsg**.

Table 1-2. ECD and SG Database Locations

Database	Location
<i>incore</i> (ECD only)	The ECD which was loaded starting at memory segment 60 (decimal) during the last boot at level 3 (53) or 4 (54). The <i>incore</i> database is the only database used by the operating system. Changes to the <i>disk</i> database will not be reflected here until the next 53 or 54 boot.
<i>root</i>	The <i>application</i> databases in the file system are mounted in <i>/database</i> . They are <i>/database/appecd</i> and <i>/database/appdmert.sg*</i> . The <i>/database/appecd</i> will be loaded in main memory at boot time when option 33 is selected on the EAI page.
<i>rootdmly</i>	The minimum configuration databases in the file system are mounted in <i>/database</i> . They are <i>/database/ecd</i> and <i>/database/dmert.sg*</i> . The <i>/database/ecd</i> will be loaded in main memory at boot time when option 32 is selected on the EAI page.
<i>broot</i>	The <i>application</i> databases in the file system are mounted in <i>/broot/database</i> . They are <i>/broot/database/appecd</i> and <i>/broot/database/appdmert.sg*</i> .
<i>brootdmly</i>	The minimum configuration databases in the file system are mounted in <i>/broot/database</i> . They are <i>/broot/database/ecd</i> and <i>/broot/database/dmert.sg*</i> .

⇒ NOTE:

root and *broot* designate where the file systems are mounted. They **do not** designate the file system in which the database resides. For example, if */dev/bdb* is mounted in */database*, the mnemonic *root* refers to the application databases in the *broot* file system under the pathname */database/appecd* and */database/appdmert.sg**. *broot* and *brootdmly* will not attach unless a *root* or *broot* file system is mounted in */broot/database*.

To modify a database, the login must have write permissions for the database files and for the */etc* file system. This is the only protective measure taken by RC/V against unauthorized database access.

Database Life Cycle

Booting

During a boot, one of the *disk* ECDs is loaded in main memory and becomes the new *incore* ECD. Which ECD gets loaded depends on which options are selected on the Maintenance Teletypewriter (MTTY) terminal. If minimum configuration (EAI Command **32**) is selected, the ECD in */database/ecd* is loaded into main memory. Otherwise, */database/appecd* is loaded into main memory. Booting overwrites the previous *incore* ECD and destroys any changes that were not written to disk via the RC/V *activate* form.

⇒ NOTE:

Whenever the selection of minimum configuration is changed, the next boot should always be done with a 53 or 54 on the EAI to bring in the correct ECD, never with a 52.

Making ECD Changes Permanent

After an ECD has been booted, the operating system uses it to drive the computers. *RCVECD* can apply changes made to the *incore* database to a *disk* database via the *activate* form. This copies the *incore* ECD to an intermediate file and then switches it with the ECD last used in booting. On completion, the changes are permanent. The intermediate file used is */database/deltaecd*. Once created, this file should not be removed.

Reclaiming Space in the Database

A large number of recent changes deletes the space available for growth. To prevent the need for space from becoming critical, the database free space must be reclaimed at regular intervals. (The exact interval depends on the volume of recent changes.) To reclaim free space, use the dump and load procedure described in “Reclaiming Fragmented Database Space (**dbdump** and **dbload**)” in Chapter 5.

⇒ NOTE:

The database resulting from this procedure is in SDP address space format. Before such an ECD can be booted, it must be transformed into flat-file format via the **LOADF3B** command. (Refer to “**LOADF3B** Command” in Chapter 4.)

Building a New Boot Image

In order for changes to the SG database to take effect, a new boot image must be built from the SG database using the *UPD:BLDBOOT* input message. (Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.) The new boot image will reflect any changes to the system generation forms (*sgen2*, *process*, and *share*) in the SG database.

Logical-to-Physical Mapping

2

Contents

Introduction	2-1
Special Device Files	2-1
Driver View	2-2

Introduction

In the *UNIX*® Real-Time Reliable (RTR) operating system, the logical-to-physical translation is accomplished through the use of special device files (on the computer disk) and the following Equipment Configuration Database (ECD) forms:

- Minor device chain table (*mdct*)
- Unit control block (*ucb*)
- Logical device (*logdev*).

Special device files are created as part of the disk image produced by **3bmkdsk**. The **3bmkdsk** obtains information necessary to create special device files from the *logdev* forms.

Special Device Files

The *UNIX* RTR operating system high-level software knows nothing of the logical-to-physical device translation. A logical device can be represented as one or more special device files. These special device files consist solely of an inode. An inode is a data structure that contains information about a file. The following examples are of the types of information kept in an inode:

- Type of file; for example, directory, special, block, regular, and record
- Size of file

- Owner of file
- File permissions
- Location of file within the file system
- Device Control Number (DCN) which identifies the driver that controls the device
- Record Identifier (RID) of the associated *mcdt* (if any)
- Partition number (identifies subset of a particular device).

A partition on a disk is an example of a subset of a device. Special device files for logical devices identify the device, the driver, and the type of device. A device can be one of the following five types:

- c — Character — Typically, software devices are character devices.
- b — Block — A file system, file partition, and Moving Head Disk (MHD) are block devices.
- r — Record — A tape drive is an example of a record device.
- i — Input/Output Processor (IOP)* — Most devices in the IOP complex are considered IOP devices with some exceptions; for example, Synchronous Data Links (SDLs) are character devices.

⇒ **NOTE:**

For DCI applications, refer to 254-303-106, *System Maintenance Manual*.

- p — Pipe — A means of transferring information from the standard output of one process to the standard input of another process without the use of explicitly named files.

Driver View

Each driver accessing a device in the *UNIX* RTR operating system has a particular DCN associated with it. The following is a current list of drivers and their corresponding DCNs:

- **dfc** — The disk file controller. The DCN value is 4, and the pathname is */prc/bde*.
- **iop** — The input/output driver. The DCN value is 5, and the pathname is */prc/cdf* or */prc/bdf*.
- **memprc** — The memory driver. The DCN value is 8, and the pathname is */prc/cdi*. This driver has six software devices associated with it. These software devices (with 3B computer path, DCN, and partition number) are console device (*/dev/console, 8, 1*), sink device (*/dev/null, 8, 1*), kernel

memory (*/dev/kmem, 8, 0*), supervisor memory (*/dev/smem, 8, 2*), physical memory (*/dev/pmем, 8, 3*), and off-line memory (*/dev/offln, 8, 4*).

- **pcpmd** — The process control process. The DCN value is 12, and the pathname is */prc/cdm*.
- **ecdman** — The ECD manager process. The DCN value is 13, and the pathname is */prc/cdn*. The ecdman driver provides user-level access to the ECD.
- **dpz** — The display page driver for logical channel “cdz” [virtual channel 3 of the X.25 dedicated link to the Switching Control Center (SCC)]. The DCN is 6, and the pathname is */prc/bdg*.
- **spy** — The process which accesses kernel performance measurement information. The DCN is 16, and the pathname is */prc/cdq*.
- **fifo** — The First-In First-Out (FIFO) driver. The DCN is 0, and the pathname is */prc/fda*.

After the driver is created, the file manager passes information contained in the inode or the ECD *logdev* records to the driver via a message. To determine the device to be accessed, the driver must perform the following:

- The *mdct* RID and partition number are obtained from the ECD *logdev* records, when possible, or from disk inode entries. These are then used to obtain the specified *mdct*.
- Once the *mdct* is located, the *ucb* RIDs contained in the *mdct* can be used to obtain the desired *ucb* forms.
- The partition number has a different meaning for each device. Either it has no meaning or:
 - **MHD Devices:** The partition number identifies one of 64 logical partitions on a disk.
 - **SDL Devices:** The partition number identifies one of 32 possible lines on a synchronous data link.
 - **MTTY, TTY, SCC, and Read-Only Printer (ROP) Devices:** The partition number identifies the type of input message and priority of the input message.
 - **MT Devices:** The partition number identifies the tape options, such as density and the rewind status.

Contents

Introduction	3-1
Control Unit Complex	3-4
■ Control Unit Device	3-4
■ Central Control Device	3-5
■ Main Store Controller Device	3-5
■ Store Address Translator Device	3-6
■ Cache Store Unit Device	3-6
■ Utility Circuit Device	3-7
■ Direct Memory Access Device	3-7
■ Channel Devices	3-8
Dual Serial Channel	3-8
Serial Channel	3-8
Disk File Controller Complex	3-9
■ Disk File Controller Device	3-9
■ SCSI Bus	3-9
■ Moving Head Disk Device	3-10
■ SCSI Magnetic Tape Device (3B21D Computer Only)	3-11
Input/Output Processor Complex	3-12
■ Input/Output Processor Device	3-13
■ Maintenance TTY Controller Device	3-14
■ TTY Controller Device	3-14
■ Magnetic Tape Controller Device (3B20D Computer Only)	3-15
■ Synchronous Data Link Controller Device	3-15
■ Direct User Interface Controller Device	3-16

Contents

- Scanner and Signal Distributor Controller Device [3-16](#)
- TTY and Maintenance TTY Devices [3-17](#)
 - TTY and MTTY Special Device Files [3-18](#)
- Magnetic Tape Device (3B20D Computer Only) [3-19](#)
- Read-Only Printer Device [3-19](#)
- Synchronous Data Link Device [3-20](#)
- Switching Control Data Link Device [3-20](#)
- Direct User Interface Device [3-21](#)

Introduction

The Equipment Configuration Database (ECD) is an integral part of the *UNIX*[®] Real-Time Reliable (RTR) operating system. The *UNIX* RTR operating system and application software access the ECD for configuration information. This chapter details device information used to specify a 3B20D or 3B21D computer system. Figures 3-1 and 3-2 show the standard ECD UCB linkages for the 3B20D and 3B21D computers, respectively.

⇒ **NOTE:**

For DCI applications, refer to 254-303-106, *System Maintenance Manual*.

ECD connection rules for *mdct*, *ucb*, *errthld*, and *errcount* forms are as follows:

- A *ucb* does not always require an *mdct*. Use the following rules:
 - The device corresponding to a top *ucb* must have an *mdct* if diagnostics are performed on its subdevices and/or the device itself.
 - A device accessed through a special device file must have an *mdct* in addition to any diagnostic *mdcts*.
- If a *ucb* specifies an *errthld*, then an **errcount** is automatically inserted by Recent Change and Verify (RC/V) during a **trend**.
- Each of the three complex types [Control Unit (CU), Disk File Controller (DFC), and Input/Output Processor (IOP) for the *UNIX* RTR operating system] described in this chapter has one *errthld* record. The Moving Head Disk (MHD) group also has one *errthld* record.

- All devices in each complex are associated with the same *errthld*.
- A device must be associated with an *errthld* only if hardware fault information is to be kept.

The 3B20D and 3B21D computers have three main complex types: CU, DFC, and IOP.

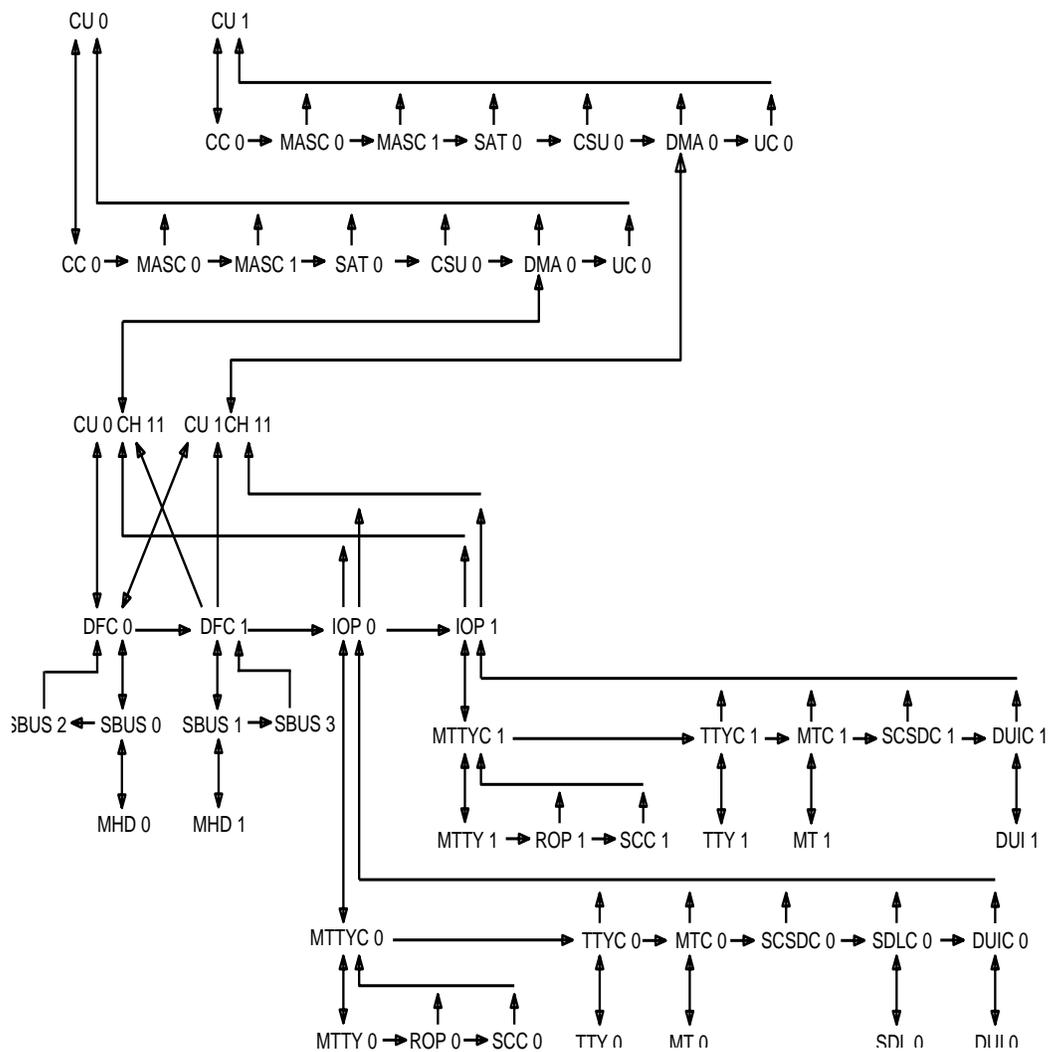
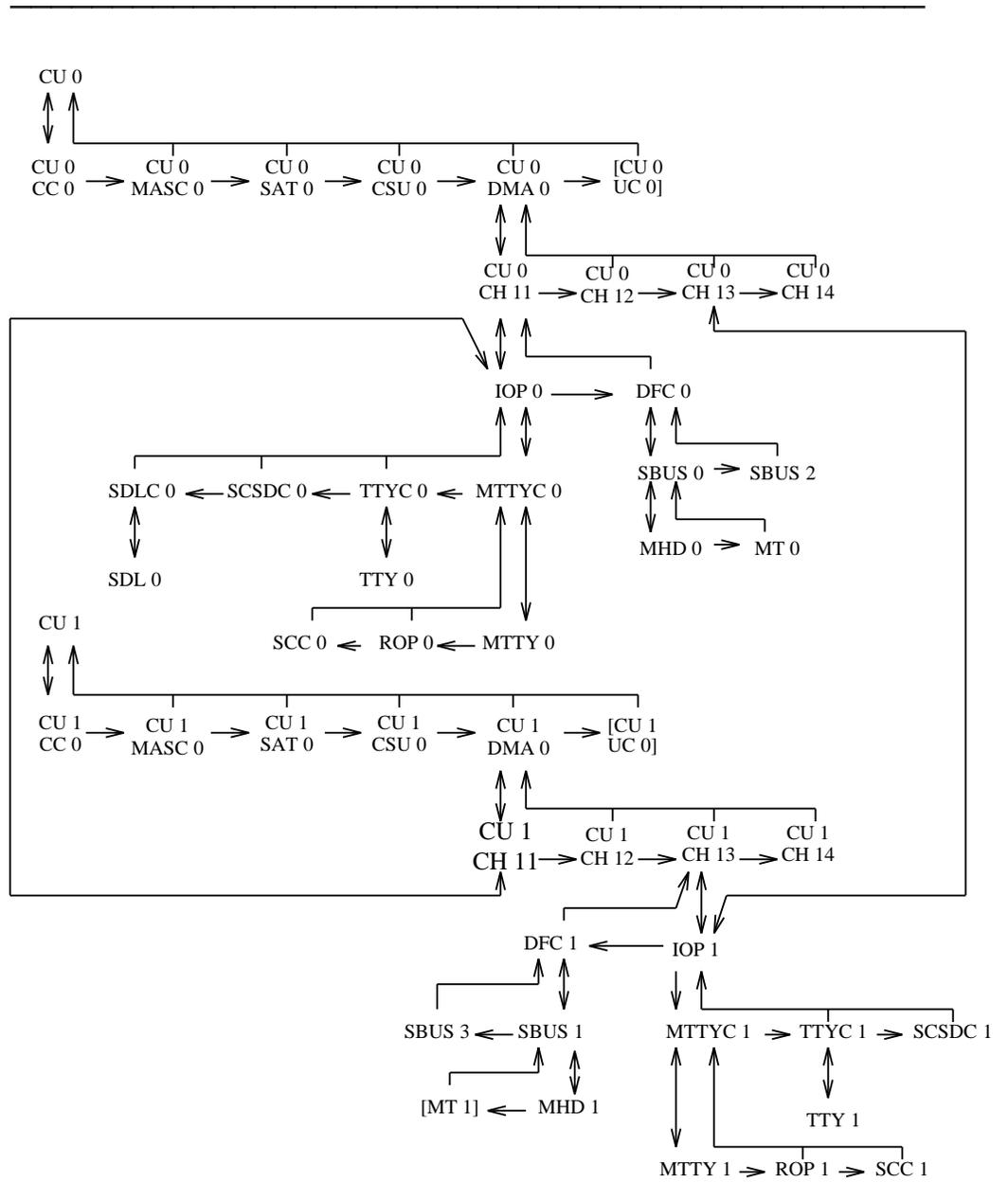


Figure 3-1. ECD Unit Control Block Linkages for the 3B20D Computer



Note: Unequipped units are shown in brackets.

Figure 3-2. ECD Unit Control Block Linkages for the 3B21D Computer

Control Unit Complex

The CU complex has the following major devices:

- Control Unit (CU)
- Central Control (CC)
- Main Store Controller (MASC)
- Store Address Translator (SAT)
- Cache Store Unit (CSU)
- Utility Circuit (UC)
- Direct Memory Access (DMA)
- Channels (CHs).

Unlike the other *UNIX* RTR operating system complex types (DFC and IOP) where each device within the complex is independent, the CU complex is considered one device. A device cannot be removed from a CU complex without affecting the entire CU complex; for example, if the MASC unit is removed from service because of a fault, the entire CU complex becomes nonfunctional. Therefore, when a device in the CU complex needs to be removed from service, the entire complex must be removed from service.

Control Unit Device

The CU device requires insertion of the *ucb*, *mdct*, and *errthld* forms. CU standard *ucb* default values are as follows:

device_type:CU	critical:y
essential:y	portswitch:n
top:y	posswitch:0
unique_unit:n	boot:n
pseudo_node:y	rexinh:n
restorable:y	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/cu
plant:y	

Default values for the *mdct* form associated with the CU device type are as follows:

logical_device:cu
mdct_type:T

Central Control Device

The CC unit controls operation of the 3B20D/3B21D computer by fetching a machine instruction from memory, decoding it, gating appropriate data paths to perform the correct operation, and preparing for processing the next machine instruction.

The CC device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the CC device. Central control standard *ucb* default values are as follows:

device_type:CC	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/cc
plant:n	

Main Store Controller Device

The MASC unit is divided in two parts: the MASC and the Main Store (MAS) array of boards. The MASC stores program instructions and data necessary for the 3B20D/3B21D computers to perform information processing.

The MASC device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the MASC device. MAS *ucb* default values are as follows:

device_type:MAS	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/masc
plant:n	

Store Address Translator Device

The SAT unit translates a virtual address into a corresponding physical address. The SAT device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the SAT device. SAT standard *ucb* default values are as follows:

device_type:SAT	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/sat
plant:n	

Cache Store Unit Device

The CSU is a high-speed memory device. The CSU requires insertion of the *ucb* form. The CU *errthld* form is also associated with the CSU device. CSU standard *ucb* default values are as follows:

device_type:CACHE	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/csu
plant:n	

Utility Circuit Device

The UC device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the UC device. UC standard *ucb* default values are as follows:

device_type:UC	critical:n
essential:n	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/uc
plant:n	

Direct Memory Access Device

The DMA unit moves data between Input/Output (I/O) devices and the main memory. The IOP and DFC devices have direct access to memory through the DMA unit. The DMA handles two Dual Serial Channels (DSCH). Each DSCH accommodates as many as 16 devices. Therefore, each DMA can handle up to 64 devices. The 3B20D/3B21D computers can be equipped with four DMAs (two per CU). The Direct Memory Access Controllers (DMACs) differ between the 3B20D and 3B21D computers as to the number of DSCHs and devices that can be equipped on the system.

The following table contains the configuration differences of the DMACs between the 3B20D and 3B21D computers:

3B20D Computer	3B21D Computer
2 DMACs/system	2 DMACs/system
2 channels/DMAC	4 channels/DMAC
16 devices/channel	4 devices/channel
32 devices/DMAC	16 devices/DMAC

The DMA device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the DMA device. DMA standard *ucb* default values are as follows:

device_type:DMA	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/dma
plant:n	

Channel Devices

There are two types of CH devices: DSCH and Serial Channel (SCH).

Dual Serial Channel

The DSCH device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the DSCH device. DSCH standard *ucb* default values are as follows:

device_type:DSCH	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/dmch
plant:n	

Serial Channel

The SCH device requires insertion of the *ucb* form. The CU *errthld* form is also associated with the SCH device. SCH standard *ucb* default values are as follows:

device_type:SCH	critical:y
essential:y	portswitch:n
top:n	posswitch:0
unique_unit:n	boot:n
pseudo_node:n	rexinh:n
restorable:n	errlog:n
removable:n	major_status:OOS
device_port:PCPEIH	pathname:cu/sch
plant:n	

Disk File Controller Complex

There are two types of DFCs used in the 3B20D/3B21D computer: the Storage Module Drive (SMD) DFC (3B20D computer only) and the Small Computer System Interface (SCSI) DFC.

The SMD DFC consists of the SMD DFC and SMD MHDs.

The SCSI DFC consists of the SCSI DFC, two SCSI Buses, SCSI MHDs, and SCSI Magnetic Tapes (MTs) (3B21D computer only).

Each hardware device [DFC, Small Computer System Interface Bus (SBUS), MHD, and MT] will have an associated *ucb* form.

Disk File Controller Device

The DFC handles the transfer of data between the main memory of the 3B20D/3B21D computers and the DFC subdevices.

Both the SMD and SCSI DFC devices require the insertion of the *ucb*, *mdct*, and *errthld* forms. Optionally, the *dfcopt* form may be used. However, some options on the *dfcopt* form are not applicable for the SMD DFC.

The SCSI and SMD DFC *ucb* forms are very similar. They are only different in the member version and up pointer linkage field values.

SCSI Bus

The SCSI Bus is used only in the SCSI-DFC type of DFC complex. The SCSI Bus connects one or more SCSI MHDs to the SCSI-DFC. For the 3B21D computer, the SCSI Bus also connects SCSI Magnetic Tapes (MTs) to the SCSI-DFC.

Moving Head Disk Device

The MHD unit stores large amounts of information which can be retrieved relatively quickly by the 3B20D computer via the DFC.

Both the SMD and SCSI MHD devices require the insertion of the *ucb*, *mdct*, *logdev*, and *errthld* forms. Optionally, the *mhdopt* form may be used. However, some options on the *mhdopt* form are not applicable for the SMD MHD.

The SCSI and SMD MHD *ucb* forms are very similar. They only differ in the member version and up pointer linkage field values.

Use the following procedure to add an MHD to the ECD:

Step	Procedure
1	Insert the unit control block(s) (<i>ucb</i> , physical entity) record for each MHD into the ECD. Each MHD being added will require its own <i>ucb</i> record. Two <i>ucbs</i> are inserted if a duplex pair is desired.
2	Insert a minor device chain table (<i>mdct</i> , logical entity) record for the MHD into the ECD. If a duplex pair of MHDs are being added, only one <i>mdct</i> is required for the pair. The <i>active_devices</i> field in the <i>mdct</i> must specify both disks of the pair (see " <i>mdct</i> " in Chapter 6 for details).
3	Insert one logical device (<i>logdev</i> , special device file) record for the MHD (or MHD duplex pair) into the ECD.
4	Optionally, insert an MHD option (<i>mhdopt</i>) record for each MHD into the ECD.
5	Insert one <i>pack</i> form (disk pack characteristics) for the MHD (or MHD duplex pair) into the SG database.

When these steps have been completed, use the *fs* (file system) forms and the *fp* (file partition) forms to dedicate space on the new MHD/MHD pair. These forms are inserted into the SG database, and each form must specify the pack name of the MHD/MHD pair.

SCSI Magnetic Tape Device (3B21D Computer Only)

The SCSI MT unit stores large amounts of information which can be retrieved relatively quickly by the 3B21D computer via the DFC.

The SCSI MT devices require the insertion of the *ucb*, *mdct*, *logdev*, and *errthld* forms.

device_type:DAT or SMT	critical:y
essential:n	portswitch:n
top:n	posswitch:_
unique_unit:y	boot:n
pseudo_node:n	rexinh:n
restorable:y	errlog:n
removable:n	major_status:OOS
device_port:DFC	pathname:pu/mt
plant:y	equipage:0x1d(DAT), 0x1a(SMT)

Default values for the *mdct* form associated with the SCSI MT device type are as follows:

```
logical_device:dk
mdct_type:m
```

Use the following procedure to add a SCSI MT to the ECD. The SCSI MT can also be added to the ECD by using the high-level *scsiadd* form.

Step	Procedure
1	Insert the unit control block(s) (<i>ucb</i> , physical entity) record for each SCSI MT into the ECD. Each MT being added will require its own <i>ucb</i> record.
2	Insert a minor device chain table (<i>mdct</i> , logical entity) record for the MT into the ECD.
3	Insert desired logical device (<i>logdev</i> , special device file) records for the MT into the ECD.

Input/Output Processor Complex

The IOP complex has the following devices:

- Input/Output Processor (IOP)
- Maintenance TTY Controller (MTTYC)
- TTY Controller (TTYC)
- Magnetic Tape Controller (MTC) (3B20D computer only)
- Synchronous Data Link Controller (SDLC)
- Direct User Interface Controller (DUIC)
- Scanner and Signal Distributor Controller (SCSDC)
- TTY and Maintenance TTY (MTTY)
- Magnetic Tape (MT) (3B20D computer only)
- Read-Only Printer (ROP)
- Synchronous Data Link (SDL)
- Switching Control Data Link (SCDL)
- Direct User Interface (DUI).

Input/Output Processor Device

The IOP device requires insertion of the *ucb*, *mdct*, *chopt*, and *errthld* forms. IOP standard *ucb* default values are as follows:

device_type:IOP	critical:y
essential:n	portswitch:n
top:y	posswitch:0
unique_unit:y	boot:n
pseudo_node:n	rexinh:y
restorable:y	errlog:n
removable:n	major_status:OOS
device_port:IOP	pathname:pu/iop
plant:y	

Default values for the *mdct* form associated with the IOP device type are as follows:

logical_device:io
mdct_type:T

I/O standard *chopt* default values are as follows:

handler_id:fpc
poll:0
time_out:0
picconf:0
pcoos:0xffff
comoos:0xf000
intvps:40
dgnflg:0x0

Maintenance TTY Controller Device

The MTTYC device requires insertion of the *ucb* and *mdct* forms. The IOP *errthld* form is also associated with the MTTYC device. MTTYC standard *ucb* default values are as follows:

device_type:MTTYC	boot:n
essential:n	rexinh:n
top:n	errlog:n
unique_unit:y	major_status:OOS
pseudo_node:n	pathname:pu/mttyc
restorable:y	h_type:0
removable:n	handler_id:ci83
device_port:IOP	packcode:t983
plant:y	issue:1
critical:y	version:0
portswitch:n	cvaddr:0xf900
posswitch:0	

Default values for the *mdct* form associated with the MTTYC device type are as follows:

logical_device:io
mdct_type:e

TTY Controller Device

The TTYC device requires insertion of the *ucb* form. The IOP *errthld* form is also associated with the TTYC device. TTYC standard *ucb* default values for support of maintenance (control display) terminals are as follows:

device_type:TTYC	boot:n
essential:n	rexinh:y
top:n	errlog:n
unique_unit:y	major_status:OOS
pseudo_node:n	pathname:pu/tyc
restorable:y	h_type:0
removable:n	handler_id:ci
device_port:IOP	packcode:t074
plant:y	issue:81
critical:y	version:20
portswitch:n	cvaddr:0x7f00
posswitch:0	

Magnetic Tape Controller Device (3B20D Computer Only)

The MTC device requires insertion of the *ucb* form. The IOP *errthld* form is also associated with the MTC device. MTC standard *ucb* default values are as follows:

device_type:MTC	boot:n
essential:n	rexinh:n
top:n	errlog:n
unique_unit:y	major_status:OOS
pseudo_node:n	pathname:pu/mtc
restorable:y	h_type:0
removable:n	handler_id:tape
device_port:IOP	packcode:u032, u052, or u145
plant:y	issue:0
critical:y	version:2
portswitch:n	cvaddr:0x7f00
posswitch:0	

Synchronous Data Link Controller Device

The SDLC device requires insertion of the *ucb* form. The IOP *errthld* form is also associated with the SDLC device. SDLC standard *ucb* default values are as follows:

device_type:SDLC	portswitch:n
essential:n	posswitch:0
top:n	boot:n
unique_unit:y	rexinh:n
pseudo_node:n	errlog:n
restorable:y	major_status:OOS
removable:n	pathname:pu/sdlc
device_port:IOP	h_type:0
plant:y	handler_id:sdl
critical:y	

Direct User Interface Controller Device

The DUIC device requires insertion of the *ucb* form. The IOP *errthld* form is also associated with the DUIC device. DUIC standard *ucb* default values are as follows:

device_type:DUIC	posswitch:0
essential:n	boot:n
top:n	rexinh:n
unique_unit:y	errlog:n
pseudo_node:n	major_status:OOS
restorable:y	pathname:pu/duic
removable:y	h_type:0
device_port:IOP	handler_id:dui
plant:y	
critical:y	
portswitch:n	

Scanner and Signal Distributor Controller Device

The SCSDC device requires insertion of the *ucb*, *mdct*, *scsdopt*, and *scsdbody* forms. The IOP *errthld* form is also associated with the SCSDC device. SCSDC standard *ucb* default values are as follows:

device_type:SCSD	critical:y
essential:n	portswitch:n
top:n	posswitch:0
unique_unit:y	boot:n
pseudo_node:n	rexinh:n
restorable:y	errlogin:n
removable:n	major_status:OOS
device_port:IOP	pathname:pu/scsdc
plant:y	

Default values for the *scsdopt* form for the SCSDC device type are as follows:

option_name:scopt0 or scopt1

TTY and Maintenance TTY Devices

The MTTY device provides a split-screen effect. The upper half of the screen continually displays processor status information. The lower half displays a scroll of consecutive craft input and output messages. Representing the MTTY involves insertion of eight forms in the ECD and the SG database. The MTTY device requires insertion of the *ucb*, *mdct*, *ciopt*, *cdopt*, *getty*, *eaiopt*, *logdev*, and *ttopt* forms in the ECD. The IOP *errthld* form is also associated with the MTTY device. Text RC/V assumes that the special device file for the MTTY area is */dev/ttya*. Use the following procedure to add the MTTY to the ECD:

Step	Procedure
1	Insert the Emergency Action Interface (EAI) option block (<i>eaiopt</i>) for the MTTY.
2	Insert the TTY option block (<i>ttopt</i>) for the MTTY.
3	Insert the control display option block (<i>cdopt</i>).
4	Insert two <i>getty</i> records for the MTTY.
5	Insert the craft interface option block (<i>ciopt</i>) for the MTTY.
6	Insert the unit control block (<i>ucb</i>) for the MTTY.
7	Insert the minor device chain table (<i>mdct</i>) for the MTTY.
8	Insert three <i>logdev</i> forms (special device files) for the MTTY.

MTTY standard *ucb* default values are as follows:

device_type:MTTY	posswitch:0
essential:y	boot:n
top:n	rexinh:n
unique_unit:y	errlog:n
pseudo_node:n	major_status:OOS
restorable:y	pathname:pu/mtty
removable:n	h_type:0
device_port:IOP	handler_id:ci83
plant:y	
critical:y	
portswitch:y	

Default values for the *mdct* form associated with the MTTY device type are as follows:

logical_device:io
mdct_type:t

TTY standard *ucb* default values are as follows:

device_type:TTY	posswitch:0
essential:n	boot:n
top:n	rexinh:y or n
unique_unit:y	errlog:n
pseudo_node:n	major_status:OOS
restorable:y	pathname:pu/tty
removable:n	h_type:0
device_port:IOP	handler_id:ci
plant:y	
critical:y	
portswitch:n	

Default values for the *mdct* form associated with the TTY device type are as follows:

logical_device:io
mdct_type:t

TTY and MTTY Special Device Files

Three special device files (*logdev*) are required to represent the MTTY.

- TTY Special Device File — Sends command requests for the requesting process to the controller for the MTTY. These requests can be in response to command requests issued by the craft and are considered low priority.
- Control Display Special Device File — Sends command requests from the requesting process to the controller for the MTTY. These requests can be in response to control display commands issued by the craft and are considered high priority.
- Acknowledgment Special Device File — Sends command requests from the requesting process to the controller for the MTTY to lock and unlock an MTTY keyboard. An MTTY keyboard is locked after a command is entered. When the craft command is executed and the response is displayed on one or more MTTYs, the MTTY is then unlocked.

All three special device files must specify the same *mdct* entry.

Magnetic Tape Device (3B20D Computer Only)

The MT device requires insertion of the *ucb* and *mdct* forms. The IOP *errthld* form is also associated with the MT device. The MT standard *ucb* default values are as follows:

device_type:MT	posswitch:0
essential:n	boot:n
top:n	rexinh:n
unique_unit:y	errlog:n
pseudo_node:n	major_status:OOS
restorable:y	pathname:pu/mt
removable:n	h_type:0
device_port:IOP	handler_id:tape
plant:y	
critical:y	
portswitch:n	

Default values for the *mdct* form associated with the MT device type are as follows:

logical_device:io
mdct_type:m

Read-Only Printer Device

The ROP device requires insertion of the *ucb*, *mdct*, *cdopt*, and *ciopt* forms. The IOP *errthld* form is also associated with the ROP device. ROP standard *ucb* default values are as follows:

device_type:ROP	posswitch:0 or 1
essential:n	boot:n
top:n	rexinh:n
unique_unit:y	errlog:n
pseudo_node:n	major_status:OOS
restorable:y	pathname:pu/ropp
removable:n	h_type:0
device_port:IOP	handler_id:ci
plant:y	
critical:y	
portswitch:y	

Default values for the *mdct* form associated with the ROP device type are as follows:

logical_device:io
mdct_type:e

Synchronous Data Link Device

The SDL device requires insertion of the *ucb*, *mdct*, *cpblx3*, and *cpphone* forms. The IOP *errthld* form is also associated with the SDL device. SDL standard *ucb* default values are as follows:

device_type:SDL	portswitch:n
essential:n	posswitch:0
top:n	boot:n
unique_unit:y	rexinh:n
pseudo_node:n	errlog:n
restorable:y	major_status:OOS
removable:n	pathname:pu/sdl
device_port:IOP	pcsdid:2
plant:y	h_type:0
critical:y	handler_id:sdl

Default values for the *mdct* form associated with the SDL device type are as follows:

logical_device:io
mdct_type:l

Switching Control Data Link Device

The SCDL device requires insertion of the *ucb*, *mdct*, *cpblx3*, and *cpphone* forms. The IOP *errthld* form is also associated with the SCDL device. SCDL standard *ucb* default values are as follows:

device_type:SCDL	posswitch:0
essential:y	boot:n
top:n	rexinh:n
unique_unit:y	errlog:n
pseudo_node:n	major_status:OOS
restorable:y	pathname:pu/sccp
removable:n	h_type:0
device_port:IOP	handler_id:sdl
plant:y	
critical:y	
portswitch:n	

Default values for the *mdct* form associated with the SCDL device type are as follows:

logical_device:io
mdct_type:e

Direct User Interface Device

The DUI device requires insertion of the *ucb*, *mdct*, and *duopt* forms. The IOP *errthld* form is also associated with the DUI device. DUI standard *ucb* default values are as follows:

device_type:DUI	critical:y
essential:n	portswitch:n
top:n	posswitch:0
unique_unit:y	boot:n
pseudo_node:n	rexinh:n
restorable:y	errlog:n
removable:y	major_status:OOS
device_port:IOP	pathname:pu/dui
plant:y	

Default values for the *mdct* form associated with the DUI device type are as follows:

logical_device:io
mdct_type:e

Default values for the *duopt* form for the DUI device type are as follows:

duiport:CCITT or RS449
duin1:32
duit1:100
duit2:255
duin2:4
duihdlc:0
duiwindow:4

Contents

Introduction	4-1
Conventions	4-1
CREATEECD and CREATESG Commands	4-2
■ Craft Shell Terminal	4-2
■ Options	4-2
■ Error Collection	4-3
RCVECD and RCVSG Commands	4-3
■ Craft Shell (Noncontrol Display) Terminal	4-3
■ Command Line Options	4-4
■ Return Values	4-6
■ Control Display Terminal	4-6
■ Database Forms Navigation	4-7
■ RC/V Session Flow	4-14
■ Serial Text Message RC/V	4-19
FORM Keyword	4-20
MSG Keyword	4-20
ABORTFORM Keyword	4-21
END Keyword	4-21
Verify FORM Operation	4-21
Recent Change FORM Operation	4-22
Keyword Data Pairs	4-22
SET Keyword	4-23
Syntax	4-24
Examples	4-24
■ Errors	4-27
LOADF3B Command	4-28

Introduction

This chapter discusses the tools used to perform the following:

- Create a skeleton database (**CREATEECD** or **CREATESG**).
- Execute a Recent Change and Verify (RC/V) session on a database on the 3B20D/3B21D computer (**RCVECD** or **RCVSG**).
- Change an Equipment Configuration Database (ECD) into bootable format (**LOADF3B**).

Conventions

The discussion of these tools uses the following conventions:

- [a] a is optional.
- [a | b] a or b or neither may be used.
- {a | b} a or b is required.
- (a)* a is repeated zero, one, or more times.

The options are always the same for the Program Documentation Standard (PDS) or Man-Machine Language (MML) terminals, but the formats are different. A prefix pathname is the part of the pathname to an ECD or a System Generation (SG) database that precedes the *.ecd** or *.sg**.

CREATEECD and CREATESG Commands

The **CREATEECD** and **CREATESG** commands are used to build a skeleton ECD and SG database. These skeleton databases have all the information Recent Change and Verify (RC/V) needs to begin populating the databases. If needed, the size of the sets allocated at this time for a particular form type can be specified. The databases generated by these commands are in angle bracket format and have separate files for the database data and database Software Demand Paging (SDP) control information.

Craft Shell Terminal

This includes the input message area of the MTTY terminal. **CREATEECD** and **CREATESG** commands are executed through **MENU** and RC/V craft shell interface. These commands are as follows:

In PDS,

```
RCV:MENU:{CREATEECD | CREATESG}, DB "dbname" (,FORMTYPE number)*!
```

In MML,

```
RCV:MENU:DATA, {CREATEECD | CREATESG}, DB="dbname"\  
(,FORMTYPE=number)*;
```

Options

DB *dbname*

The dbname is the full or partial prefix pathname of the database to be created. All databases are created in SDP address space format.

FORMTYPE *number*

FORMTYPE is used here to represent any of several available form types for which sets are created by **CREATEECD** or **CREATESG**.

number is the number of form instances required for that form type.

For **CREATEECD**, **FORMTYPE** can be replaced by **AUDINST**, **AUDREC**, **BXSLLOC**, **BXSLRMT**, **DEVICE**, **LOGDEV**, **MDCT**, **RUNA**, **UCB**, or **ULARP**.

For **CREATESG**, **FORMTYPE** can be replaced by **PACK**, **PROCESS**, or **SHARE**.

Set sizes can be specified for any or all of the legal form types.

Error Collection

Fatal errors terminate the process. Messages describing the errors are written to files *createecd.err* or *createsg.err*. These files are located in the */etc* directory on the 3B20D/3B21D computer.

RCVECD and RCVSG Commands

The **RCVECD** and **RCVSG** commands are used on the 3B20D/3B21D computer to do the RC/V functions.

The 3B20D and 3B21D computers have several RC/V interfaces. RC/V forms are available via craft shell (noncontrol display) terminals running MML or PDS and control display terminals (such as the MTTY). A serial text message interface is available via the craft shell running on craft shell terminals, the bottom of the control display terminals, and the Switching Control Center (SCC).

Craft Shell (Noncontrol Display) Terminal

The RC/V is executed through the use of **MENU** and RC/V craft shell interface. The commands are as follows:

In PDS,

```
RCV:MENU:{RCVECD | RCVSG} [,DB "dbname" | ,DBF "dbname" ]\  
[,ECHO "echofile" ] [,TERM termtype] [,MASKS "maskdir" ] [,PRINT "printfile" ]\  
[,HELP | ,NHELP] [,REVIEW | ,NREVIEW] [,PRCMSG | ,NPRCMSG]\  
[,JOURNAL | ,NJOURNAL] [,SYSGEN | ,NSYSGEN]
```

In MML,

```
RCV:MENU:DATA, {RCVECD | RCVSG} [, DB="dbname" | ,DBF="dbname" \  
[,TERM=termtype] [,MASKS="maskdir" ] [,PRINT="printfile" ] [,ECHO="echofile" ]\  
[,HELP | ,NHELP] [,REVIEW | ,NREVIEW] [,JOURNAL | ,NJOURNAL]\  
[,PRCMSG | ,NPRCMSG] [,SYSGEN | ,NSYSGEN]
```

Some typical commands are as follows:

```
RCV:MENU:DATA,RCVECD, DB= "incore";
```

This message allows you only to review the *incore* database (to change the database, you must specify **NREVIEW**).

```
RCV:MENU:DATA,RCVSG, DB= "rootdmly", NREVIEW!
```

This message allows you to make changes or to modify the minimum configuration SG database (pathname */database/dmvert.sg**).

RCV:MENU:DATA,RCVECD, DB= "root", NREVIEW, PRINT="/xxx/xxx"!

This message allows you to make changes, to modify the *root* database (*/database/appecd*), and to print the desired forms in the file */xxx/xxx*.

Command Line Options

If only the TERM or MASKS parameters or both have been specified or correctly defaulted, the remainder of the options are selected from the *rcvparams* form. (Refer to "Control Display Terminal" in this chapter.) If any other parameters have been specified, all options are taken from the command line (or defaulted). The options are as follows:

DB "dbname"

The *dbname* is the full or relative prefix pathname to an ECD or SG database. If the special mnemonics are not used, the databases are assumed to be in SDP address space format. Special mnemonics indicate certain ECDs or SG databases. These are *incore* (ECD only), *root*, *rootdmly*, *broot*, and *brootdmly*.

DBF "dbname"

(ECD only.) The *dbname* is the full or relative pathname to an ECD when the ECD is in flat-file format.

ECHO "echofile"

The name of a file where the keystrokes made during the RC/V session are saved. A full or relative pathname should be specified. If this file is to be used as batch input for a later RC/V session, care should be taken to avoid including backspace characters in the *echofile*.

TERM "termtype"

Default = vt100.

The type of terminal on which the RC/V session is to be run. Supported terminals are vt100, vt52, adm3a, adm42, hp2621, gnat, 5410, 5420, adm5, tty5410, tty5420, and dumb.

If vt100 is selected, the internal terminal setup options must be correctly initialized. Critical values for the On-line Data Integrity (ODIN) program are contained in the fourth grouping which must be 1010. To check the options, press the **Setup** key then **5**. To change the options, use the left cursor key to tab over to the desired flag and then press **6** to toggle its value. Pressing the **Setup** key returns the terminal to the normal mode of operation.

MASKS "maskmdir"

Default for *rcvecd* = *//la/ecdmasks*. Default for *rcvsg* = *//la/sgmasks*.

Full or relative pathname of the directory containing the ODIN masks.

PRINT "printfile"

Pathname of a file to which manual requests for printed form instances are directed. This file can then be printed to obtain hard copy. A full or relative pathname should be specified.

HELP | NHELP

Default = HELP.

HELP enables help messages at the bottom of the screen; **NHELP** disables them. The enabling/disabling of help messages can be toggled by entering % as the first and only character in a field.

REVIEW | NREVIEW

Default = REVIEW.

REVIEW mode allows only requests for information about the database. **NREVIEW** mode allows requests to modify the database as well as requests for database information.

PRCMSG | NPRCMSG

Default = PRCMSG.

PRCMSG is used to tell ODIN whether to print processing messages in the upper right corner of the CRT screen. **PRCMSG** indicates processing messages are to be printed; **NPRCMSG** indicates no processing messages.

JOURNAL | NJOURNAL

Default = JOURNAL.

JOURNAL prints form images of instances of inserts, deletes, updates, and executes done within a successful transaction block to the journal file. **NJOURNAL** turns off the printing. The journal file is */etc/rcvecd.jrnI[0,1]* or */etc/rcvsg.jrnI[0,1]* on the 3B20D/3B21D computers. In both cases, the journal is appended to in each session.

The command to dump the journal files is **OP:LOG:ECDLOG** or **OP:LOG:SGLOG**. For additional parameters, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Should it become necessary to delete the journal files, the **DEL:LOG:ECDLOG** or **DEL:LOG:SGLOG** command must be used. For additional parameters, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*. Under no circumstances should the journal files simply be removed. If this occurs, the associated Spooler Output Process (SOP), (that is, the SOP associated with ECDLOG for rcvecd or the SOP process associated with SGLOG for rcvsg) must be killed before journalling will continue properly.

SYSGEN | NSYSGEN

Default = NSYSGEN.

SYSGEN is a system generation mode that allows changes to be written into the database as applied rather than at the end of the transaction block. This option should be used only when executing a *dbload* or database evolution. **SYSGEN** is not allowed when attaching to the *incore* database.

NSYSGEN does not apply changes to the database until the end of the transaction block.

Return Values

RC/V can terminate with one of the following values:

- 0 Successful execution.
- >0 The return value gives the position in the command invocation of an incorrect input parameter.
- 1 RC/V fatal error. (Refer to Chapter 8, "RC/V Error Code Explanations.")
- 2 RC/V fatal error and unable to open error file.
- 10 Error during batch input.

Control Display Terminal

The **RCVECD** command is invoked from the control display area of a maintenance terminal by selecting menu option **199** when the cursor is at the *cmd* area at the top of the screen. Similarly, **RCVSG** is invoked via menu option **198**. Because this prohibits command line options, the *rcvparams* form is provided to set the session parameters. (Refer to "*rcvparams*" in Chapter 6.) This form must be successfully completed for the session to continue.

After the *rcvparams* form has been executed successfully, the remainder of the session is the same as on other terminals with the following exceptions:

- Type ahead is not allowed.
- The PF function keys are supported to allow you to suspend the RC/V session, perform some other work, and resume the RC/V session.

Each PF key has two states, a display and an input position. The state components are as follows:

- EAI Emergency action display
- DAP Current control display page
- RCV An RC/V page
- cmd Command area at the top of the screen

msg Shell input at the bottom of the screen
txt Inside an RC/V page.

As shown in Figure 4-1, when the system is booted, the Emergency Action Interface (EAI) page is displayed and the cursor is positioned in the *cmd* area of the Cathode Ray Tube (CRT) screen. Pressing the **PF3** key moves the cursor to the *msg* area. Alternatively, pressing the **PF2** key causes the control display page to be shown with the cursor in the *cmd* area.

The **MENU** command may also be used to invoke RC/V from the *msg* area provided both input and output have been redirected. (Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.)

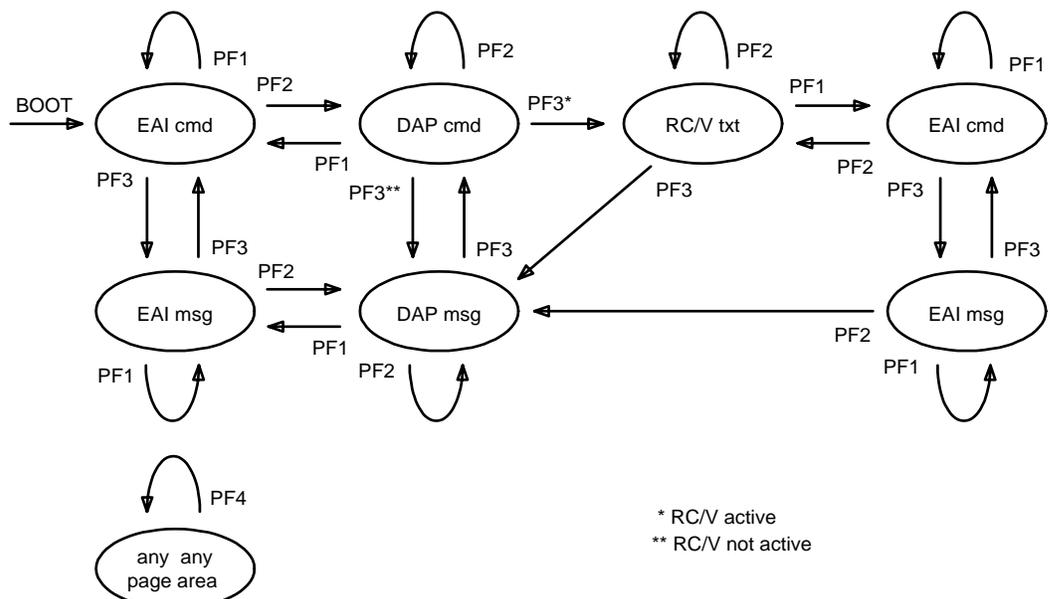


Figure 4-1. PF Key Functions

Database Forms Navigation

The form editor provides the terminal operator with commands which either affect the value of a field in a form or allow the operator to navigate (move the cursor) within a form (refer to Table 4-1). It also provides informational commands, such as turning on/off the display of help messages in a session. A few commands [abort (!), return (CR), and back tab (<)] act as responses to system prompts.

Commands are invoked by entering the appropriate character, that is, entering a character and pressing the <CR> key. On the MTTY, do not press the <CR> key after entering the help (?) command because it is acted on immediately. All the form navigation control characters are single, printable characters. They serve as control characters when entered as the first and only character in a field; otherwise, they are part of a field value. For example, the end of form command is invoked by the ~ character. Entering a single ~ invokes the command. Entering ~~ does not. It is a value, and field data checking occurs.

If a command is not meaningful, either in an input field or as a response to a system prompt, it is ignored and the cursor returns to its original position.

The following definitions are necessary to understand the list of form navigation commands.

- A form is a set of logically related data fields. There are three types of fields:
 - An *atomic* or simple field – a single piece of information; for example, “city.”
 - A *structure* – composed of several different atomic fields; for example, the structure “date” may contain the atomic field’s month, day, and year.
 - A *list* – a set of identical rows (called *elements*), each of which has one or more atomic fields and/or structures in it; for example, the list “checking-transactions” might have the fields “check#,” “date,” “description,” “debit,” “credit,” and “balance.” The “date” is a structure; the others are atomic fields.

- A compressed list is a list in which empty elements cannot be interspersed between nonempty elements. Regardless of where rows are entered in a compressed list, the data will be condensed so that all of the data is at the top of the list and all empty rows follow the data.

The following is a list of the form navigation commands along with their default control characters in parentheses. The commands are summarized in Table 4-1. Except where indicated (C), commands are not valid in the change mode.

Value Commands

- **123CR** — Enter a value into a field. Type the value (for example, **123**) followed by the <CR> key.
- **CR** — No value or response; skip this field or prompt. Pressing the <CR> key only (no characters entered) serves to retain the current value of an input field and proceed. For some system prompts, this signifies no response and results in the display of the next prompt or preceding prompt. For a required field, the message `Required Field!` flashes in the acknowledgment area and the cursor returns to the beginning of the field.

- **null item** (`) — Set this atomic field to the null (empty or missing) value. For an optional field, the current field is set to null and displayed as a string of underscore (_) characters. For a required field, the message `Required Field!` flashes in the acknowledgment area and the cursor returns to the beginning of the field.
- **default value** (;) — Set this atomic field to its default value. If a default value exists, it is assigned to the current field. If no default value exists, processing continues as if a null item had been entered.

Navigational Commands

*

- **back up** (^) — Back up to the previous field. The current field remains unchanged and the cursor moves to the previous field.
 - If the previous field is a structure or list, the cursor moves to the beginning of the structure or list.
 - If the previous field is on another screen (multiscreen forms), that screen is automatically displayed.
 - If used repeatedly within a list, the cursor backs up to the beginning of the current row and then backs up within the first column to the beginning of the list.
- **forward tab** (>) — Skip forward to the next tab stop (or retrieve next form). Form designers can set tab stops at certain fields in a form. This command leaves the current field unchanged and moves the cursor to the next tab stop.
 - If no tab stop exists between the current field and the end of the form, the cursor moves to the form operation prompt on the command line.
 - If the next tab stop is on another screen (multiscreen forms), that screen is automatically displayed.
 - If an empty required field is encountered before the next tab stop, the message `Required Field!` flashes in the acknowledgment area and the cursor moves to that field.
 - If the cursor is positioned at the beginning of a list row, the cursor moves to the next tab stop.
 - If the cursor is within a row, but not at the beginning, it is positioned at the beginning of the next row.
 - If entered into the first key field of a form and in a retrieval mode (review, update, and delete), the next form in the database is retrieved.

- **back tab (<)** — Back up to the previous tab stop or prompt. The current field remains unchanged, and the cursor moves to the previous tab stop.
 - If the previous tab stop is on another screen (multiscreen forms), that screen is automatically displayed.
 - If no previous tab stop exists, the cursor is positioned at the first data field.
 - If the cursor is positioned at the beginning of a list row, it moves to the previous tab stop.
 - If the cursor is within a row, but not at the beginning, it is positioned at the beginning of the current row.

- **end of form (~)** — Skip to the end of the form. The current field remains unchanged and the cursor moves to the form operation prompt on the command line. If it finds an empty required field enroute, the effect is the same as if the forward tab (>) command had been entered.
- **execute (*)** — Skip forward to the end of the form and perform the form operation. Processing continues as if the end of form (~) command had been entered, followed by the appropriate form operation code. When entered in response to a form operation prompt, it matches the current operation code.
- **abort (!)** — Abort this operation.
 - If entered into a data field, the active form is discarded, the “aborted” message flashes in the acknowledgment area, and the cursor moves to the first key field.
 - If entered in response to a command line prompt, it serves to abort the active form.
 - If entered while in change mode, it aborts the change attempt and the cursor returns to the form operation prompt.

List Value Commands

*

These commands are recognized only when entered at the first atomic field in a row of a list.

- **null row (`)** — Set the current row to null values. Each atomic field in the current row is set to the null value. The cursor returns to the beginning of the current row.
- **default row (:)** — Set the current row to default values. Each atomic field in the current row is set to its default value. The cursor returns to the beginning of the current row.

- **insert row (}** — Insert a new row at the current location (for compressed type lists only). Each row, from the current row to the end of the list, is “pushed down” and a null row is inserted at the current row location. The cursor returns to the current (null) row. If the list is “full” (no null rows), then the message `List is FULL!` flashes in the acknowledgment area and the cursor returns to the current row.
- **delete row ({** — Delete the current row (for compressed type lists only). The current row is discarded and each row, from the next to the last, is “pushed up” one row position. The last row is set to null values. The cursor returns to the current row location.
- **copy row (=** — Copy previous row. The previous row is copied to the current row. The cursor returns to the current row.

List Navigational Commands

*

These commands are recognized only when entered at the first atomic field in a row of a list.

- **previous row ([** — Back up one row. The cursor moves “up” to the beginning of the previous row in the list. If the current row is the first row, processing continues as if the back up (^) command had been entered.
- **next row (]** — Skip to the next row. The cursor moves “down” to the beginning of the next row in the list. If the current row is the last row, the cursor moves to the next field.
- **end of list (#** — Skip to the end of the list. The cursor moves to the next field following the current list.

Information Commands

These commands are recognized in all database operation modes when entered into a field but not in response to a system prompt.

- **help (?** — Display help information. The form on the screen is cleared and a full screen help page is displayed. Pressing the <CR> key clears the help page, the current form is redisplayed, and the cursor returns to its previous position.

- **help messages (%)** — Turn on/off help message display. In all cases, the cursor returns to the current field.
 - If help messages are currently being displayed, they will be turned off starting with the next atomic field.
 - If help messages are not currently being displayed and a help message exists for the current field, then the message will be displayed on the help line.
 - If no help message exists, the message `No help available` will flash in the acknowledgment area.

- **current mode (.)** — Display current database operation mode. The current database operation mode flashes as a message in the acknowledgment area (for example, `Insert mode`), and the cursor returns to the current field.

- **escape to shell (\)** — Enter a *UNIX*® system shell command. (This command is not available on the MTTY.) The prompt `Enter <return>`, `new shell command`, or `\:` is displayed on the command line, and a *UNIX* system command is entered by the operator. After the command is completed, the prompt is redisplayed and another command may be entered. This process continues until the `<CR>` key is pressed. Pressing the `<CR>` key redisplayes the current form and the cursor moves to its previous position. Entering `\` executes the previous shell command.

Table 4-1. Form Navigation Commands

Command	Description
data CR	Enter data value supplied into this field.
CR	On a form data field, keep current value of this item and go forward to next field. In response to a prompt, it means the same as <.
^	Go back to previous data item or processing level.
<	Go back to previous tab stop or to form selection prompt.*
>	Go forward to next tab stop.
!	Abort this form instance and start over.
~	Go to end of form.
[Go back to previous row.
]	Go forward to next row.
:	Set row to default values (C).
;	Set item to default value (C).
'	Set item to null value (C).
`	Set row to null value (C).
=	Copy last row (C).
#	Go to end of list.
}	Insert a new row (C).
{	Delete this row (C).
q	Exit edit mode.
%	Turn on/off help messages (C).
*	Perform intended action (for example, insert).*
.	Display current processing mode (C).
\	Escape to shell (not available on MTTY) (C).
?	Display form navigation command help page.*

* Certain interfaces (for example, SCC link and craft terminal) between RC/V and the user impact the response from the noted navigation commands because the command (that is, the input character) is read without receiving the carriage return. In these cases, pressing the <CR> key should be avoided in order to get the expected response.

RC/V Session Flow

When RC/V is successfully invoked, the CRT screen is cleared and the form selection prompt appears.

```
UNIX RTR RCV (ODIN) - Data Entry
=====
Enter Form Name: |
```

If the selection prompt does not appear, an error occurred in invocation. (Refer to "Errors" in this chapter.) In addition, the name of the database being accessed is displayed in the upper left corner of the screen, and the help message, may also enter '?' or 'help' to list forms, appears at the bottom of the screen. |

You can then enter:

?	or help	Lists all the form types in the database.	
form name		Name of the desired form.	
<, !, or <CR>		Return to the shell.	

If the form name is invalid, a diagnostic message is printed and the cursor is returned to the beginning of the input field. After the form name has been successfully entered (depending on whether the session is *REVIEW* or *NREVIEW*), two options are possible. If the session is *NREVIEW* and if you are inside a **trbegin-trend** pair, you will be given the operation selection prompt. |

```
Enter Database Operation |
I=Insert, R=Review, U=Update, or D=Delete: |
```

You can then enter:

i		This operation will be an insert.	
d		Operation will be a delete.	
u		Operation will be an update.	
r		Operation will be a review or a change-insert.	
!, , < or <CR>		Go to the form selection prompt.	

Invalid input is ignored, and the cursor is repositioned at the beginning of the input field.

If the session is *REVIEW* outside a **trbegin-trend** pair or for a review-only form, the operation code prompt is skipped and the session continues as if a carriage return had been entered. For an execute-only form, the operation code prompt is also skipped.

After an operation has been selected, the screen is cleared and the first page of the form is displayed. The cursor is positioned at the first field on the form. All forms have a key field to differentiate specific form instances. The key field is the first field on the form and must be filled in before processing continues.

If the first key field is **<**, processing returns to the form-select level. If the first key field is a **^**, processing returns to the specific operation-select level and the current form remains on display. If input for key fields or data is invalid, a diagnostic message is printed at the bottom of the form and the cursor is returned to the beginning of the field. After the key field is filled in, processing depends on the operation selected.

On a review operation if the key field specifies a nonexistent form instance, the message `FORM NOT FOUND!!` is printed in the upper right corner of the screen and the cursor is placed at the beginning of the form. If the key field specifies an existing form instance, the first page of the form is filled in and the cursor is placed at the bottom of the form. You can then enter:

page number	Display the requested page of the mask.
r	Return the cursor to the first field to review another form.
!	Abort the review.
v	Run ODIN level 1 and level 2 error checks.
p	If a printfile was specified, print this form instance.

The next option is valid if the review is done within a **trbegin-trend** pair.

c	Enter change-insert mode. (Refer to the following change-insert mode information.)
----------	--

The change-insert mode allows you to enter change mode and modify the key field and data values when a form is being reviewed. When a change-insert is requested, the rest of the interaction for the form is the same as an insert. However, using **!** to abort change-insert mode will return RC/V to review mode. A second **!** is required to abort review mode.

On a delete operation if the key field specifies a nonexistent form instance, the message `FORM NOT FOUND!!` is printed in the upper right corner of the screen and the cursor is placed at the beginning of the form. If the key field specifies an existing form instance, the first page of the form is displayed and the cursor is placed at the bottom of the form. You can then enter:

page number	Display this page of the mask.
d	Delete the form. If delete is successful, the message <code>FORM DELETED</code> appears in the upper right corner of the screen. If not, error messages are displayed at the bottom of the form.
!	Abort this delete.
v	Run ODIN level 1 and level 2 error checks.
p	If a printfile was specified, print this form instance.

On an update operation if the key field specifies a nonexistent form, the message `FORM NOT FOUND!!` is printed in the upper right corner of the screen and the cursor is placed at the beginning of the form. If you specify an existing form, the first page of the form is displayed and the cursor is positioned at the bottom of the form. You can then enter:

page number	Display this page of the form.
u	Perform the update. If the update is successful, the message <code>FORM UPDATED</code> appears in the upper right corner of the screen. If not, error messages appear at the bottom of the screen.
!	Abort this update.
v	Run ODIN level 1 and level 2 error checks.
p	If a printfile is specified, print this form instance.
c	Enter change mode. (Refer to the change mode information that follows.)
s	Enter substitute mode. (Refer to the substitute mode information that follows.)

On execute, the cursor is placed at the beginning of a form and you must fill in the fields one by one. After all the necessary fields are filled in, you can then enter:

- e** Execute the form. If the execute is successful, the message `FORM EXECUTED` appears in the upper right corner of the screen. If not, error messages appear at the bottom of the screen.
- !** Abort this form.
- p** If a printfile is specified, print this form instance.
- v** Run ODIN level 1 and level 2 error checks.
- c** Enter change mode. (Refer to the change mode information that follows.)
- s** Enter substitute mode. (Refer to the substitute information that follows.)

On an insert operation if the key field specifies a currently existing form, the message `DUPLICATE FORM!!` is printed in the upper right corner of the screen and the cursor is placed at the first field of the form. If the key field is unique, the cursor is placed on the first nonkey field and you can fill in the form by using the form navigation commands discussed earlier. After the form is filled in, the cursor is placed at the bottom of the form. You can then enter:

- page number** Display the indicated page.
- i** Insert the form. If the insert is successful, the message `FORM INSERTED` appears. If not, error messages are displayed at the bottom of the form.
- !** Abort the insert.
- v** Run ODIN level 1 and level 2 checks.
- p** If a printfile was specified, print this form instance.
- c** Enter change mode. (Refer to the change mode information that follows.)
- s** Enter substitute mode. (Refer to the substitute mode information that follows.)

In the change mode, you will be prompted with the following:

Change Field:

You may respond with the number of the data field to be changed, the name of the data field to be changed (see Note), or by pressing the <CR> key to terminate change mode. Chapter 6, "Database Form Guide," describes the field names and numbers for each form and the data fields to which they refer.

If the field name or field number entered is not a list, the cursor will be positioned to the field to be changed. When the data is entered, the cursor is returned to the change mode prompt.

If a field name or field number is entered and the item to be changed is a list, you will be prompted for the row in the list. The cursor is then positioned to the field to be changed. When the data is entered, the cursor is returned to the bottom of the screen. If the item to be changed is a list, you will be prompted for the next row in the list. Pressing the <CR> key returns the cursor to the change mode prompt.

In the substitute mode, you will be prompted for the field name (see Note) or number for which the substitution is to be made. The prompting occurs in the same manner as for the change mode described in the previous paragraphs. After the field name (or number) and the row number (if necessary) have been entered, you are prompted for the character pattern to be changed within the item. The prompt is as follows:

from:

⇒ NOTE:

Field names with underscores cannot be used in this way if the *ttopt* record for the terminal in use has the erase character set to underscore (this is the default setting).

An asterisk (*) will match the entire item. You will then be prompted for the character pattern to substitute. The prompt is:

to:

Any character(s) entered here will be exchanged for those entered in the *from* prompt.

Serial Text Message RC/V

An RC/V session can also be entered from any craft shell input area as a serial text message in either PDS or MML format. Standard craft shell syntax applies to input of text RC/V commands. Text RC/V is available from any shell input area including the bottom of the control display terminals (such as MTTY). **RCV:DMTECD** is used to access the ECD, and **RCV:DMTSG** is used to access the SG database.

Text RC/V is similar to forms RC/V (**trbegin**, **trend**, **dbdump**, and so forth) with the following exceptions:

- *REVIEW* [Verify (VFY)] sessions are limited to one form operation, and reviews are the only operation types allowed.
- *NREVIEW* [Recent Change (RC)] sessions do not allow reviews.
- Interactive feedback is on a line basis, not on data fields, and so forth. Errors will be flagged after a message line.
- All key fields must be entered (no defaults) before any data fields.
- Assigned default values are not viewable without a secondary operation (VFY).
- Review/change-insert mode is not supported.

A text RC/V session is a single, logical message that can be broken into one or more physical lines between keyword units. When the message is broken into multiple lines, a continuation character must be entered at the end of each line. The current continuation characters are a slash (/) when running under the PDS shell or an exclamation point (!) when running under the MML shell. At the end of continued lines, an acknowledgment is printed. This acknowledgment **OK** is a prompt for more input or the number of error message lines generated for the input line and a prompt for more input. A **?D**, **?E**, or **?I** acknowledgment indicates that the shell could not properly parse the current input line. If these acknowledgments are printed, text RC/V ignores the input line and allows you to continue.

A text session is logged at the SCC if it is entered from the SCC or the local MTTY. If a text RC/V message is entered from any other terminal, only a message indicating the start and end of RC/V activity is logged at the SCC.

Because the SCC logs input lines and their acknowledgments and output is always spooled to the originating terminal, no additional work is needed to log SCC originated RC/V messages. With MTTY originated messages, the Teletypewriter (TTY) name is used to differentiate the MTTY from other terminals. Text RC/V assumes that the MTTY has TTY name **ttya**.

A text RC/V session is divided into two processing segments: process parameters and process forms. The parameter processing must be first and must be successfully completed before forms processing is allowed. No parameter processing can be done once forms processing has begun. The valid text parameters are shown in this section. The parameter section follows the invocation and ends with the next END or FORM keyword. Note that these keywords cannot be entered on the first line of parameter input. The parameters are verified and errors are generated, as needed.

If errors are associated with a noncontinued line, the errors are output as a spooler message. The form produced as a result of a verify operation is also displayed via a spooler message at the termination of the verify.

Several special keywords are used in text RC/V. These keywords are not process parameter names and are not unique to any particular form; they are necessary for proper text forms processing.

FORM Keyword

Following successful parameter checks, forms are processed. A form operation begins with the FORM keyword. For verify sessions, the FORM keyword requires one argument, the form type name. For recent change sessions, the FORM keyword usually requires two arguments: the form type name and the name of the operation to be done on the form. These operation names is discussed later in this section. Once the FORM keyword and its arguments are entered correctly, you may then start entering form-specific data. This data must be entered in the form of *keyword data pairs* which is discussed later. If errors are found while processing a form, the rest of the input line is ignored and the number of errors is acknowledged. The form operation ends when the operation name is entered a second time. The name is used as an operation execution command. (Refer to "Examples" in this chapter.)

MSG Keyword

The MSG keyword reads error messages whenever an error condition occurs during form processing. To view the error message lines, enter the keyword MSG in response to the prompt. The acknowledgment is the next error message and a prompt for more input. This sequence is repeated for each input line. If **MSG** is entered after all error message lines have been acknowledged, you are so informed. If anything but **MSG** is entered after the prompt, the error acknowledgment mode is exited. If you do not view the error messages when prompted, the messages are lost.

NOTE:

The MSG keyword is recognized only when an error condition exists and you have been prompted with the number of error lines.

ABORTFORM Keyword

The ABORTFORM keyword terminates operations for a particular form instance and can be used when excessive errors are encountered. ABORTFORM allows you to choose a new form without terminating the session.

⇒ NOTE:

The ABORTFORM keyword is recognized only inside a form operation. Do not use this keyword at any other time during the session.

END Keyword

The END keyword is used for two purposes in a text RC/V session. First, it is used outside any form operation to check session termination status. A secondary use is to end parameter processing although the FORM keyword is normally used for this purpose.

When **END** is entered followed by the continuation character, the **OK** acknowledgment is displayed if the preceding portion of the message (for example, syntax) is correct. Otherwise, an error message is generated.

The END keyword will not terminate a session. Termination occurs when either a noncontinued line or shell execute character (! for PDS, ; for MML) is entered; but END should be used with the execute character because the shell cannot parse the execute character on a line by itself.

The END keyword may be used outside a form operation at any time. If it is entered inside a form operation, text RC/V views it as a field name for the current form. If the field name does not exist, an error message is generated.

Verify FORM Operation

The VFY FORM operation allows you to review data contained within a database. The VFY format is as follows:

In PDS,
FORM "*name*", *keyword_data_pair(s)*, **VFY** ...

In MML,
FORM="*name*", *keyword_data_pair(s)*, **VFY** ...

FORM is a keyword indicating that a form is to be reviewed.

name is the name of the form to be reviewed; for example, *eaiopt*, *audrec*, and so forth.

keyword_data_pair is the key field name and key field value for the form being reviewed. (Refer to "Keyword Data Pairs" in this chapter.)

VFY is the keyword that indicates the operation is to actually be performed. Only one form may be processed (reviewed) per VFY session.

Recent Change FORM Operation

An RC form operation changes the data stored within a particular database or obtains information concerning the database (*prtlst*, *dbinfo*, and so forth). The RC form operation has the following syntax:

In PDS,
FORM ("name", *opr_spec*), *keyword_data_pair*(s), *opr_exec* ...

In MML,
FORM=*"name"* & *opr_spec*, *keyword_data_pair*(s), *opr_exec* ...

FORM is the keyword indicating that a form is to be processed.

name is the name of the form to be processed, for example, *ucb*, *trbegin*, and so forth.

opr_spec specifies the type of operation to be performed.

keyword_data_pair is explained in the following text.

opr_exec causes the specified operation to be performed.

The *opr_spec* argument and the *opr_exec* parameter must be the same for any given form operation. (Refer to "Examples" in this chapter.) Valid operations for an RC session are as follows:

<i>XEQ</i>	execute
<i>CHG</i>	update
<i>NEW</i>	insert
<i>OUT</i>	delete

The default is execute (XEQ) if an execute-only form is to be processed. These operations, with exception of executes, can only be performed within a transaction block. (Refer to "Examples" in this chapter.) All form operations are not valid for all form types. (Refer to "Database Form Guide" in Chapter 6.) Unlike the VFY session, any number of form operations may be performed within a single session.

Keyword Data Pairs

A *keyword_data_pair* comprises most text RC/V operations. It can be in any one of the following formats:

In PDS,
keyword "data", ...
keyword, ...
SET ("keyword") ...
SET ("keyword", "data"), ...

In MML,
keyword="data", ...
keyword, ...
SET="keyword", ...
SET="keyword"& "data", ...

The first format is used to specify a field name (keyword) from the specified form and the value to be entered in that field (data). The second format is similar, but no data value is given for the named field. This format is used when a field is to be nulled or when a field name must be given but when that field contains no data, for example, parts of the *ucb* form key field. The third format is explained as follows.

SET Keyword

If the field name is part of a structure or list or if the field name contains nonalphanumeric characters such as brackets ([]), periods, and so forth, the SET keyword must be used. Any string containing nonalphanumeric characters must also be enclosed in double quotes ("). The SET construction consists of the SET keyword and up to two arguments. The first argument specifies the field name; the second argument specifies the associated data. If the field is to be nulled out, the second data field may be omitted. Some variations of the SET format are as follows:

In PDS,
SET ("*list.field*"), ...
SET ("*list.field*", "*data*"), ...
SET ("*structure.list.field*[*row*]"), ...
SET ("*structure.list.field*[*row*]", "*data*"), ...

In MML,
SET = "*list.field*", ...
SET = "*list.field*"& "data", ...
SET = "*structure.list.field*[*row*]", ...
SET = "*structure.list.field*[*row*]"& "data", ...

The SET keyword has other formats. The format may vary depending on the domain and form specifications of the specific form being used. All key fields for each form are listed in Chapter 6. Associated with each form field is the keyword to be used for text RC/V. Some fields listed in Chapter 6 also contain an asterisk (*). The asterisk is not part of the field name but is an indicator that the SET keyword must be used. Chapter 6 gives the correct format for any field of any form requiring the SET keyword.

The SET keyword is necessary because field names for structures and lists (*list.structure.field*[*row*]) are invalid keyword syntax.

Syntax

Recent changes and verifies are entered separately. The RC/V parameters listed are the same as those discussed in "Command Line Options." in this chapter.

The PDS shell verify:

```
RCV:{DMTECD DMTSG};VFY:{DB "dbname" DBF "dbname"}\  
[,MASKS "maskdir"]/ <return>  
[END,] FORM "formname", keyword_data_pairs, VFY [,END]*!
```

The MML shell verify:

```
RCV:{DMTECD DMTSG};VFY:DATA, {DB="dbname" DBF="dbname"}\  
[,MASKS= "maskdir"]! <return>  
[END,] FORM="formname", keyword_data_pairs, VFY [,END]*;
```

The PDS recent change:

```
RCV:{DMTECD DMTSG};RC:{DB"dbname" DBF="dbname" } [,MASKS "maskdir1"]\  
[,JOURNAL ,NJOURNAL] [,SYSGEN ,NSYSGEN]/ <return>  
[END,] {FORM "formname", keyword_data_pairs, XEQ \  
FORM ("formname",XEQ), keyword_data_pairs, XEQ \  
FORM ("formname",CHG), keyword_data_pairs, CHG \  
FORM ("formname",NEW), keyword_data_pairs, NEW \  
FORM ("formname",OUT), keyword_data_pairs, OUT }* [,END]*!
```

The MML recent change:

```
RCV:{DMTECD DMTSG};RC:DATA,{DB="dbname" DBF="dbname"}\  
[,MASKS="maskdir"] [,JOURNAL ,NJOURNAL] [,SYSGEN ,NSYSGEN]! <return>  
[END,] {FORM="formname", keyword_data_pairs, XEQ \  
FORM="formname"& XEQ, keyword_data_pairs, XEQ \  
FORM="formname"& CHG, keyword_data_pairs, CHG \  
FORM="formname"& NEW, keyword_data_pairs, NEW \  
FORM="formname"& OUT, keyword_data_pairs, OUT}* [,END]*;
```

Examples

The following examples illustrate text RC/V. The MML format is used here, but the session flow is the same for PDS. Each line in the following examples contains two parts: one is entered by the user and one is produced by the system. The part appearing to the left of the continuation character (!) is what the user would enter. Anything appearing to the right of the continuation character is produced by the system.

The first example shows a VFY session. The top part illustrates what the user would type in at a terminal. The bottom part is the system response after the END keyword was entered.

⇒ NOTE:

Data fields not enclosed in quotes will be translated to uppercase by the craft shell.

```
RCV:DMTECD:VFY:DATA,DB="root"!OK
:FORM="eaiopt"!OK
:OPTIONNAME="eaiop0"!OK
:VFY!OK
:END;PF
```

```
M 11 RCV DMTECD VFY DB root PAGE 1 OF 1
                                eaiopt          (1/1)
Emergency Action Interface Option (Recent Change and
Verify)
    1.option_name:eaiop0
    2.stime:2000      3.ltime:30000      4.cmon:10000
    5.sysname:LAB X      6.systype:3B/UNIX RTR
    7.genid:2.0.0.0      8.time_zone:CST6CDT
    9.eaiinit_hex:c
```

```
M 11 RCV DMTECD VFY DB root COMPLETED
```

The following examples show each of the RC operation types. For CHG, NEW, and OUT operations, a subsequent VFY operation is required to view the change in the database.

```
RCV:DMTECD:RC:DATA, DB="root" !OK
:FORM="trbegin", TRNAME=TR3, XEQ !OK
:FORM="classdef"& NEW !OK
:ALARM="n" !OK
:SET="devicelist.logicaldevname[1]"& "ttya" !OK
:SET="devicelist.logicaldevname[2]"& "rop0" !OK
:NEW !OK
:FORM="trend", TRNAME=TR3, XEQ !OK
:END !OK
:END;PF
```

```
RCV:DMTECD:RC:DATA, DB="root" !OK
:FORM="trbegin", TRNAME=TR4, XEQ !OK
:FORM="ucb"& OUT !OK
:KCOMPLEXNAME, KCOMPLEXNUMBER, KUNITNAME=TTYC,
KUNITNUMBER=8 !OK
:OUT !OK
:FORM="trend"& XEQ, TRNAME=TR4, XEQ !OK
:END !OK
:END;PF
```

```
RCV:DMTSG:RC:DATA, DB="root", NJOURNAL !OK
:FORM "dbdump"& XEQ !OK
:REQUESTFILE="/tmp/reqfle ", LOADFILE="/tmp/ldfle " !OK
:XEQ !OK
:END !OK
:END;PF
```

```
RCV:DMTECD:RC:DATA, DB="root", NJOURNAL !OK
:FORM="trbegin", TRNAME=TR2, XEQ !OK
:FORM="audinst"& CHG !OK
:AUDNAME=ECD, MEMNO=2, INSTNAME !OK
:TSUN="y" !OK
:T2100="y" !OK
:CHG !OK
:FORM="trend", TRNAME=TR2, XEQ !OK
:END !OK
:END;PF
```

The last example is an actual RC session where the user wants to change the contents of the database. The user enters the command line using RC instead of VFY. The user has specified that a journal file should not be produced; but where a journal file is desired and created, it can be found in */etc* along with the fatal error file. The fatal error file is discussed in the “Errors” section. The default journal filename is *rcvecd.jrn*[0,1] or *rcvsg.jrn*[0,1].

Like forms RC/V, text RC/V requires that changes occur within a transaction block. No operation type is specified for the *trbegin* form. This form is an execute-only form, and XEQ is the default for these form types. Multiple parameters can appear on the same line. The user attempts to enter an *eiiopt* form but changes his mind and uses the ABORTFORM keyword to cancel that group of changes. The next form illustrates entering required fields with no associated data and gives an example of using the SET keyword. Finally, the user tries to end the session but forgets to enter a *trend* form. If a continuation character is used, the user will receive one warning to enter a *trend* form. These changes will not actually be made to the database if the *trend* is not entered.

```
RCV:DMTECD:RC:DATA, DB="rootdmly", NJOURNAL !OK
:FORM="trbegin",TRNAME=TR5, XEQ !OK
:FORM="eaiopt" !?E 2 error message lines
:MSG !operation type must be specified for form eaiopt
:MSG !line ignored from keyword[1]: FORM
:MSG !?E no more error messages
:FORM="eaiopt"& NEW !OK
:OPTIONNAME="eaiopt2" !OK
:SYSNAME="LABY" !OK
:GENERICID="2.1.1" !?E 1 error message lines
:MSG !?D GENERICID: invalid keyword for /lla/ecdmasks/eaiopt
:GENID="2.1.1" !OK
:ABORTFORM !OK
:FORM="ucb"& CHG !OK
:KCOMPLEXNAME,KCOMPLEXNUMBER,KUNITNAME=IOP,KUNITNUMBER=0
!OK
:SET="side.unitname"& IOP !OK
:SET="side.unitnumber"& 1 ?OK
:CHG !OK
:END !?E 1 error message lines
:MSG !** WARNING ** trend must be performed
:FORM="trend"& XEQ,TRNAME=TR5 !OK
:XEQ !OK
:END;PF
```

Errors

RC/V has two types of errors: integrity errors caught by RC/V that can be corrected and internal fatal RC/V errors. The integrity errors produce diagnostic messages. You can correct these errors and continue this session. These errors are self-explanatory and are not discussed in the section.

RC/V fatal errors terminate the RC/V session. The cause of the fatal error is usually encoded in a string of error codes consisting of capital letters followed by two digits. (Refer to "RC/V Error Code Explanations" in Chapter 8, for the message text for error codes.) The error codes are written to */etc/rcvecd.err* or */etc/rcvsg.err* for forms RC/V and to */etc/dmtecd.err* or */etc/dmtsg.err* for serial text message RC/V. The error file is appended in each section. When it reaches a certain size (beyond which it is too large to edit), it is moved to a backup file (named as before, but suffixed with *.errbck* rather than *.err*) and a new error file is created. An error file may be dumped as follows:

In PDS,
DUMP:FILE:ALL, FN "/etc/rcvecd.err"!

In MML,
DUMP:FILE:DATA,ALL,FN = "/etc/rcvecd.err";

A list of fatal error messages is provided in Chapter 8, "RC/V Error Code Explanations."

The ECD also has a limited amount of space for growth. If growth space is not available, a sequence of error codes will be given beginning with RP02 (attempt to read beyond file configuration) and FA04 (error returned from readpage). For example:

```
RP02,FA04,US06,DC01,CR06, and CR02
```

indicates the ECD ran out of contiguous space while trying to create a record (error code CR02).

To recover growth space, perform a dump and load. [Refer to "Reclaiming Fragmented Database Space (**dbdump** and **dbload**)" in Chapter 5.]

LOADF3B Command

The **LOADF3B** command combines the files *prefix.ecd<>*, *prefix.ecd<>0*, and so forth. The output file is a contiguous (or flat) file. The commands are as follows:

In PDS,

```
RCV:MENU:LOADF3B,DB("dbname", dsize), LF ("lfname", lpoint [,growth])
```

In MML,

```
RCV:MENU:DATA, LOADF3B, DB="dbname" & dsize, LF="lfname"&lpoint [& growth];
```

Where:

<i>dbname</i>	The full or partial pathname of the ECD.
<i>dsize</i>	The SDP page size for the ECD, for ECDs, dsize = 512.

The **LF "lfname" lpoint [growth]** portion of the command specifies how the load file is to be produced.

<i>lfname</i>	The name of the file that will contain the flat-file ECD after LOADF3B has completed.
<i>lpoint</i>	The address of the segment that the ECD is loaded into during a boot [Current value: 7864320 (segment 60)].

growth The amount of space to be allocated for ECD growth. The database size plus *growth* cannot exceed the number of segments specified by the *ecd_segs* field on the *btparm* form. *growth* can be specified directly as the number of bytes (truncated to page size) or as the percentage of database size (truncated to page size). It can also be specified indirectly as the total ECD size in segments, and **LOADF3B** will calculate the amount of growth space as the remainder of the total size minus the database size. If *no growth* is specified, the default value is equal to whatever extra space is left in the last segment that contains data; no extra segments will be allocated.

Contents

Creating a Database	5-1
Verifying a Database	5-1
Modifying Recent Change Database Items (trbegin and trend)	5-1
Aborting a Transaction (trabort)	5-2
Listing Keys (klist)	5-3
Printing Forms (prtlist)	5-3
Reclaiming Fragmented Database Space (dbdump and dbload)	5-3
Finding All UCBs Under a Specific UCB	5-4
Listing UCBs Under a Specific IOP	5-4
Finding All Links to a Specific Form	5-5
Finding the RID of a Specified Form	5-5
Applying a Batch Script	5-5
Making Incore Changes Permanent (activate)	5-5
Placing an ECD in Flat-File Format	5-5
Moving Databases	5-6
Performing Routine Maintenance	5-6
Color RC/V	5-6
■ Black and White Terminal	5-8
■ Color Terminal	5-8
Example of an RC/V Session	5-8

Using Database Tools

5

Creating a Database

To create an Equipment Configuration Database (ECD) or System Generation (SG) database, use the **CREATEECD** or **CREATESG** command. (Refer to “**CREATEECD** and **CREATESG** Commands” in Chapter 4.)

Verifying a Database

To verify a database, invoke Recent Change and Verify (RC/V) with the *REVIEW* option. When the RC/V session starts, request the form(s) that should be verified. When the form is displayed, enter the keys of the specific form instances to be reviewed.

Modifying Recent Change Database Items (trbegin and trend)

Typically, the ECD or the SG database recent changes consist of many operations spread over many forms. RC/V groups logically consistent sets of these changes into an entity called a transaction. Within transaction blocks (NSYSGEN), changes are verified as a whole and are applied to the database as a whole. If any part of the set of changes causes database inconsistency, none of the changes are applied. This maintains database integrity.

RC/V requires transactions to start with the *trbegin* form and end with the *trend* form. The *trbegin* form indicates to the system that changes are going to be made to the database. The *trend* form verifies that the changes made are logically consistent both among themselves and with the rest of the database. If the changes pass the checks, *trend* applies the changes to the database. In the SYSGEN mode, *trbegin* and *trend* are still required for changes, although they do not bracket a transaction and all changes are applied as made. The *trend* form is used in the SYSGEN mode to run consistency checks and error reports.

To apply recent changes to the database, invoke RC/V with the *NREVIEW* option. (Refer to “**RCVECD** and **RCVSG** Commands” in Chapter 4.) A *trbegin* (execute-only) form must be entered before the first *update*, *delete*, or *insert* (*reviews* are allowed any time).

After all recent changes have been entered, verify their consistency and apply them to the database by using the *trend* (execute-only) form. The data fields should all be defaulted (press the <CR> key). Overriding these defaults turn off checks and can cause an inconsistent database. Executing the *trend* will run the checks, and if they are successful, apply the changes to the database. If they are unsuccessful, you remain in the transaction and have the option of making the recent changes required to fix the inconsistencies or of aborting the transaction. To abort the transaction, see the following section.

Aborting a Transaction (trabort)

There are two ways to abort a transaction. One way is to exit RC/V without completing a *trend* form. Aborting the transaction this way requires two consecutive attempts to exit RC/V because the first attempt will produce a warning message.

The *trabort* (execute-only) form may also be used to abort a transaction. This method is often preferable because it does not require the user to exit RC/V. Executing the *trabort* form simply “erases” all database modifications made since the last *trbegin* (in NSYSGEN mode). The key to this form is *tr_name* which defaults to TRABORT.

⇒ NOTE:

In the SYSGEN mode, aborting a transaction can result in an inconsistent database.

Listing Keys (klist)

Form instances are retrieved by key(s). To list all keys associated with a particular form type, use the *klist* (execute-only) form. The *klist* form is only valid outside a *trbegin-trend* pair.

The key for the *klist* is the name of the file to which the keys for the indicated form is printed. After the key has been entered, fill out the form by using the value **y** for those form types whose keys are to be printed. When the execution is successful, the file will contain all keys for each instance of the selected form types. Exit RC/V to view this file.

Printing Forms (prtlist)

The *prtlist* form prints form instances of key-based forms and uses the output of *klist* as input. The first key field, *keylist_file*, is the name of a file that was previously produced by *klist*. The second key field, *print_list_file*, is a file where the form instances for the keys will be printed. [Refer to “Listing Keys (*klist*)” in this chapter.] The form *prtlist* is execute-only and is valid only outside a **trbegin-trend** pair.

Form instances can also be printed on request by using the *PRINT printfile* option on the **RCVECD** or **RCVSG** input message. In an RC/V session, particular form instances can be printed in the printfile by entering **p** in the appropriate places during the session.

Reclaiming Fragmented Database Space (dbdump and dbload)

To reclaim fragmented database space, run a dump and load. To perform a dump, use the *dbdump* (execute-only) form. The *dbdump* form key is the name of a file that is used as a work area and the name of the dump file to be produced. When the **dbdump** execute is successful, the dump file contains an American Standard Code for Information Interchange (ASCII) representation of all data in the database including a prepended **trbegin** and an appended *trend*. The **dbdump** must be done outside a **trbegin-trend** pair and cannot be done on the *incore* database.

This data must be inserted into a new ECD or SG database. First, generate the new database by using the **CREATEECD** or **CREATESG** command. Then use RC/V to attach to this database.

Load the database by using the *dbload* (execute-only) form. The *dbload* form must be done outside a **trbegin-trend** pair and must also be done in the SYSGEN and NREVIEW mode. One key field, *load_file*, should be the file produced by a **dbdump**. The **trend** appended to this file by *dbdump* will resolve links and determine database consistency. The other key field is a file where forms that do not load because of errors are printed. Always examine this file when the load terminates. When the execute is successful, all records are loaded into the database.

To reclaim space in a fragmented *incore* database, the following steps are required.

1. Activate the *incore* database to the disk. [Refer to “Making Incore Changes Permanent (**activate**)” in this chapter.]
2. Dump the *disk* database just activated (**dbdump**).
3. Create a new database using **CREATEECD** or **CREATESG**.
4. Load the new database (**dbload**).
5. Use **LOADF3B** to make a flat, bootable database. (Refer to “**LOADF3B** Command” in Chapter 4.)
6. Replace the old *disk* database with the newly loaded database. (Refer to “Moving Databases” in this chapter.)
7. Boot the system at level 3 (command **53**) or level 4 (command **54**).

⇒ NOTE:

The **dbdump** and **dbload** may or may not work in a split lab configuration. See a lab administrator for instructions in a split lab environment.

Finding All UCBs Under a Specific UCB

Use the *dbinfo* form to list all Unit Control Blocks (UCBs) under the specified UCB. The key for *dbinfo* is the file in which the list is generated. Select the *ucb_list* option and fill in the *ucb* key. Exit RC/V to examine this file. The *dbinfo* is an execute-only form that must be used outside a transaction block.

Listing UCBs Under a Specific IOP

Use the *dbinfo* form to list UCBs under an Input/Output Processor (IOP). The key is the file in which output will be generated. Select the *IOP_list* option and fill in the *ucb* key. Exit RC/V to examine the file. The *dbinfo* is an execute-only form that must be used outside a transaction block.

Finding All Links to a Specific Form

The *dbinfo* form can be used to list the forms that have links to a given form. The key is the file in which output will be generated. Select the *pointer_list* option and supply the form type and its key. Exit RC/V to examine the file. The *dbinfo* is an execute-only form that must be used outside a transaction block.

Finding the RID of a Specified Form

The *dbinfo* form can be used to retrieve the Record Identifier (RID) of a requested form. The key is the file in which output will be generated. Select the *get_form_rid* option and supply up to five form types (*type_of_form*) and their key fields. RC/V must be exited to examine the file. The *dbinfo* is an execute-only form that must be used outside a transaction block.

Applying a Batch Script

Standard input must be redirected from the batch input file. Standard output should be redirected to */dev/null*. (If the standard output is not redirected, the forms show on the screen as the batch script executes.) If the *ttopt* record of the terminal on which RC/V will be run has the erase character set to underscore (*_*), the script should not contain references to field names containing underscores, unless standard output is redirected. Always examine the error files to determine if the batch input was successful.

Making Incore Changes Permanent (activate)

To write the *incore* copy of the database to disk, use the *activate* (execute-only) form. This form must be completed outside a **trbegin-trend** pair and in the NREVIEW mode. Enter **YES** in the key field (*copy_inc_to_disk*). When the execute is successful, the *disk* database contains the information that is in the *incore* database.

Placing an ECD in Flat-File Format

To place the ECD in flat-file format, use the **LOADF3B** command. (Refer to “**LOADF3B** Command” in Chapter 4.)

Moving Databases

To move *tmpecd*, *tmp.sg<>*, and *tmp.sg<>0* over the *root* databases, perform the following:

In PDS,

```
COPY:FILESYS:FILE, SRC "tmp.sg<>", DEST "/database/appdmert.sg<>"!  
COPY:FILESYS:FILE, SRC "tmp.sg<>0", DEST "/database/appdmert.sg<>0"!  
COPY:FILESYS:FILE, SRC "tmpecd", DEST "/database/appecd"!  
COPY:FILESYS:CFILE, FN "/database/appecd"!
```

In MML,

```
COPY:FILESYS:DATA, FILE, SRC="tmp.sg<>", DEST="/database/appdmert.sg<>";  
COPY:FILESYS:DATA, FILE, SRC="tmp.sg<>0", DEST="/database/appdmert.sg<>0";  
COPY:FILESYS:DATA, FILE, SRC="tmpecd", DEST="/database/appecd";  
COPY:FILESYS:DATA, CFILE, FN="/database/appecd";
```

Performing Routine Maintenance

One maintenance activity must be performed regularly. The exact interval depends on the volume of recent changes. Free space in the database must be reclaimed.

A large number of recent changes deletes the space available for growth. To prevent the need for space from becoming critical, the database free space must be reclaimed at regular intervals. To reclaim free space, use the dump and load procedure described in “Reclaiming Fragmented Database Space (**dbdump** and **dbload**)” in this chapter.

Color RC/V

The following text, which describes the implementation of color RC/V, can be used as a guide to the different areas of the screen controlled by On-line Data Integrity (ODIN) and to the colors used in these areas.

Color RC/V will work on a black and white terminal without changes.

Table 5-1 gives the ODIN display field type, the black and white representation, and the color for the color terminal. Notice the last line is always underlined.

The RC/V terminal interface will not change except for the addition of color. Only terminals that match KS specifications are guaranteed to work with color. The database changes needed to support color RC/V are in the *mode_map* structure of the *cdopt* form.

Table 5-1. Terminal Displays

ODIN Display Field Type	Black and White	Color Terminal
Background text (text strings and name strings for a field)	White-Black	White-Black
No change input (underscored or default data)	White-Black	White-Black
Normal input (underscored before data entered)	White-Black Bold	Yellow-Black
Required input (underscored before data entered)	White-Black Bold, Underlined	Yellow-Blue Underlined
Acknowledgment message (temporary message in upper right corner)	Black-White Bold	White-Red Black-Green
Prompt message (command line—second from last—message)	White-Black	White-Black
Help message (help line—last line—message)	White-Black Underlined	White-Black
Error message (error line—first from last—message)	Black-White Bold	White-Red
Second error message (second error line—last line—of two-line error message)	Black-White Bold, Underlined	White-Red
Empty last line (last line with no text)	White-Black Underlined	White-Black Underlined

Black and White Terminal

Input Sequence	Set Sequence	Reset Sequence
(mp1-mp4)	(mp5-mp10)	(mp11-mp16)
ESC 'a'	ESC '[' '0' 'm'	ESC '[' '0' 'm'
ESC 'b'	ESC '[' '1' 'm'	ESC '[' '0' 'm'

Color Terminal

Input Sequence	Set Sequence	Reset Sequence
(mp1-mp4)	(mp5-mp10)	(mp11-mp16)
ESC 'a'	ESC '[' ';' '3' '7' 'm'	ESC '[' ';' '3' '7' 'm'
ESC 'b'	ESC '[' ';' '3' '3' 'm'	ESC '[' ';' '3' '7' 'm'

⇒ NOTE:

Use the character sequence `\e` instead of **ESC** in the actual input fields.

Example of an RC/V Session

The following information is an example of a typical RC/V transaction. It is intended to familiarize the RC/V user with the forms necessary to begin and end a transaction, the keystrokes necessary to complete a change, and to introduce the interactive messages that appear on the screen during an RC/V transaction. The user must determine the proper command line syntax, which is documented in Chapter 4, "Database Tools." The inputs that need to be specified by the user are shown in bold. Any user input must be followed by the <CR> key. For this example, it is assumed that the RC/V command has been entered with the intention of updating the information on a particular *classdef* form. The RC/V transaction is as follows:

1. The first display made by RC/V is:

```
UNIX RTR RCV (ODIN) - Data Entry
=====
Enter Form Name:
```

2. The form type necessary to begin a transaction is named *trbegin*, so enter **trbegin**.

```
UNIX RTR RCV (ODIN) - Data Entry
=====
Enter Form Name:trbegin
```

3. The *trbegin* form will be displayed.

```
Transaction Begin          trbegin      (1/1)          (Execute Only)
1.tr_name:TRBEGIN
```

The cursor will appear at the beginning of the input for the `tr_name` field. Press the <CR> key to use the default transaction begin name of TRBEGIN. The word `reading...` will appear in the upper right corner after the <CR> key is pressed.

4. At the bottom of the screen beneath the *trbegin* form, the following command line will be printed:

```
Enter Execute, Change, Substitute, Validate, or Print:
```

The *trbegin* form must be executed to initialize the RC/V transaction, so type `e` on the command line. The word `executing...` will appear in the upper right corner to inform the user that a transaction initialization is taking place.

5. When the *trbegin* form is successfully executed, the screen will refresh as shown in the following example and wait for the next form type to be requested.

```
UNIX RTR RCV (ODIN) - Data Entry
=====
Enter Form Name:
```

6. As shown in the example, now is the time to request the *classdef* form so that a particular *classdef* form instance may be updated.

```
UNIX RTR RCV (ODIN) - Data Entry
=====
Enter Form Name:classdef
```

7. After entering **classdef**, RC/V will prompt the user for the desired type of action to be done on the form as follows:

```
Enter Database Operation
I=Insert R=Review U=Update D=Delete
```

In this case an update is desired, so enter a **u**.

8. The *classdef* form will be displayed.

Class Definition Record		classdef	(1/1)
		(Recent Change and Verify)	
1.class_name: __	2.alarm: _		
3.device_list	logical_devname	8.rep_dest: _____	
	1) _____	11) _____	
	2) _____	12) _____	
	3) _____	13) _____	
	4) _____	14) _____	
	5) _____	15) _____	
	6) _____	16) _____	
	7) _____	17) _____	
	8) _____	18) _____	
	9) _____	19) _____	
	10) _____	20) _____	

9. Now assume that the *classdef* form for class 6 is to be updated. Enter a 6 in the key field *class_name* as follows:

Class Definition Record		classdef	(1/1)
		(Recent Change and Verify)	
1.class_name:6	2.alarm:_		
3.device_list	logical_devname	8.rep_dest:_____	
	1) _____	11) _____	
	2) _____	12) _____	
	3) _____	13) _____	
	4) _____	14) _____	
	5) _____	15) _____	
	6) _____	16) _____	
	7) _____	17) _____	
	8) _____	18) _____	
	9) _____	19) _____	
	10) _____	20) _____	

The word *reading...* will appear in the upper right corner to inform the user that the form is being read from the database.

10. After reading the form, it appears as follows:

Class Definition Record		classdef	(1/1)
		(Recent Change and Verify)	
1.class_name:6	2.alarm:y		
3.device_list	logical_devname	8.rep_dest:_____	
	1)ERLOG	11) _____	
	2)tty25	12) _____	
	3) _____	13) _____	
	4) _____	14) _____	
	5) _____	15) _____	
	6) _____	16) _____	
	7) _____	17) _____	
	8) _____	18) _____	
	9) _____	19) _____	
	10) _____	20) _____	

The command line of:

Enter Update, Change, Substitute, Validate, or Print: |
will be printed at the bottom of the screen. |

11. Now assume that the entry of "tty25" needs to be changed to "tty11."
Enter **c** on the command line. The command line will now prompt for the field number as follows:

Change field:

Enter a **3**

Change field: 3

RC/V will now prompt for row number.

Change field: 3 row:

Enter a **2**

Change field: 3 row: 2

The cursor will position at the "tty25" entry on the form.

12. Now type in **ttty11** to replace "tty25" with the new information. At this point, the updated form is displayed as follows:

Class Definition Record		classdef	(1/1)
			(Recent Change and Verify)
1.class_name:6	2.alarm:y		
3.device_list	logical_devname	8.rep_dest:_____	
	1)ERLOG	11)_____	
	2)tty11	12)_____	
	3)_____	13)_____	
	4)_____	14)_____	
	5)_____	15)_____	
	6)_____	16)_____	
	7)_____	17)_____	
	8)_____	18)_____	
	9)_____	19)_____	
	10)_____	20)_____	

- The cursor is currently at the bottom of the screen next to the "row" prompt. Press the <CR> key twice to clear the "row" and "Change field" prompts. The command line will once again be displayed.

```
Enter Update, Change, Substitute, Validate, or Print: |
```

- The user is now ready to make the update to the *classdef* form. Type **u** on the command line. The word *updating...* will appear in the upper right corner. When the update finishes, the cursor will reposition to the beginning of input for field No. 1 on the form.
- Type a < to return to the form selection prompt.

```
UNIX RTR RCV (ODIN) - Data Entry  
===== |  
Enter Form Name: |
```

The work in the transaction is complete, and a *trend* form must be executed to apply the changes.

- Enter **trend** as the form name.

```
UNIX RTR RCV (ODIN) - Data Entry  
===== |  
Enter Form Name:trend |
```

- The *trend* form will be displayed.

```
Transaction End                                trend          (1/1)          (Execute Only)  
  
1.tr_name:TREND  
2.dis_cf_checks:n  
3.apply_tr:y  
4.o_missing_links:n
```

The cursor will be positioned at the input for the *tr_name* field. Press the <CR> key four times to use the default values on the form.

18. The command line will once again be printed at the bottom of the screen.

```
Enter Execute, Change, Substitute, Validate, or Print:|
```

19. Enter an **e** to execute the *trend* form. At this time, validity checks are being run, and the changes specified in the transaction are being applied to the database. The word *executing...* will appear in the upper right corner to inform the user that work is being done.
20. When a successful trend completes, RC/V will display the form selection screen again.

```
UNIX RTR RCV (ODIN) - Data Entry  
=====|  
Enter Form Name:|
```

A <CR> key or < will end any further RC/V activity. If other changes need to be applied to the database, another RC/V transaction can be started at this time. Keep in mind that the interactive messages encountered when form insertion or deletion is desired are similar to this example.

After a SOAK period has passed, the change is written to disk via the *activate* form. Enter once again into recent change.

21. The first display made by RC/V is:

```
UNIX RTR RCV (ODIN) - Data Entry  
=====|  
Enter Form Name:|
```

22. The form type necessary to write changes to disk is named *activate*, so enter **activate**.

```
UNIX RTR RCV (ODIN) - Data Entry  
=====|  
Enter Form Name:activate|
```

23. The *activate* form will be displayed.

```
Copy Incore ECD to Root ECD          activate      (1/1)          (Execute Only)
1. copy_inc_to_disk: _
```

The cursor will appear at the beginning of the input for the `copy_inc_to_disk` field.

24. Enter a **y** to copy *incore* to disk.
25. Enter an **e** to execute the *activate* form. The word `executing...` will appear in the upper right corner to inform the user that work is being done. When *activate* is completed, `FORM EXECUTED` will flash once and the display will return to the data entry page. A `<CR>` key or `<` will end any further RC/V activity.

Contents

General Form Information	6-1
ECD Forms	6-5
■ <i>activate</i>	6-5
■ <i>aftopt</i>	6-6
■ <i>aphopt</i>	6-8
■ <i>aphuser</i>	6-10
■ <i>auderr</i>	6-15
■ <i>audhist</i>	6-20
■ <i>audinst</i>	6-22
■ <i>audrec</i>	6-24
■ <i>audstat</i>	6-31
■ <i>authdef</i>	6-34
■ <i>btihdev</i>	6-36
■ <i>btparm</i> [G2-R6.3, R6.5-R6.8]	6-38
■ <i>btparm</i> [R6.4 Only]	6-44
■ <i>btparm</i> [R21.1 & R21.3]	6-50
■ <i>btparm</i> [R21.5 & Later]	6-57
■ <i>bxslinfo</i>	6-63
■ <i>bxslloc</i>	6-64
■ <i>bxslrmt</i>	6-69
■ <i>cdopt</i>	6-71
■ <i>cftinfo</i>	6-74
■ <i>chopt</i>	6-75
■ <i>ciopt</i>	6-78
■ <i>classdef</i>	6-81
■ <i>cpblx3</i>	6-83
■ <i>cpphone</i>	6-92

Contents

■ <i>dbinfo</i>	6-99
■ <i>dciopt</i>	6-107
■ <i>device</i>	6-110
■ <i>dfcopt</i>	6-117
■ <i>dst</i>	6-121
■ <i>duopt</i>	6-126
■ <i>duoptdyn</i>	6-128
■ <i>eaiopt</i>	6-130
■ <i>ecdorg</i>	6-138
■ <i>errcount</i>	6-142
■ <i>errthld</i>	6-145
■ <i>ftaminopt</i>	6-148
■ <i>ftamopt</i>	6-150
■ <i>ftamuser</i>	6-153
■ <i>getty</i>	6-159
■ <i>gfsappl</i>	6-163
■ <i>gfscom</i>	6-174
■ <i>gfsinit</i>	6-194
■ <i>gfsresp</i>	6-203
■ <i>gfssec</i>	6-212
■ <i>klist</i>	6-221
■ <i>logdev</i>	6-223
■ <i>mdct</i>	6-230
■ <i>mhdopt</i>	6-235
■ <i>nphopt</i>	6-239
■ <i>objectid</i>	6-254
■ <i>pphopt</i>	6-256
■ <i>pphuser</i>	6-259
■ <i>rcvparams</i>	6-263
■ <i>runa</i>	6-265
■ <i>scsdbody</i>	6-267
■ <i>scsdmap</i>	6-270
■ <i>scsdopt</i>	6-274
■ <i>simcntl</i>	6-275
■ <i>sphopt</i>	6-279
■ <i>splrinfo</i>	6-281
■ <i>tphopt</i>	6-285
■ <i>tsapinfo</i>	6-289
■ <i>ttopt</i>	6-294
■ <i>ucb</i>	6-301

Contents

■ <i>ucbdyn</i>	6-324
■ <i>ularp</i>	6-328
■ <i>workhrs</i>	6-332
SG Forms	6-335
■ <i>fp</i>	6-335
■ <i>fpb</i>	6-339
■ <i>fs</i>	6-341
■ <i>fsb</i>	6-346
■ <i>klist</i>	6-351
■ <i>pack</i>	6-352
■ <i>process</i>	6-354
■ <i>rcvparams</i>	6-355
■ <i>sgen2</i>	6-357
■ <i>share</i>	6-360
ECD/SG Forms	6-361
■ <i>dbdump</i>	6-361
■ <i>dbload</i>	6-362
■ <i>prtlist</i>	6-363
■ <i>trabort</i>	6-364
■ <i>trbegin</i>	6-365
■ <i>trend</i>	6-366

General Form Information

This chapter illustrates each of the low-level mode database forms used to review or modify the Equipment Configuration Database (ECD) and the System Generation (SG) database and serves as a guide in filling out these forms.

The forms can be divided into the following categories:

- ECD Contains forms that describe hardware relationships, options, equipage, and locations on the 3B20D/3B21D computer.
- SGEN Contains forms that describe software parameters and processes required at bootstrap.
- 3BGEN Contains forms that alter or access the database.
- MKDSK Contains forms that describe the disk file system.

Table 6-1 lists the forms associated with the ECD and the SG database, their categories, and their operation modes.

Operation modes include the following:

- insert (i) Inserts a new form into the database.
- review (r) Allows you to review a specified form in the database.
- delete (d) Deletes a form from the database.
- update (u) Changes specified fields on an existing database form.
- execute (e) Allows you to run a database utility.

Table 6-1. ECD and SG Forms

Form	Operation Modes				Name
Equipment Configuration Database Category 1					
aftopt	i	r	d	u	Application file transfer option record
aphopt	i	r	d	u	Application protocol handler option record
aphuser	i	r		u	Application protocol handler user record
auderr	i	r	d	u	Audit error recovery
audhist		r			Audit instance information
audinst	i	r	d	u	Audit instance information
audrec	i	r	d	u	Audit record
audstat		r			Audit information
authdef	i	r	d	u	Authorization definition record
btihdev	i	r	d	u	British Telecom I/O interface handler option
btparm	i	r		u	Boot time parameters
bxslinfo	i	r		u	X.25 session layer information
bxslloc	i	r	d	u	X.25 session layer local users information
bxslrmt	i	r	d	u	X.25 session layer remote users information
cdopt	i	r	d	u	Control display options
cftinfo	i	r		u	Craft information record
chopt	i	r	d	u	IOP controller option
ciopt	i	r	d	u	Craft interface option
classdef	i	r	d	u	Class definition record
cpblx3	i	r	d	u	X.25 data link option
cpphnone	i	r	d	u	X.25 data link option
dciopt	i	r	d	u	DCI operation options
device	i	r	d	u	Device definition record
dfcopt	i	r	d	u	DFC operation options
dst	i	r	d	u	Daylight saving time option
duopt	i	r	d	u	Direct user interface option
duoptdyn		r			Direct user interface dynamic
eaiopt	i	r	d	u	Emergency action interface option
ecdorg	i	r		u	ECD origination record
errcount		r			Error count
errthld	i	r	d	u	Error threshold
getty	i	r	d	u	Get shell record
logdev	i	r	d	u	Logical device information
mdct	i	r	d	u	Minor device chain table
mhdopt	i	r	d	u	MHD operation options
nphopt	i	r	d	u	Network protocol handler option
rcvparams				e	RC/V parameters information
runa	i	r	d	u	Remote user network information
scsdbody	i	r	d	u	Scanner and signal distributor
scsdmap		r			Scanner and signal distributor map
scsdopt	i	r	d	u	Scanner and signal distributor option
simcntl	i	r		u	System integrity monitor control record

Table 6-1. ECD and SG Forms (Contd)

Form	Operation Modes					Name
Equipment Configuration Database Category 1						
sphopt	i	r		u		Session protocol handler option record
splinfo	i	r		u		Spooler information record
tpthopt	i	r	d	u		Transport protocol handler option block record
tsapinfo	i	r	d	u		Transport service access point information record
ttopt	i	r	d	u		TTY terminal option
ucb	i	r	d	u		Unit control block
ucbdyn		r				Unit control block dynamic
ularp	i	r	d	u		ULARP ECD record
workhrs	i	r		u		Working hours for exchange information
SGEN (System Generation) Category 2						
process	i	r	d	u		Processes
sgen2	i	r	d	u		System generation 2 (boot processes)
share	i	r	d			Share segment
3BGEN (Recent Change and Verify) Category 3						
activate					e	Activate a transaction
dbdump					e	Dump
dbinfo					e	Information
dbload					e	Load
klist					e	Key list
prtlist					e	Print list
rcvparms					e	RC/V parameter information
trabort					e	Transaction abort
trbegin					e	Transaction begin
trend					e	Transaction end
MKDSK (Make Disk) Category 4						
fp	i	r	d	u		File partition
fpb	i	r	d	u		File partition body
fs	i	r	d	u		File system
fsb	i	r	d	u		File system body
pack	i	r	d	u		Pack (disk pack information)
i=insert		r=review		d=delete		u=update
						e=execute

The following mnemonics are used throughout the field descriptions. A definition of each is given.

ESCAPE – An escape sequence beginning with a backslash (\)

HEX – A hexadecimal number beginning with 0x; example, (0xa4)

HEX2D – A hexadecimal number beginning with x; example, (xb8)

QCHAR – A character in single quotation marks; example, ('c')

MAXINT – The largest positive integer that a 32-bit integer will hold:
2147483647.

null – Fields for which this is legal may be left blank.

⇒ NOTE:

Each field description gives the field number, field name, and serial text message Recent Change and Verify (RC/V) keyword to be used. The following is an example of the pack field on a *ucb* form.

35.location.pack (*LOCATION.PACK)

Field number 35 may be used to reference the pack data field. As an alternative, the field name location.pack may be used. The keyword to use during a serial text message RC/V session would be LOCATION.PACK; the asterisk (*) is not part of the field name but indicates that the SET keyword must be used.

The numbering convention for the forms containing bit fields is the far right bit field designated as bit 0 while the bit to the far left is the high order bit.

ECD Forms

activate

When this form is executed, the incore copy of the ECD is copied over the disk copy from which the system was booted.

Form Layout

```
Copy Incore ECD to Root ECD          activate      (1/1)          (Execute Only)
1.copy_inc_to_disk:_
```

Definition of Fields

1.copy_inc_to_disk: (COPYINCTODISK) Required

Copy incore to disk. This is a key field. For processing to occur, "yes" must be entered.

Legal value: yes

Default value: yes

aftopt

This form is used to enter option information for the Application File Transfer (AFT). It consists of buffer sizes and timers for this feature.

Form Layout

```
Application File Transfer Option Record          aftopt          (1/1)
                                                (Recent Change and Verify)
1.aftopt_name:_____ 2.dsk_bufsize:_____
3.rft_bufsize:_____ 4.open_timer:_____
5.data_timer:_____ 6.data_end_timer:_____
```

Definition of Fields

1 .aftopt_name: (CAFTOPTNAME) Required

Option block name. This is a key field.

Legal values: must be a logical device name: */dev/* followed by a filename

2 .dsk_bufsize: (DSKBUFSIZE)

The size of buffer for reading data from disk.

Legal values: 2048, 4096 ,8192, 16384, 32768, 65536, or 131072

Default value: 131072

3. rft_bufsize: (RFTBUFSIZE)

The size of buffer for sending data to the Application Protocol Handler (APH).

Legal values: 2048, 4096 ,8192, 16384, 32768, 65536, or 131072

Default value: 131072

4. open_timer: (OPENTIMER)

The maximum time between open requests.

Legal values: 30 – 300

Default value: 300

5. data_timer: (DATATIMER)

The maximum allowable time for a write request of data to APH.

Legal values: 30 – 300

Default value: 300

6. data_end_timer: (DATAENDTIMER)

The maximum allowable time between a data end request and deselection from the remote side.

Legal values: 30 – 600

Default value: 300

Form Checks

```
dsk_bufsize >= rft_bufsize  
error: "dsk_bufsize must be greater than or equal to rft_bufsize";
```

aphopt

⇒ NOTE:

Before an *aphopt* record can be inserted into the database, an *sphopt* record must already exist in the database.

This form enters option information for the International Telegraph and Telephone Consultative Committee (CCITT) application protocol handler.

Form Layout

Application Protocol Handler	aphopt	(1/1)
Option Record		(Recent Change and Verify)
1.aphoptname:_____	2.application_type:_____	
3.max_ac:___	4.local_ssap_sel:_____	
5.connect_timer:___	6.disconnect_timer:___	

Definition of Fields

1.aphoptname: (APHOPTNAME) Required

Application form identifier. This is a key field.

Legal values: 1 – 8 alphanumeric characters

2.application_type: (APPLICATIONTYPE) Required

Application process type identifier.

Legal values: polling, terminal, spont-output, basic

3.max_ac: (MAXAC) Required

Maximum number of application connections.

Legal values: 0 – 200, decimal

Default value: 0

4.local_ssap_sel: (LOCALSSAPSEL) Required

Local session service access point (SSAP) selector.

Legal values: 1 – 16 printable characters

5.connect_timer: (CONNECTTIMER) Required

Application connection response timer. This timer is used during connection establishment.

Legal values: 120 – 600

Default value: 300

6.disconnect_timer: (DISCONNECTTIMER) Required

Application connection disconnect timer. This timer is used during the connection release to wait for the data token to be able to release the connection.

Legal values: 120 – 600

Default value: 300

aphuser

This form enters information for the International Telegraph and Telephone Consultative Committee (CCITT) application protocol handler. The information is used to map between special device files and application connection identifiers.

Form Layout

aphuser		(1/2)
Application Protocol Handler User Record		(Recent Change and Verify)
1.logdev_name:_____	2.loc_connect_id:_____	
3.rmt_connect_id:_____	4.aphoptname:_____	
5.initiator:_____	6.multiple_opens:_____	
7.rmt_ssap_sel:_____		
8.rmt_tsap_addr:_____		
9.process_name:_____		

aphuser		(2/2)
10.full_half:_____	11.sending_side:_____	
12.eom:_____	13.exp_data:_____	
14.qos_priority:_____		
15.qos_max_out:_____	16.qos_avg_out:_____	
17.qos_max_in:_____	18.qos_avg_in:_____	

Definition of Fields

1. logdev_name: (LOGDEVNAME) Required

Logical device name. This is a key field in conjunction with the loc_connect_id: field for Recent Change and Verify (RC/V) access.

Legal values: 1 – 19 printable characters consisting of /dev/ followed by a filename

2. loc_connect_id: (LOCCONNECTID) Required

The calling connection identifier for locally established application connections and the called connection identifier for remotely established application connections. This is a key field in conjunction with the logical_name field for RC/V access. This is a key field for program access via Low-Level Access (LLA).

Legal values: 1 – 16 printable characters

3. rmt_connect_id: (RMTCONNECTID) Required

The called connection identifier for locally established application connections and the calling connection identifier for remotely established application connections.

Legal values: 1 – 16 printable characters

4. aphoptname: (APHOPTNAME) Required

Name of associated *aphopt* record. This record name will be entered and the Record Identifier (RID) will be stored.

Legal values: 1 – 8 alphanumeric characters

5. initiator: (INITIATOR) Required

Indication of whether or not the local side is the initiator of the application connection establishment. A value of "a" indicates an archive library user.

Legal values: Local = y, Remote = n, Archive Library User = a

6. multiple_opens: (MULTIPLEOPENS) Required

Indication of whether or not multiple processes will open the same logical name for use of the same application connection. This field is ignored if the value of the initiator field is "a."

Legal values: y or n

Default value: n

7. rmt_ssap_sel: (RMTSSAPSEL) Required

Called Session Service Access Point (SSAP) identifier.

Legal values: 1 – 16 printable characters

8. rmt_tsap_addr: (RMTSAPADDR) Required

Called Transport Service Access Point (TSAP) identifier.

Legal values: 1 – 44 decimal characters

9. process_name: (PROCESS)

Process name. Full pathname of the process.

Legal values: 1 – 63 printable characters consisting of “/” followed by the full pathname of the process.

10. full_half: (FULHALF) Required

Full or half-duplex connection.

Legal values: full, half

Default value: full

11. sending_side: (SENDINGSIDE)

Initial sender. Indication of which side will first transmit normal data. This field is ignored if full duplex is selected.

Legal values: local, remote, or null

12. eom: (EOM) Required

End of message indication. Declaration of whether or not the option will be used.

Legal values: y, n

Default value: n

13. exp_data: (EXPDATA) Required

Expedited data indicator. Indicates if the option will be used. This field is applicable only for archive library users.

Legal values: y, n

Default value: n

14. qos_priority: (QOSPRIORITY) Required

Quality of service application connection priority.

Legal values: 0 – 64

Default value: 64

15. qos_max_out: (QOSMAXOUT)

Quality of service maximum output value. This value should not exceed the baud rate of the Data Terminal Equipment (DTE).

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000, null

Default value: 75

16. qos_avg_out: (QOSAVGOUT)

Quality of service average acceptable output value. This value should not exceed the baud rate of the DTE(s).

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000, null

Default value: 75

17. qos_max_in: (QOSMAXIN)

Quality of service input value.

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000, null

Default value: 75

18. qos_avg_in: (QOSSAVGIN)

Quality of service average acceptable input value. This value should not exceed the baud rate of the DTE(s).

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000, null

Default value: 75

Form Checks

```
qos_max_out >= qos_avg_out
  error: "qos_max_out must be greater than or equal to qos_avg_out";

qos_max_in >= qos_avg_in
  error: "qos_max_in must be greater than or equal to qos_avg_in";
```

auderr

This form enters information for recovery from one of the six error categories in an audit. The error is identified by audit name, member number, and category number. One of these forms must be entered for each error category that applies.

Form Layout

```

                                auderr      (1/1)
Audit Error Recovery Record      (Recent Change and Verify)
1.audname:_____  2.memno:___  3.category:_  4.threshold:_____
5.time_interval:_____  6.alarm_lvl:_  7.rtn_corr:_____  8.dmd_corr:_____
9.error_recovery
 10.stage  11.action  12.rdata1  13.rdata2  14.raudname  15.rmemno
   -      -         -         -         -         -
   -      -         -         -         -         -
   -      -         -         -         -         -
   -      -         -         -         -         -
   -      -         -         -         -         -
   -      -         -         -         -         -

```

Definition of Fields

1. audname: (AUDNAME) Required

Audit class name. This is a key field.

Legal values: alphanumeric characters

2. memno: (MEMNO) Required

Audit member number. This is a key field.

Legal values: 1 – 255

3. category: (CATEGORY) Required

The error category number. This is a key field.

Legal values: 0 – 5

4. threshold: (THRESHOLD)

The error threshold for the error category.

Legal values: 0 – 32767, null

5. time_interval: (TIMEINTERVAL)

The error counting interval in seconds.

Legal values: 0 – 604800 seconds, null

6. alarm_lvl: (ALARMLVL) Required

The alarm level that should be used on the output message when errors are reported in this category.

Legal values: A (action), I (information), M (minor alarm), R (major alarm), C (critical alarm)

7. rtn_corr: (RTNCORR) Required

Routine error correction limit.

Legal values: 0 – 32767

8. dmd_corr: (DMDCORR) Required

Demand error correction limit.

Legal values: 0 – 32767

9. error_recovery: (ERRORRECOVERY)

Each recovery category has six possible recovery actions that constitute the escalation strategy that System Integrity Monitor (SIM) will use when the error threshold for this category has been exceeded. The following information specifies each stage of recovery escalation.

10. error_recovery.stage: (*ERRORRECOVERY.STAGE[ROW])

Stage of recovery action.

Legal values: 1 – 6, null

11. error_recovery.action: (*ERRORRECOVERY.ACTION[ROW])

Recovery action code. The action that should be taken to recover from the error. Some possibilities are run a demand audit, request a Phase 1, request a Phase 2, kill a class of processes, and so forth.

Legal values: DFLT PHASE SAMAUD
 NOACT RPTAIM TRMPRC
 PANDMP RUNAUD null

12. error_recovery.rdata1: (*ERRORRECOVERY.RDATA1[ROW])

A piece of data that complements the recovery action. For example, if the recovery action specifies PHASE, the rdata1 field contains the phase level to be taken. If the recovery action specifies TRMPRC, the rdata1 field contains the utility id of the process to be terminated.

Legal values: 0 – 2047, HEX: 0x0 – 0x7ff, null

13. error_recovery.rdata2: (*ERRORRECOVERY.RDATA2[ROW])

A piece of data that complements the recovery action. If the recovery action specifies TRMPRC, the rdata2 field contains the process class number to be terminated. If the recovery action specifies PHASE, the rdata2 field contains a word to be inserted into the Processor Recovery Message (PRM).

Legal values: 0 – 2147483647, HEX: 0x0 – 0x7fffffff, null

14. error_recovery.raudname: (*ERRORRECOVERY.RAUDNAME[ROW])

Audit class name. If the recovery action specifies that an audit should be demanded, these fields (14 and 15) should contain the audit name and member number. If appropriate, SIM will assume that the instance name for the demand audit will be the same as the one used by the audit that initiated the demand.

Legal values: alphanumeric characters, null

15. error_recovery.rmemno: (*ERRORRECOVERY.RMEMNO[ROW])

Audit member number.

Legal values: 1 – 255, null

Form Checks

```
for every error_recovery element:

    do;

        case (action);
            when {"RUNAUD"}: do;
                rdata1 must_not_be known
                error: "rdata1 must not be set when action is 'RUNAUD'";
                rdata2 must_not_be known
                error: "rdata2 must not be set when action is 'RUNAUD'";
            end;

            when {"SAMAUD"}: do;
                rdata1 must_not_be known
                error: "rdata1 must not be set when action is 'SAMAUD'";
                rdata2 must_not_be known
                error: "rdata2 must not be set when action is 'SAMAUD'";
            end;

            when {"PHASE"}: do;
                rdata1 must_be known
                error: "rdata1 must be set when action is 'PHASE'";
                rdata must_be_in {regex("[0-3]"),
                                   regex("^0x[0-3]")}
                error: "rdata1 must be in the range 0-3
                        when action is 'PHASE'";

                raudname must_not_be known;
                error: "raudname cannot be set when action is 'PHASE'";
                rmemno must_not_be known
                error: "rmemno cannot be set when action is 'PHASE'";
            end;

            when {"TRMPRC"}: do;
                raudname must_not_be known;
                error: "raudname cannot be set when action is 'TRMPRC'";
                rmemno must_not_be known
                error: "rmemno cannot be set when action is 'TRMPRC'";
            end;
        end;
    end;
```

```
when {unknown, "RPTAIM", "PANDMP", "DFLT", NOACT}: do;
  raudname must_not_be known;
    error: "raudname can only be set when action is
      'SAMAUD' or 'RUNAUD'";
  rmemno must_not_be known
    error: "rmemno can only be set when action is
      'SAMAUD' or 'RUNAUD'";
  rdata1 must_not_be known
    error: "rdata1 can only be set when action is
      'PHASE' or 'TRMPCR'";
  rdata2 must_not_be known
    error: "rdata2 can only be set when action is
      'PHASE' or 'TRMPCR'";
end;
end;
end;
```

audhist

This is a review-only form that contains information about a particular audit instance.

Form Layout

```

                                audhist      (1/1)
Audit Instance History                                (Review Only)
1.audname:_____  2.memno:___  3.instname:_____
4.sch_grp:_____  5.inhibit_state:_____  6.block:_____  7.running:_
8.error_counts
category    9.timed_count  10.long_term_count
0           _____  _____
1           _____  _____  11.total_found:_____
2           _____  _____
3           _____  _____  12.total_corrected:_____
4           _____  _____
5           _____  _____
```

Definition of Fields

1. audname: (AUDNAME) Required

Name of the audit. This is a key field.

Legal values: alphanumeric characters

2. memno: (MEMNO) Required

Member number of the audit. This is a key field.

Legal values: 1 – 255

3. instname: (INSTNAME) Optional

Instance name. This is a key field.

Legal values: printable characters, null

4. sch_grp:

Scheduling group of the audit.

Legal values: kernel, supv

5. inhibit_state:

Inhibit state of this instance.

Legal values: ALLOWED, AUTOINH, MANINH, REININH

6. block:

If nonzero, this instance of the audit is blocked and this entry is the process number of the process that blocked it.

Legal values: 0 – MAXINT

7. running:

If this is y, this instance of the audit is running.

Legal values: y, n

8. error_counts:

9. timed_count:

The error counts for the current error threshold interval.

Legal values: 0 – MAXINT

10. long_term_count:

The error counts since the last system bootstrap.

Legal values: 0 – MAXINT

11. total_found:

The total errors reported by the audit for this instance since the last system bootstrap.

Legal values: 0 – MAXINT

12. total_corrected:

The total errors corrected by the audit for this instance since the last system bootstrap.

Legal values: 0 – MAXINT

audinst

This form is used to enter information about one of four possible instances of an audit. It is identified by audit name, member number, and instance name. One of these forms must be entered for each instance of the audit. At least one *audinst* form must be entered for each *audrec* form that is entered.

Form Layout

Audit Instance Record		audinst	(1/1)	
			(Recent Change and Verify)	
1.audname:_____	2.memno:_____	3.instname:_____		
4.freq:_____	5.t_sun :_	12.t_0000:_____	20.t_0800:_____	28.t_1600:_____
	6.t_mon :_	13.t_0100:_____	21.t_0900:_____	29.t_1700:_____
Timed	7.t_tues:_____	14.t_0200:_____	22.t_1000:_____	30.t_1800:_____
Audit	8.t_wed :_	15.t_0300:_____	23.t_1100:_____	31.t_1900:_____
Schedule	9.t_thur:_____	16.t_0400:_____	24.t_1200:_____	32.t_2000:_____
	10.t_fri :_	17.t_0500:_____	25.t_1300:_____	33.t_2100:_____
	11.t_sat :_	18.t_0600:_____	26.t_1400:_____	34.t_2200:_____
		19.t_0700:_____	27.t_1500:_____	35.t_2300:_____

Definition of Fields

1. audname: (AUDNAME) Required

Audit class name. This is a key field.

Legal values: alphanumeric characters

2. memno: (MEMNO) Required

Audit member number. This is a key field.

Legal values: 1 – 255

3. instname: (INSTNAME) Optional

Instance name. This is a key field. The instance name can contain up to 19 characters. If the audit will never have more than a single instance, this field may be left null. Otherwise, the full instance name should be given. Blanks are not allowed in this field.

Legal values: printable characters, null

4. freq: (FREQ)

The routine scheduling frequency group of the audit. "A" through "H" refer to frequency group audits. A "T" frequency means that the audit is a timed audit. If the audit is a timed audit, the time parameters must be set.

Legal values: A, B, C, D, E, F, G, H, T, null

5. t_sun: (TSUN) – 11.t_sat: (TSAT)

Day-of-the-week fields for timed audits.

Legal values: y, n

Default value: n

12. t_0000: (T0000) – 35.t_2300: (T2300)

Hour-of-the-day fields for timed audits.

Legal values: y, n

Default value: n

Form Checks

```
form_checks:
  instname must_not_be blank
  error: "instname must not be blank";
```

audrec

This form enters instance-independent information about an audit. It is identified by audit name and audit member number.

Form Layout

audrec			(1/2)
Audit Record			(Recent Change and Verify)
1.audname:_____	2.memno:___	3.prname:_____	
4.timeout_int:_____	5.maxseg:_____	6.utility_id:_____	
7.segmented:_	8.appaud:_	9.correcting:_	
10.kill_type:_____	11.sched_grp:_____	12.num_er_repts:___	
13.fixdpnum:_	14.prc_num:_____		

audrec			(2/2)
Modes of Execution			
15.rtn_mode_permit:_	16.rtn_dp_type:_____	17.rtn_dp_data:_____	
18.dmd_mode_permit:_	19.dmd_dp_type:_____	20.dmd_dp_data:_____	
21.man_mode_permit:_	22.man_dp_type:_____	23.man_dp_data:_____	
24.sft_mode_permit:_	25.sft_dp_type:_____	26.sft_dp_data:_____	

Definition of Fields

1. audname: (AUDNAME) Required

Audit class name. This is a key field. It can be up to six characters.

Legal values: alphanumeric characters

2. memno: (MEMNO) Required

The member number of an audit. This is a key field.

Legal values: 1 – 255

3. pname: (PRNAME)

Transient audit process name which has to be 14 characters long. The audit processes are placed in a common directory (*/audprc* for RTR).

Legal values: printable characters, null

4. timeout_int: (TIMEOUTINT) Required

Time-out interval (seconds). The maximum elapsed time in seconds [not Central Processing Unit (CPU) time] that a nonsegmented audit or a segment of a segmented audit is permitted to run. This time should take into account the percentage of system time that is usually available at the execution level of the audit.

For a nonsegmented audit, the value of time-out should be based on the number of data units to be audited and, therefore, may vary from one application to another. For an audit with multiple instances, the number of data units in the largest instance should be used in the time-out calculation. As a precautionary measure, a fudge factor should be included to compensate for periods of intense CPU utilization.

If the time-out limit is reached, SIM terminates the audit.

Legal values: 0 – 32767

5. maxseg: (MAXSEG) Required

Maximum segments that a segmented audit is permitted to run. If this value is exceeded, SIM terminates the audit.

Legal values: 0 – 32767

6. utility_id: (UTILITYID) Required

The utility identification (utilid) from the process loader (LDP) spec file of the audit process. This will be used to identify audits that do not have fixed process ids.

Legal values: 0 – 2047, 0x0 – 0x7FF

7. segmented: (SEGMENTED)

Segmented audit. If this flag is set, the audit is segmented.

Legal values: y, n

Default value: n

8. appaud: (APPAUD)

Application audit on. If this flag is set, the audit is an application audit.

Legal values: y, n

Default value: n

9. correcting: (CORRECTING)

Correcting audit.

Legal values: y, n

Default value: y

10. kill_type: (KILLTYPE) Required

Audit process type.

Legal values:	nonkill	A nonkillable audit process
	qkill	A quasikillable audit process
	transient	A transient audit process

11. sched_grp: (SCHEDGRP) Required

The scheduling group of the audit.

Legal values:	kernel	A kernel level audit. This group includes special kernel process audits.
	supv	A supervisor level audit. Because the same algorithm schedules supervisor and <i>UNIX</i> ® system level audits, all <i>UNIX</i> system level audits should specify supv as their scheduling group.

12. num_er_repts: (NUMERREPTS) Required

Maximum number of raw data reports.

Legal values: 0 – 255

13. fixdpnum: (FIXDPNUM)

A flag to indicate that the audit has a fixed process number. If this field is set, then prc_num must be provided.

Legal values: y, n

Default value: n

14. prc_num : (PRCNUM)

Audit process number. This is the process id of the audit process if the audit is a nonkillable audit process with a fixed process id. This must be entered if fixdpnum is set.

Legal values: 0 – MAXINT, HEX, null

15. rtn_mode_permit: (RTNMODEPERMIT) Required

Routine mode permitted. If set, routine audit execution is permitted.

Legal values: y, n

16. rtn_dp_type: (RTNDPTYPE)

Routine mode dispatch type.

Legal values:

event	The audit will be dispatched by an event.
evmsg	The audit will be dispatched by an emergency mechanism. The emergency dispatching is not correctly implemented.
msg	The audit will be dispatched by a message.
null	

17. rtn_dp_data: (RTNDPDATA)

The data that dispatches routine audits. If the audit is message dispatched, it will be the message type. If the audit is event dispatched, it will be the event number. The value for emergency dispatched audits has not been determined.

Legal values: 0 – MAXINT, HEX, null

Default value: 0

18. dmd_mode_permit: (DMDMODEPERMIT) Required

Demand mode permitted. This serves the same purpose for demand audits as the rtn_mode_permit field serves for routine audits.

Legal values: y, n

19. dmd_dp_type: (DMDDPTYPE)

Demand dispatch type. This serves the same purpose for demand audits as the rtn_dp_type field serves for routine audits.

Legal values: msg, event, evmsg, null

20. dmd_dp_data: (DMDDPDATA)

Demand dispatching data. This serves the same purpose for demand audits as the rtn_dp_data field serves for routine audits.

Legal values: 0 – MAXINT, HEX, null

Default value: 0

21. man_mode_permit: (MANMODEPERMIT) Required

Manual mode permitted. This serves the same purpose for manual audits as the rtn_mode_permit field serves for routine audits.

Legal values: y, n

22. man_dp_type: (MANDPTYPE)

Manual dispatch type. This serves the same purpose for manual audits as the rtn_dp_type field serves for routine audits.

Legal values: msg, event, evmsg, null

23. man_dp_data: (MANDPDATA)

Manual dispatching data. This serves the same purpose for manual audits as the rtn_dp_data field serves for routine audits.

Legal values: 0 – MAXINT, HEX, null

Default value: 0

24. sft_mode_permit: (SFTMODEPERMIT) Required

External mode on. This serves the same purpose for external audits as the rtn_mode_permit field serves for routine audits.

Legal values: y, n

25. sft_dp_type: (SFTDPTYPE)

External dispatch type. This serves the same purpose for external audits as the rtn_dp_type field serves for routine audits.

Legal values: msg, event, evmsg, null

26. sft_dp_data: (SFTDPDATA)

External dispatching data. This serves the same purpose for external audits as the rtn_dp_data field serves for routine audits.

Legal values: 0 – MAXINT, HEX, null

Default value: 0

Form Checks

```
if fixdpnum is "y"
  then prc_num must_not_be unknown
    error: "prc_num must be known when fixdpnum is 'y'";

if kill_type is "transient"
  then prname must_not_be unknown
    error: "prname must be known when kill_type is transient";

if sched_grp is "supv"
  then do;
    segmented must_be "n"
      error: "segmented must be n";
    rtn_dp_type must_not_be "event"
      error: "rtn_dp_type must not be event";
    rtn_dp_type must_not_be "emer";
      error: "rtn_dp_type must not be emer";
    dmd_dp_type must_not_be "event"
      error: "dmd_dp_type must not be event";
    dmd_dp_type must_not be "emer";
      error: "dmd_dp_type must not be emer";
    man_dp_type must_not_be "event";
      error: "man_dp_type must not be event";
    man_dp_type must_not_be "emer"
      error: "man_dp_type must not be emer";
    sft_dp_type must_not_be "event"
      error: "sft_dp_type must not be event";
    sft_dp_type must_not_be "emer"
      error: "sft_dp_type must not be emer";
  end;
```

```
if rtn_mode_permit is "y"
  then do:
    rtn_dp_type must_not_be unknown
    error: "rtn_dp_type must be known when rtn_mode_permit is on";
    rtn_dp_data must_not_be unknown
    error: "rtn_dp_data must be known when rtn_mode_permit is on";
  end;

if dmd_mode_permit is "y"
  then do:
    dmd_dp_type must_not_be unknown
    error: "dmd_dp_type must be known when dmd_mode_permit is on";
    dmd_dp_data must_not_be unknown
    error: "dmd_dp_data must be known when dmd_mode_permit is on";
  end;

if man_mode_permit is "y"
  then do:
    man_dp_type must_not_be unknown
    error: "man_dp_type must be known when man_mode_permit is on";
    man_dp_data must_not_be unknown
    error: "man_dp_data must be known when man_mode_permit is on";
  end;

if sft_mode_permit is "y"
  then do:
    sft_dp_type must_not_be unknown
    error: "sft_dp_type must be known when sft_mode_permit is on";
    sft_dp_data must_not_be unknown
    error: "sft_dp_data must be known when sft_mode_permit is on";
  end;
```

audstat

This is a review-only form that contains information about a particular audit.

Form Layout

```

                                audstat      (1/1)
Audit Status                                (Review Only)
1.audname:____ 2.memno:____ 3.num_instances:____ 4.inhibit_state:____
EXECUTION STATUS      RECOVERY ACTION      LONG-TERM HISTORY
5.running_mode:____ 6.rcvy_category:___ 7.completions:_____
8.dispatched:_ 9.rcvy_stage:___ 10.disp_failures:_____
11.stopped:_ 12.rcvy_step:___ 13.intf_failures:_____
14.segments_run:____ 15.rcvy_status:_____ 16.fault_aborts:_____
17.running_inst:_____ 18.aud_timeouts:_____
```

Definition of Fields

1. audname: (AUDNAME) Required

Name of audit. This is a key field.

Legal values: alphanumeric characters

2. memno: (MEMNO) Required

Member number of audit. This is a key field.

Legal values: 1 – 255

3. num_instances:

Number of instances present for this audit.

Legal values: 0 – 255

4. inhibit_state:

Inhibit state of audit.

Legal values: ALLOWED, AUTOINH, MANINH, REININH

5. running_mode:

Mode in which audit is running.

Legal values: RTN, DMD, MAN, SFT, null

6. rcvy_category:

Category of current or most recent recovery action initiated by this audit.

Legal values: 0 – 15

7. completions:

Number of times audit completed since last bootstrap.

Legal values: 0 – MAXINT

8. dispatched:

If this is y, audit is currently dispatched.

Legal values: y, n

9. rcvy_stage:

Stage of current or most recent recovery action initiated by this audit.

Legal values: 0 – 15

10. disp_failures:

Number of dispatched failures that have occurred for this audit since last bootstrap.

Legal values: 0 – MAXINT

11. stopped: (STOPPED)

If this is y, a request has been made to stop the audit.

Legal values: y, n

12. rcvy_step:

The number of recovery steps initiated for that recovery stage.

Legal values: 0 – 15

13. intf_failures:

Number of interface failures that have occurred for this audit since last bootstrap.

Legal values: 0 – MAXINT

14. segments_run:

Number of segments that have been executed by a segmented audit.

Legal values: 0 – 32767, null

15. rcvy_status:

Current status of recovery action taken for this audit.

Legal values: COMPLETED INPROGRESS WAITRESULT
 DELAYED NORCVY WAITSTART

16. fault_aborts:

Number of times audit has fault aborted or died while correcting since last bootstrap.

Legal values: 0 – MAXINT

17. running_inst:

Instance name of currently running instance.

Legal values: printable characters, null

18. aud_timeouts:

Number of times audit has been timed out since last bootstrap.

Legal values: 0 – MAXINT

authdef

This form defines the different command groups.

Form Layout

```

                                     authdef      (1/1)
Authority Definition Record          (Recent Change and Verify)
1.comgr_name:_____
2.comgr_number:___
3.denied_request:                   6.allowed_request:
   4.alarm_level:___                7.alarm_level:___
   5.log_flag:___                   8.log_flag:___
```

Definition of Fields

1. comgr_name: (COMGRNAME) Required

Command group name. This is the key field for the form. It must be, at most, eight alphanumeric characters long, uppercase, and the first character must be alpha.

2. comgr_number: (COMGRNUMBER)

Command group number.

Legal values: 0 – 127

3. denied_request: (DENIEDREQUEST)

4. denied_request.alarm_level: (*DENIEDREQUEST.ALARMLEVEL)

Alarm level for a denied input request.

Legal values:

- CR – critical alarm
- MJ – major alarm
- MN – minor alarm
- IF – information alarm
- NO – no alarm

Default value: NO

5.denied_request.log_flag: (*DENIEDREQUEST.LOGFLAG)

Command logging for a denied input request.

Legal values: y, n

Default value: y

6. allowed_request: (ALLOWEDREQUEST)

7. allowed_request.alarm_level: (*ALLOWEDREQUEST.ALARMLEVEL)

Alarm level for an allowed input request.

Legal values:

CR – critical alarm

MJ – major alarm

MN – minor alarm

IF – information alarm

NO – no alarm

Default value: NO

8. allowed_request.log_flag: (*ALLOWEDREQUEST.LOGFLAG)

Command logging for an allowed input request.

Legal values: y, n

Default value: y

btihdev

This form is used to enter option information for the standard Input/Output (I/O) Interface Handler for British Telecom.

Form Layout

```
btihdev (1/1)
British Telecom Interface Handler Device Record (Recent Change and Verify)
1.crtl_dev:_____ 2.tsap_id:_____
3.a_timer:_          4.i_timer:_____
5.d_timer:_
6.output_devname:_
(1)_____
(2)_____
(3)_____
(4)_____
(5)_____
(6)_____
```

Definition of Fields

1. crtl_dev: (CTRLDEV) Required

The name of logical control device. This is a key field.

Legal values: must begin with /dev/ and at least a one-character filename.

2. tsap_id: (TSAPID) Required

The name of Transport Service Access Point (TSAP) selector which is owned by the user of control device.

Legal values: 1 – 9999

3. a_timer: (ATIMER) Required

Acknowledgment timer for checkpoint.

Legal values: 1 – 300

Default value: 300

4. i_timer: (ITIMER) Required

The inactivity timer is started when a connect request has come in, a reply negative has been sent, a final checkpoint acknowledgment is received, or a close has been done on the output device.

Legal values: 1 – 300

Default value: 300

5. d_timer: (DTIMER) Required

This is the value of the timer which is started when a checkpoint acknowledgment (not a final acknowledgment) is received.

Legal values: 1 – 300

Default value: 300

6. output_devname: (OUTPUTDEVNAME)

(1) – (6) dev_name:

The names of logical output devices associated with the control device.

Legal values: must begin with /dev/ and at least a one-character filename

Form Checks

```
count {select dev_name from output_dev}>0
  error: "a minimum of one output device must be specified";
{select dev_name from output_dev} must be unique}
  error:"dev_name values must be unique";
```

btparm [G2-R6.3, R6.5-R6.8]

This form specifies most of the boot time parameters for the *UNIX*® Real-Time Reliable (RTR) operating system.

⇒ NOTE:

This form applies only to *UNIX* RTR operating system Generic 2 through Release 6.3 and Release 6.5 through Release 6.8.

Form Layout

```
                                btparm      (1/2)
Boot Time Parameters           (Recent Change and Verify)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.
```

```
1.btparmname:_____  2.messages:_____  3.ports:_____
4.processes:_____    5.segments:_____  6.page_tables:_____
7.pages:_____        8.kernel_stack:_____  9.istack:_____
10.initintv:_____    11.maxlev0:_____  12.maxlev1:_____
13.maxlev2:_____     14.maxlev3:_____  15.swapmin:_____
16.swapstart:_____   17.swapblks:_____  18.ecd_segs:_____
```

```
                                btparm      (2/2)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.
```

```
19.l_pas_size:_____  20.h_pas_size:_____  21.l_pas_strt:_____
22.h_pas_strt:_____  23.part_boundary:_____  24.pas_dump:_____
25.s_stick:_____
```

Definition of Fields

1. btparmname: (BTPARMNAME) Required

This is the key for the *btparm* form. Since there will be only one *btparm* form per database, this field always expects the value "btparm."

Legal value: BTPARM

Default value: BTPARM

2. messages: (MESSAGES) Required

The amount of space the operating system should reserve for buffering messages (64-byte message buffers).

Legal values: 128 – 2048

Default value: 512

3. ports: (PORTS) Required

Maximum number of system process ports.

Legal values: 50 – 32768

Default value: 50

4. processes: (PROCESSES) Required

Maximum number of existing processes.

Legal values: 64 – 512

Default value: 64

5. segments: (SEGMENTS) Required

Maximum number of possible segments.

Legal values: 384 – 4096

Default value: 512

6. page_tables: (PAGETABLES) Required

Maximum number of page tables.

Legal values: 384 – 2048

Default value: 512

7. pages: (PAGES) Required

Maximum number of memory pages with which the operating system may be configured.

Legal values: 2048 – 32768

Default value: 2048

8. kernel_stack: (KERNELSTACK) Required

Size of kernel stack in bytes.

Legal values: 6144 – 131072

Default value: 6144

9. istack: (ISTACK) Required

Size of interrupt stack in bytes.

Legal values: 1200 – 4096

Default value: 1200

10. initintv: (INITINTV) Required

Initialization interval.

Legal values: 60 – 1800

Default value: 100

11. maxlev0: (MAXLEV0) Required

Initialization level 0.

Legal values: 0 – 15

Default value: 4

12. maxlev1: (MAXLEV1) Required

Initialization level 1.

Legal values: 0 – 15

Default value: 4

13. maxlev2: (MAXLEV2) Required

Initialization level 2.

Legal values: 0 – 15

Default value: 4

14. maxlev3: (MAXLEV3) Required

Initialization level 3.

Legal values: 0 – 15

Default value: 4

15. swapmin: (SWAPMIN) Required

Swap pages needed by biggest supervisor or user process.

Legal values: 400 – 4096

Default value: 400

16. swapstart: (SWAPSTART) Required

Starting block number of swap space.

Legal values: 0 – 100000

Default value: 0

17. swapblks: (SWAPBLKS) Required

Size of total swap space in blocks.

Legal values: 5000 – 100000

Default value: 5000

18. ecd_segs: (ECDSEGS) Required

ECD size in segments.

Legal values: 4 – 16

Default value: 4

19. l_pas_size: (LPASSIZE) Required

Size of lower protected application segment in pages.

Legal values: 0 – 2048

Default value: 0

20. h_pas_size: (HPASSIZE) Required

Size of higher protected application segment in pages.

Legal values: 0 – 6144

Default value: 0

⇒ NOTE:

Although the legal values range up to 6144, the actual maximum values to be input for this field are based on the following hardware configurations:

16 MB Memory 2048

32 MB Memory 4096

64 MB Memory 6144

21. l_pas_strt: (LPASSTRT) Required

Starting segment index of lower protected application segment.

Legal values: 0 – 31

Default value: 0

22. h_pas_strt: (HPASSTRT) Required

Starting segment index of higher protected application segment.

Legal values: 64 – 510

Default value: 64

23. part_boundary: (PARTBOUNDARY) Required

Partition boundary segment index.

Legal value: 64

Default value: 64

24. pas_dump: (PASDUMP)

Protected application segment included in dump?

Legal values: y, n

Default value: n

25. s_stick: (SSTICK) Required

Scheduling interval.

Legal values: 300 – 2500 (1000 = 1 second)

Default value: 500

Form Checks

```
segments must_not_be_less_than page_tables
  error: "segments must be greater than or equal to
  page_tables";

if h_pas_size != 0
  then h_pas_strt >= (60 + ecd_segs)
    error: "high pas and ecd segments overlay one another";

/* Convert h_pas_strt to pages and check that the
  start page plus the h_pas_size does not exceed the
  maximum segment size in pages (128 * 64) */
(h_pas_strt * 64) + h_pas_size <= 8192
  error: "high pas page specification too large in relation to
  starting high pas segment";

/* Convert l_pas_strt to pages and check that the
  start page plus the l_pas_size size does not exceed the
  maximum segment size in pages (32 * 64) */
(l_pas_strt * 64) + l_pas_size <= 2048
  error: "low pas page specification too large in relation
  to starting low pas segment";
```

btparm [R6.4 Only]

This form specifies most of the boot time parameters for the *UNIX*® Real-Time Reliable (RTR) operating system.

⇒ NOTE:

This form applies only to *UNIX* RTR operating system Release 6.4.

Form Layout

```

                                     btparm      (1/2)
Boot Time Parameters                 (Recent Change and Verify)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.

 1.btparmname:_____  2.messages:_____  3.ports:_____
 4.processes:_____   5.segments:_____  6.page_tables:_____
 7.pages:_____       8.kernel_stack:_____  9.istack:_____
10.initintv:_____   11.maxlev0:_____  12.maxlev1:_____
13.maxlev2:_____    14.maxlev3:_____  15.swapmin:_____
16.swapstart:_____  17.swapblks:_____  18.ecd_segs:___
```

```

                                     btparm      (2/2)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.

19.l_pas_size:_____  20.h_pas_size:_____  21.l_pas_strt:___
22.h_pas_strt:_____  23.part_boundary:___  24.pas_dump:___
25.s_stick:_____    26.fds_activate:_____
```

Definition of Fields

1. btparmname: (BTPARMNAME) Required

This is the key for the *btparm* form. Since there will be only one *btparm* form per database, this field always expects the value "btparm."

Legal value: BTPARM

Default value: BTPARM

2. messages: (MESSAGES) Required

The amount of space the operating system should reserve for buffering messages (64-byte message buffers).

Legal values: 128 – 2048

Default value: 512

3. ports: (PORTS) Required

Maximum number of system process ports.

Legal values: 50 – 32768

Default value: 50

4. processes: (PROCESSES) Required

Maximum number of existing processes.

Legal values: 64 – 512

Default value: 64

5. segments: (SEGMENTS) Required

Maximum number of possible segments.

Legal values: 384 – 4096

Default value: 512

6. page_tables: (PAGETABLES) Required

Maximum number of page tables.

Legal values: 384 – 2048

Default value: 512

7. pages: (PAGES) Required

Maximum number of memory pages with which the operating system may be configured.

Legal values: 2048 – 32768

Default value: 2048

8. kernel_stack: (KERNELSTACK) Required

Size of kernel stack in bytes.

Legal values: 6144 – 131072

Default value: 6144

9. istack: (ISTACK) Required

Size of interrupt stack in bytes.

Legal values: 1200 – 4096

Default value: 1200

10. initintv: (INITINTV) Required

Initialization interval.

Legal values: 60 – 1800

Default value: 100

11. maxlev0: (MAXLEV0) Required

Initialization level 0.

Legal values: 0 – 15

Default value: 4

12. maxlev1: (MAXLEV1) Required

Initialization level 1.

Legal values: 0 – 15

Default value: 4

13. maxlev2: (MAXLEV2) Required

Initialization level 2.

Legal values: 0 – 15

Default value: 4

14. maxlev3: (MAXLEV3) Required

Initialization level 3.

Legal values: 0 – 15

Default value: 4

15. swapmin: (SWAPMIN) Required

Swap pages needed by biggest supervisor or user process.

Legal values: 400 – 4096

Default value: 400

16. swapstart: (SWAPSTART) Required

Starting block number of swap space.

Legal values: 0 – 100000

Default value: 0

17. swapblks: (SWAPBLKS) Required

Size of total swap space in blocks.

Legal values: 5000 – 100000

Default value: 5000

18. ecd_segs: (ECDSEGS) Required

ECD size in segments.

Legal values: 4 – 16

Default value: 4

19. l_pas_size: (LPASSIZE) Required

Size of lower protected application segment in pages.

Legal values: 0 – 2048

Default value: 0

20. h_pas_size: (HPASSIZE) Required

Size of higher protected application segment in pages.

Legal values: 0 – 4096

Default value: 0

⇒ NOTE:

Although the legal values range up to 6144, the actual maximum values to be input for this field are based on the following hardware configurations:

16 MB Memory	2048
32 MB Memory	4096
64 MB Memory	6144

21. l_pas_strt: (LPASSTRT) Required

Starting segment index of lower protected application segment.

Legal values: 0 – 31

Default value: 0

22. h_pas_strt: (HPASSTRT) Required

Starting segment index of higher protected application segment.

Legal values: 64 – 510

Default value: 64

23. part_boundary: (PARTBOUNDARY) Required

Partition boundary segment index.

Legal value: 64

Default value: 64

24. pas_dump: (PASDUMP)

Protected application segment included in dump?

Legal values: y, n

Default value: n

25. s_stick: (SSTICK) Required

Scheduling interval.

Legal values: 300 – 2500 (1000 = 1 second)

Default value: 500

26. fds_activate: (FDSACTIVATE) Required

This field activates or deactivates the Force Disk Selection Feature. This feature also allows selection of application disks to be restored during bootstrap. Refer to 254-303-106, *System Maintenance Manual*, for a description of this feature.

Legal value: y, n

Default value: n

Form Checks

```
segments must_not_be_less_than page_tables
  error: "segments must be greater than or equal to
  page_tables";

if h_pas_size != 0
  then h_pas_strt >= (60 + ecd_segs)
    error: "high pas and ecd segments overlay one another";

/* Convert h_pas_strt to pages and check that the
  start page plus the h_pas_size does not exceed the
  maximum segment size in pages (128 * 64) */
(h_pas_strt * 64) + h_pas_size <= 8192
  error: "high pas page specification too large in relation to
  starting high pas segment";

/* Convert l_pas_strt to pages and check that the
  start page plus the l_pas_size size does not exceed the
  maximum segment size in pages (32 * 64) */
(l_pas_strt * 64) + l_pas_size <= 2048
  error: "low pas page specification too large in relation
  to starting low pas segment";
```

btparm [R21.1 & R21.3]

This form specifies most of the boot time parameters for the *UNIX*® Real-Time Reliable (RTR) operating system.

⇒ NOTE:

This form applies only to *UNIX* RTR Operating System Release 21.1 and Release 21.3.

Form Layout

```

                                     btparm      (1/2)
Boot Time Parameters                (Recent Change and Verify)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.

1.btparmname:_____  2.messages:_____  3.ports:_____
4.processes:_____    5.segments:_____  6.page_tables:_____
7.pages:_____        8.kernel_stack:_____  9.istack:_____
10.initlintv:_____   11.maxlev0:_____   12.maxlev1:_____
13.maxlev2:_____     14.maxlev3:_____   15.swapmin:_____
16.swapstart:_____   17.swapblks:_____   18.ecd_segs:_____
```

```

                                     btparm      (2/2)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.

19.l_pas_size:_____  20.h_pas_size:_____  21.l_pas_strt:_____
22.h_pas_strt:_____   23.part_boundary:_____  24.pas_dump:_____
25.s_stick:_____     26.init2intv:_____
```

Definition of Fields

1. btparmname: (BTPARMNAME) Required

This is the key for the *btparm* form. Since there will be only one *btparm* form per database, this field always expects the value "btparm."

Legal value: BTPARM

Default value: BTPARM

2. messages: (MESSAGES) Required

The amount of space the operating system should reserve for buffering messages (64-byte message buffers).

Legal values: 128 – 2048

Default value: 512

3. ports: (PORTS) Required

Maximum number of system process ports.

Legal values: 50 – 32768

Default value: 50

4. processes: (PROCESSES) Required

Maximum number of existing processes.

Legal values: 64 – 512

Default value: 64

5. segments: (SEGMENTS) Required

Maximum number of possible segments.

Legal values: 384 – 4096

Default value: 512

6. page_tables: (PAGETABLES) Required

Maximum number of page tables.

Legal values: 384 – 2048

Default value: 512

7. pages: (PAGES) Required

Maximum number of memory pages with which the operating system may be configured.

Legal values: 2048 – 65536

Default value: 2048

8. kernel_stack: (KERNELSTACK) Required

Size of kernel stack in bytes.

Legal values: 6144 – 131072

Default value: 6144

9. istack: (ISTACK) Required

Size of interrupt stack in bytes.

Legal values: 1200 – 4096

Default value: 1200

10. init1intv: (INIT1INTV) Required

This is the initialization interval which corresponds to bootstrap recoveries. (Values entered are interpreted as seconds.)

Legal values: 60 – 1800

Default value: 100

11. maxlev0: (MAXLEV0) Required

Initialization level 0.

Legal values: 0 – 15

Default value: 4

12. maxlev1: (MAXLEV1) Required

Initialization level 1.

Legal values: 0 – 15

Default value: 4

13. maxlev2: (MAXLEV2) Required

Initialization level 2.

Legal values: 0 – 15

Default value: 4

14. maxlev3: (MAXLEV3) Required

Initialization level 3.

Legal values: 0 – 15

Default value: 4

15. swapmin: (SWAPMIN) Required

Swap pages needed by biggest supervisor or user process.

Legal values: 400 – 4096

Default value: 400

16. swapstart: (SWAPSTART) Required

Starting block number of swap space.

Legal values: 0 – 100000

Default value: 0

17. swapblks: (SWAPBLKS) Required

Size of total swap space in blocks.

Legal values: 5000 – 100000

Default value: 5000

18. ecd_segs: (ECDSEGS) Required

ECD size in segments.

Legal values: 4 – 16

Default value: 4

19. l_pas_size: (LPASSIZE) Required

Size of lower protected application segment in pages.

Legal values: 0 – 2048

Default value: 0

20. h_pas_size: (HPASSIZE) Required

Size of higher protected application segment in pages.

Legal values: 0 – 6144

Default value: 0

⇒ NOTE:

Although the legal values range up to 6144, the actual maximum values to be input for this field are based on the following hardware configurations:

16 MB Memory	2048
32 MB Memory	4096
64 MB Memory	6144

21. l_pas_strt: (LPASSTRT) Required

Starting segment index of lower protected application segment.

Legal values: 0 – 31

Default value: 0

22. h_pas_strt: (HPASSTRT) Required

Starting segment index of higher protected application segment.

Legal values: 64 – 510

Default value: 64

23. part_boundary: (PARTBOUNDARY) Required

Partition boundary segment index.

Legal value: 64

Default value: 64

24. pas_dump: (PASDUMP)

Protected application segment included in dump?

Legal values: y, n

Default value: n

25. s_stick: (SSTICK) Required

Scheduling interval.

Legal values: 300 – 2500 (1000 = 1 second)

Default value: 500

26. init2intv: (INIT2INTV) Required

This field represents a second initialization interval for level 1 recoveries. (Values entered will be interpreted as seconds). This value should never exceed the value of "init1intv."

Legal value: 60 – 300

Default value: 100

Form Checks

```
segments must_not_be_less_than page_tables
  error: "segments must be greater than or equal to
  page_tables";

if h_pas_size != 0
  then h_pas_strt >= (60 + ecd_segs)
    error: "high pas and ecd segments overlay one another";

/* Convert h_pas_strt to pages and check that the
  start page plus the h_pas_size does not exceed the
  maximum segment size in pages (128 * 64) */
(h_pas_strt * 64) + h_pas_size <= 8192
  error: "high pas page specification too large in relation to
  starting high pas segment";
```

```
/* Convert l_pas_strt to pages and check that the
   start page plus the l_pas_size size does not exceed the
   maximum segment size in pages (32 * 64) */

(l_pas_strt * 64) + l_pas_size <= 2048
   error: "low pas page specification too large in relation
         to starting low pas segment";

init2intv < or = initlintv
   error: "init2intv cannot be greater than initlintv";
```

btparm [R21.5 & Later]

This form specifies most of the boot time parameters for the *UNIX*® Real-Time Reliable (RTR) operating system.

⇒ NOTE:

This form applies only to *UNIX* RTR Operating System Release 21.5 and later.

Form Layout

```

                                     btparm      (1/2)
Boot Time Parameters                (Recent Change and Verify)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.

1.btparmname:_____  2.messages:_____  3.ports:_____
4.processes:_____    5.segments:_____  6.page_tables:_____
7.pages:_____        8.kernel_stack:_____  9.istack:_____
10.initlintv:_____  11.maxlev0:_____  12.maxlev1:_____
13.maxlev2:_____    14.maxlev3:_____  15.swapmin:_____
16.swapstart:_____  17.swapblks:_____  18.ecd_segs:_____
```

```

                                     btparm      (2/2)
THESE ARE CRITICAL SYSTEM PARAMETERS!! CHANGE WITH EXTREME CAUTION.

19.l_pas_size:_____  20.h_pas_size:_____  21.l_pas_strt:_____
22.h_pas_strt:_____  23.part_boundary:_____  24.pas_dump:_____
25.s_stick:_____    26.init2intv:_____  27.fds_activate:_____
```

Definition of Fields

1. btparmname: (BTPARMNAME) Required

This is the key for the *btparm* form. Since there will be only one *btparm* form per database, this field always expects the value “btparm.”

Legal value: BTPARM

Default value: BTPARM

2. messages: (MESSAGES) Required

The amount of space the operating system should reserve for buffering messages (64-byte message buffers).

Legal values: 128 – 2048

Default value: 512

3. ports: (PORTS) Required

Maximum number of system process ports.

Legal values: 50 – 32768

Default value: 50

4. processes: (PROCESSES) Required

Maximum number of existing processes.

Legal values: 64 – 512

Default value: 64

5. segments: (SEGMENTS) Required

Maximum number of possible segments.

Legal values: 384 – 4096

Default value: 512

6. page_tables: (PAGETABLES) Required

Maximum number of page tables.

Legal values: 384 – 2048

Default value: 512

7. pages: (PAGES) Required

Maximum number of physical memory pages with which the operating system may be configured.

Legal values: 2048 – 65536

Default value: 2048

8. kernel_stack: (KERNELSTACK) Required

Size of kernel stack in bytes.

Legal values: 6144 – 131072

Default value: 6144

9. istack: (ISTACK) Required

Size of interrupt stack in bytes.

Legal values: 1200 – 4096

Default value: 1200

10. init1intv: (INIT1INTV) Required

This is the initialization interval which corresponds to bootstrap recoveries. (Values entered are interpreted as seconds.)

Legal values: 60 – 1800

Default value: 100

11. maxlev0: (MAXLEV0) Required

Initialization level 0.

Legal values: 0 – 15

Default value: 4

12. maxlev1: (MAXLEV1) Required

Initialization level 1.

Legal values: 0 – 15

Default value: 4

13. maxlev2: (MAXLEV2) Required

Initialization level 2.

Legal values: 0 – 15

Default value: 4

14. maxlev3: (MAXLEV3) Required

Initialization level 3.

Legal values: 0 – 15

Default value: 4

15. swapmin: (SWAPMIN) Required

Swap pages needed by biggest supervisor or user process.

Legal values: 400 – 4096

Default value: 400

16. swapstart: (SWAPSTART) Required

Starting block number of swap space.

Legal values: 0 – 100000

Default value: 0

17. swapblks: (SWAPBLKS) Required

Size of total swap space in blocks.

Legal values: 5000 – 100000

Default value: 5000

⇒ NOTE:

For releases 21.31 and later: the maximum legal value was increased to 250000 from 100000.

18. ecd_segs: (ECDSEGS) Required

ECD size in segments.

Legal values: 4 – 16

Default value: 4

19. l_pas_size: (LPASSIZE) Required

Size of lower protected application segment in pages.

Legal values: 0 – 2048

Default value: 0

20. h_pas_size: (HPASSIZE) Required

Size of higher protected application segment in pages.

Legal values: 0 – 12288

Default value: 0

⇒ NOTE:

Protected Application Segments (PAS) are reserved portions of physical memory. The physical memory size as specified by field 7, pages, may need to be increased accordingly with an increase in PAS size. Field 6, page_tables, may also need to be adjusted accordingly.

21. l_pas_strt: (LPASSTR) Required

Starting segment index of lower protected application segment.

Legal values: 0 – 31

Default value: 0

22. h_pas_strt: (HPASSTR) Required

Starting segment index of higher protected application segment.

Legal values: 64 – 510

Default value: 64

23. part_boundary: (PARTBOUNDARY) Required

Partition boundary segment index.

Legal value: 64

Default value: 64

24. pas_dump: (PASDUMP)

Protected application segment included in dump?

Legal values: y, n

Default value: n

25. s_stick: (SSTICK) Required

Scheduling interval.

Legal values: 300 – 2500 (1000 = 1 second)

Default value: 500

26. init2intv: (INIT2INTV) Required

This field represents a second initialization interval for level 1 recoveries. (Values entered will be interpreted as seconds). This value should never exceed the value of "init1intv."

Legal value: 60 – 300

Default value: 100

27. fds_activate: (FDSACTIVATE) Required

This field activates or deactivates the Force Disk Selection Feature. This feature allows selection of application disks to be restored during bootstrap. Refer to 254-303-106, *System Maintenance Manual*, for a description of this feature.

Legal value: y, n

Default value: n

Form Checks

```
segments must_not_be_less_than page_tables
  error: "segments must be greater than or equal to
  page_tables";

if h_pas_size != 0
  then h_pas_strt >= (60 + ecd_segs)
    error: "high pas and ecd segments overlay one another";

/* Convert h_pas_strt to pages and check that the
  start page plus the h_pas_size size does not exceed the
  maximum segment size in pages (511 * 64) */
(h_pas_strt * 64) + h_pas_size <= 32704
  error: "high pas page specification too large in relation to
  starting high pas segment";

/* Convert l_pas_strt to pages and check that the
  start page plus the l_pas_size size does not exceed the
  maximum segment size in pages (32 * 64) */
(l_pas_strt * 64) + l_pas_size <= 2048
  error: "low pas page specification too large in relation
  to starting low pas segment";

init2intv < or = init1intv
  error: "init2intv cannot be greater than init1intv";
```

bxslinfo

The X.25 session layer information record.

Form Layout

Session Layer Information	bxslinfo	(1/1) (Recent Change and Verify)
1.bxslinfofname: _____	2.maxsess: _____	

Definition of Fields

1. bxslinfofname: (BXSLINFOFNAME) Required

bxslinfofname is the key for the *bxslinfo* form. Since there can be only one *bxslinfo* form in a database, this field always contains "bxslinfo."

Legal value: BXSLINFO

Default value: BXSLINFO

2 .maxsess: (MAXSESS) Required

Maximum sessions. This field is the maximum number of established sessions.

Legal values: 0 – 128

Default value: 0

bxslloc

The X.25 session layer local users information.

Form Layout

```
bxslloc      (1/2)
Session Layer Local Users List      (Recent Change and Verify)

1.loc_name: _____

2.loc_addr   loc1   loc2   loc3   loc4   loc5   loc6   loc7   loc8
             -     -     -     -     -     -     -     -
             loc9  loc10  loc11  loc12  loc13  loc14  loc15  loc16
             -     -     -     -     -     -     -     -

19.origin_type: ___ 20.sess_dialog: ___ 21.prs_type: _____ 22.resync: _
23.recov: _         24.scm_resp_time: ___ 25.spdu_resp_time: ___
26.sresm_resp_time: ___ 27.srecm_wait_time: ___ 28.srecm_resp_time: ___
```

```
bxslloc      (2/2)

29.spdu_count: _ 30.sresm_count: _ 31.srecm_count: _ 32.scm_count: _
33.procnm: _____
34.maxproc: ___
```

Definition of Fields

1. loc_name: (LOCNAME) Required

Session user name. A 16-character name uniquely identifying this session. This is the key field to the *bxslloc* form.

Legal values: printable characters

2. loc_addr.loc1 — loc_addr.loc16: (*LOCADDR.LOC1 — LOCADDR.LOC16)

Session address. This is a 1 to 16 array that specifies the session address. An entry in the first location is required. A blank field in the middle of the address will translate to a 0.

Legal values: hexadecimal digits, null

19 .origin_type: (ORIGINTYPE) Required

Origin type. This gets translated to a valid X.25 session layer protocol origin type which is a parameter in the Session Count Message (SCM).

Legal values: MSL, SL, TOP

Default value: SL

20. sess_dialog: (SESSDIALOG) Required

Dialogue type. This gets translated to a valid X.25 session layer protocol dialogue type which is a parameter in the SCM.

Legal values: 1IN, 1OUT, 2ALT, 2SIM

Default value: 2SIM

21. prs_type: (PRSTYPE) Required

Presentation type. This gets translated to a valid X.25 session layer protocol presentation type which is a parameter in the SCM.

Legal values: ASCII, BINARY, EBCDIC

Default value: ASCII

22. resync: (RESYNC) Required

Resynchronization. This permits the following two items:

1. The sending session layer to retransmit Session Protocol Data Units (SPDUs) which have not been acknowledged by the receiving session layer.
2. The receiving session layer to request the transmission of SPDUs starting from the one that is missing from the sequence it has received.

Legal values: y, n

Default value: n

23. recov: (RECOV) Required

Recovery. This allows the initiating session layer to reconnect the existing session over a new transport connection in the event of the failure of original transport connection.

Legal values: y, n

Default value: n

24. scm_resp_time: (SCMRESPTIME) Required

SCM response timer. This timer is used during the establishment phase of all sessions.

Legal values: 10 – 600 seconds

Default value: 260

25. spdu_resp_time: (SPDURESPTIME) Required

SPDU response timer. This is used only for a session using the resynchronization option.

Legal values: 10 – 600 seconds

Default value: 260

26. sresm_resp_time: (SRESMRESPTIME) Required

Session Resynchronization Message (SRESM) response timer. This timer is used only for a session using the resynchronization option.

Legal values: 10 – 600 seconds

Default value: 120

27. srecm_wait_time: (SRECMWAITTIME) Required

Wait for SRECM timer. This timer is used only for a session using the recovery option.

Legal values: 10 – 600 seconds

Default value: 300

28. srecm_resp_time: (SRECMRESPTIME) Required

SRECM response timer. This timer is used only for a session using the recovery option.

Legal values: 10 – 600 seconds

Default value: 300

29. spdu_count: (SPDUCOUNT) Required

SPDU attempt count. This is a count of the number of times the SPDU will be retransmitted.

Legal values: 0 – 10 attempts

Default value: 2

30. sresm_count: (SRESMCOUNT) Required

SRESM attempt count. This is a count of the number of times the session resynchronization message will be retransmitted.

Legal values: 1 – 10 attempts

Default value: 2

31. srecm_count: (SRECMCOUNT) Required

SRECM attempt count. This is a count of the number of times the session reconnect message will be retransmitted.

Legal values: 1 – 10 attempts

Default value: 1

32. scm_count: (SCMCOUNT) Required

SCM attempt count. This is a count of the number of times the session connect message will be retransmitted.

Legal values: 1 – 10 attempts

Default value: 1

33. procnm: (PROCNM)

Process name. Full pathname of process name.

Legal values: printable characters, null

34. maxproc: (MAXPROC) Required

Maximum process. This is the maximum number of invocations of the process.

Legal values: 0 – 128

Default value: 0

Form Checks

```
if procnm is known
  then maxproc must_not_equal 0
    error: "maxproc cannot be 0 when procnm is set";
```

bxslrmt

The X.25 session layer remote users information.

Form Layout

```
bxslrmt (1/1)
Session Layer Remote Users List (Recent Change and Verify)

1.rmt_name: _____

2.rmt_addr  rmt1  rmt2  rmt3  rmt4  rmt5  rmt6  rmt7  rmt8
            -    -    -    -    -    -    -    -
            rmt9 rmt10 rmt11 rmt12 rmt13 rmt14 rmt15 rmt16
            -    -    -    -    -    -    -    -

19.remote_dte  addr_type  address
  1)  ___  _____
  2)  ___  _____
  3)  ___  _____
  4)  ___  _____
```

Definition of Fields

1. rmt_name: (RMTNAME) Required

Session user name. A 16-character name uniquely identifying this session. This is the key field to the *bxslrmt* form.

Legal values: printable characters

2. rmt_addr.rmt1 — rmt_addr.rmt16: (*RMTADDR.RMT1 — RMTADDR.RMT16)

Session address. This is a 1 to 16 array that specifies the session address. An entry in the first location is required. A blank field in the middle of the address will translate to 0.

Legal values: hexadecimal digits, null

19.remote_dte: (REMOTEDTE)

A minimum of one remote address must be specified.

remote_dte.addr_type: (*REMOTEDTE.ADDRTYPE[ROW])

Address type. This is the type of remote address.

Legal values: PVC, SVC, null

remote_dte.address: (*REMOTEDTE.ADDRESS[ROW])

Address. This is the remote network address.

Legal values: decimal digits, null

Form Checks

```
for every remote_dte element:
```

```
do;
```

```
  if addr_type is known
```

```
    then address must_not_be unknown
```

```
      error: "address must be set when addr_type is set";
```

```
  if address is known
```

```
    then addr_type must_not_be unknown
```

```
      error: "addr_type must be set when address is set";
```

```
end;
```

```
count {select address from remote_dte} > 0
```

```
  error: "a minimum of one dte address must be specified";
```

cdopt

This form represents the terminal control structure for a control display terminal. It contains mapping and physical characteristics information needed for Cathode Ray Tube (CRT) control.

Form Layout

```

                                     cdopt          (1/4)
Terminal Control Structure For /dev/cdx      (Recent Change and Verify)

1.option_name: _____

2.direct_map
dmap1 dmap2 dmap3 dmap4 dmap5 dmap6 dmap7 dmap8 dmap9 dmap10
1)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
2)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
3)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
4)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
5)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
6)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
7)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
8)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
9)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
```

```

                                     cdopt          (2/4)
2.direct_map(cont)
dmap1 dmap2 dmap3 dmap4 dmap5 dmap6 dmap7 dmap8 dmap9 dmap10
10)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
11)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
12)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
13)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
14)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
15)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
16)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

25.mode_map
mp1 mp2 mp3 mp4 mp5 mp6 mp7 mp8 mp9 mp10 mp11 mp12 mp13 mp14 mp15 mp16
1)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
2)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
3)  ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
```

```

                                     cdopt          (3/4)
25.mode_map(cont)
  mp1 mp2 mp3 mp4 mp5 mp6 mp7 mp8 mp9 mp10 mp11 mp12 mp13 mp14 mp15 mp16
4)  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
5)  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
6)  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
7)  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
8)  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
9)  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
10) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
11) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
12) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
13) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
14) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
15) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
16) _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _

```

```

                                     cdopt          (4/4)
60.init_seq:
  is1 is2 is3 is4 is5 is6 is7 is8 is9 is10 is11 is12 is13 is14 is15 is16
  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _

77.scroll_seq:
  ss1 ss2 ss3 ss4 ss5 ss6 ss7 ss8 ss9 ss10 ss11 ss12 ss13 ss14 ss15 ss16
  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _

94.io_area:   ioa1   ioa2   ioa3   ioa4
              _  _  _  _

99.input_map: im1 im2 im3 im4 im5 im6 im7 im8 im9
              _  _  _  _  _  _  _  _  _

109.output_map: om1 om2 om3 om4 om5 om6 om7 om8 om9
                _  _  _  _  _  _  _  _  _

```

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. An eight-character name uniquely identifying this cdopt option block. This is the key field.

Legal values: alphanumeric characters

2. direct_map.dmap1 — direct_map.dmap10:
(*DIRECTMAP.DMAP1[ROW] — DIRECTMAP.DMAP10[ROW])

Direct map. This field translates cursor positioning input sequences to terminal output sequences. The direct map field is divided into two parts: input map (virtual) and output map (terminal). Sixteen possible cursor positioning input sequences exist.

Legal values: 0 – 255, HEX2D, QCHAR, ESCAPE, null

25. mode_map.mp1 — mode_map.mp16:
(*MODEMAP.MP1[ROW] — MODEMAP.MP16[ROW])

Mode map.

Legal values: 0 – 255, HEX2D, QCHAR, ESCAPE, null

60. init_seq.is1 — init_seq.is16: (*INITSEQ.IS1 — INITSEQ.IS16)

Initialize sequence.

Legal values: 0 – 255, HEX2D, QCHAR, ESCAPE, null

77. scroll_seq.ss1 — scroll_seq.ss16: (*SCROLLSEQ.SS1 — SCROLLSEQ.SS16)

Scrolling sequence.

Legal values: 0 – 255, HEX2D, QCHAR, ESCAPE, null

94. io_area.ioa1 — io_area.io4: (*IOAREA.IOA1 — IOAREA.IOA4)

Input/Output (I/O) area. This field has the format $nmi j$ where n gives the first line of the display; m gives the last line of the display; and i, j gives the starting location of the current input line.

Legal values: 0 – 255, HEX, null

99. input_map.im1 — input_map.im9: (*INPUTMAP.IM1 — INPUTMAP.IM9)

Input conversion map.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

109. output_map.om1 — output_map.om9:
(*OUTPUTMAP.OM1 — OUTPUTMAP.OM9)

Output conversion map.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

cftinfo

This form contains various information about the tunable parameters used by the Display Administration Process (DAP).

Form Layout

ECD Craft Information Record		cftinfo	(1/1)
		(Recent Change and Verify)	
1.cftinfofname:_____		2.num_devs:_____	
3.num_pages:_____		4.scc_avail:___	

Definition of Fields

1.cftinfofname: (CFTINFONAME) Required

This is a key field. Because there is only one *cftinfo* form in a database, this field always contains "cftinfo."

Legal value: CFTINFO

Default value: CFTINFO

2.num_devs: (NUMDEVS)

Number of active control display devices.

Legal values: 2 – 16

Default value: 16

3.num_pages: (Numpages)

Number of pages to be initialized.

Legal values: 10 – 256

Default value: 256

4.scc_available: (SCCAVAILABLE)

To specify whether the ttyz is a Switching Control Center (SCC) device.

Legal values: y or n

Default value: y

chopt

This form gives the options for the Input/Output Processor (IOP) controller.

For *UNIX*® Real-Time Reliable (RTR) Operating System Release 21.3 only, the *chopt* record is used as the Dual Serial Channel (DSCH) Computer Interconnect (DCI) option record. Every DCI *ucb* record must have a DCI option record. Only three fields are used by the DCI driver. Field 4 (*poll*) will be used as the minimum polling interval. Field 5 (*time_out*) will be used as the maximum polling interval. Field 6 (*picconf*) will be used as the auto restore interval.

Form Layout

Option Block For IOP Controller	chopt	(1/1) (Recent Change and Verify)
1.option_name: _____	2.iop_id: ____	
3.handler_id: ____	4.poll: _____	
5.time_out: _____	6.picconf: _____	
7.pcoos: _____	8.comoos: _____	
9.intvps: ____	10.dgnflg: _____	

Definition of Fields

1.option_name: (OPTIONNAME) Required

Option block name. This is the key field.

Legal values: alphanumeric characters

2.iop_id: (IOPID)

IOP identification (id). Id of the IOP (not used, set to zero).

Legal values: 0 – 255

Default value: 0

3.handler_id: (HANDLERID)

Handler id. This field gives the handler id (not used, set to zero).

Legal values: See this field on the *ucb* form.

4.poll: (POLL)

Polling interval (if in poll mode) (not used, set to zero).

Legal values: 0 – 65535

Default value: 0

⇒ NOTE:

For *UNIX* RTR Operating System Release 21.3 only:

If this record is being used as the DCI option record, this field will be interpreted as the minimum polling interval in milliseconds.

Appropriate values to use for DCI should range from 10 to 65535.

These values will not be validated by RC/V.

5.time_out: (TIMEOUT)

Audit time-out interval (not used, set to zero).

Legal values: 0 – 65535

Default value: 0

⇒ NOTE:

For *UNIX* RTR Operating System Release 21.3 only:

If this record is being used as the DCI option record, this field will be interpreted as the maximum polling interval in milliseconds.

Appropriate values to use for DCI should range from 10 to 65535.

These values will not be validated by RC/V.

6.picconf: (PICCONF)

Peripheral interface controller (PIC) configuration (not used, set to zero).

Legal values: 0 – 65535

Default value: 0

⇒ NOTE:

For *UNIX* RTR Operating System Release 21.3 only:

If this record is being used as the DCI option record, this field will be interpreted as the auto restore interval in seconds. Appropriate values to use for DCI should range from 30 to 65535. These values will not be validated by RC/V.

7.pcoos: (PCOOS)

Peripheral Controller (PC) out-of-service list. Bits 15 – 0 imply PCs 15 – 0. Always initialized to 0xffff.

Legal values: 0 – 65535, HEX

Default value: 0xffff

8.comoos: (COMOOS)

PC community out-of-service list. Bits 15 – 12 imply PC community 0 – 3. Always set to 0xf000.

Legal values: 0 – 65535, HEX

Default value: 0xf000

9.intvps: (INTVPS)

Intervals per second. Number of IOP batching interrupts per second (4 to 100). This field is normally set to 40 interrupts per second. This parameter greatly affects system response.

Legal values: 5 – 200

Default value: 80

10.dgnflg: (DGNFLG)

Diagnostic control flag (not used, set to zero).

Legal values: 0 – 65535, HEX

Default value: 0

ciopt

This form gives information for craft interface peripheral hardware.

Form Layout

Craft Interface Option Block	ciopt	(1/1)
		(Recent Change and Verify)
1.option_name: _____		
2.ttopt_name: _____	3.cdopt_name: _____	4.eaiopt_name: _____
5.logon_dev:_	6.ttyacctyp: _____	7.upd_subdev:_
8.port1: _____	9.port2: _____	

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. A seven-character name uniquely identifying this ciopt option block. This is the key field.

Legal values: alphanumeric characters

2. ttopt_name: (TTOPTNAME)

Name of the ttopt form corresponding to this terminal.

Legal values: alphanumeric characters, null

3. cdopt_name: (CDOPTNAME)

Name of the cdopt record corresponding to this form.

Legal values: alphanumeric characters, null

4. eaiopt_name: (EAIOPTNAME)

Name of eaiopt record.

Legal values: alphanumeric characters, null

5. logon_dev: (LOGONDEV)

Indicates whether this device is a login device.

Legal values: y, n, null

Default value: n

6. ttyacctyp: (TTYACCTYP)

Hardware access type to TTY.

Legal values: acu indicates automatic call unit
 dial indicates dial
 direct indicates direct hardware line
 null

7. upd_subdev: (UPDSUBDEV) Required

Update subdevice header each 5 seconds?

Legal values: y, n

8. port1: (PORT1)

Port assignment.

Legal values: NOPORT, PRT, VDU

Default value: VDU

9. port2: (PORT2)

Port assignment.

Legal values: NOPORT, PRT, VDU

Default value: NOPORT

⇒ NOTE:

On the TN74, port1 and port2 are physically connected as follows if both ports are enabled and are in full duplex mode:

- The output lines from the TN74 to the device are in parallel. Thus, any output to one port also goes to the other.
- The input lines to the TN74 from the device are electronically "OR'd" together; therefore, any response from either port will be interpreted as

input for both. Any condition such that the input from both ports are not identical will result in unpredictable results. The following basic scenarios exist that will cause this problem:

- The devices respond differently to data received from the TN74.
- One device responds more slowly than the other device to data received from the TN74.
- Different input is received from both devices by the controller.

It is strongly recommended that only one port (either port1 or port2) be defined as "VDU" in the ECD or you will run the risk of unpredictable results.

Form Checks

```
ttopt_name must_not_equal option_name
  error: "ttopt_name cannot be the same as option_name";

cdopt_name must_not_equal option_name
  error: "cdopt_name cannot be the same as option_name";

ttopt_name must_not_equal cdopt_name
  error: "ttopt_name and cdopt_name cannot be the same";

eaiopt_name must_not_equal option_name
  error: "eaiopt_name cannot be the same as option_name";

eaiopt_name must_not_equal ttopt_name
  error: "eaiopt_name and ttopt_name cannot be the same";

eaiopt_name must_not_equal cdopt_name
  error: "eaiopt_name and cdopt_name cannot be the same";
```

classdef

This form specifies a list of devices that are to be associated with a particular spooler output class.

Form Layout

```
Class Definition Record                                classdef      (1/1)
                                                       (Recent Change and Verify)

1.class_name: __   2.alarm: _

3.device_list      logical_devname                    8.rep_dest: _____

   1) _____   11) _____
   2) _____   12) _____
   3) _____   13) _____
   4) _____   14) _____
   5) _____   15) _____
   6) _____   16) _____
   7) _____   17) _____
   8) _____   18) _____
   9) _____   19) _____
  10) _____  20) _____
```

Definition of Fields

1. class_name: (CLASSNAME) Required

Class name. This is a key field.

Legal values: 0 – 253

2. alarm: (ALARM)

Alarm enable. This specifies whether the alarms on output messages will be sent on to the Alarm Control Process (ACP). Some processes interface directly to the ACP making this option necessary.

Legal values: y, n, null

Default value: y

3. device_list: (DEVICELIST)

device_list.logical_devname (*DEVICELIST.LOGICALDEVNAME[ROW])

Logical device name. When an output message is requested for a class, it will be queued for each name in this list.

Legal values: alphanumeric characters, null

8. rep_dest: (REPDEST)

Report destination.

Legal values: printable characters, null

cpblx3

This form gives options for the X.25 data link handler with both level 2 and level 3 protocols.

Form Layout

cpblx3 (1/4)
(Recent Change and Verify)

Communications Protocol Option Block

1.option_name: _____

2.speed: _____ 3.duplex: _____ 4.ds_type: _____

5.carrier:_____ 6.line_access: _____ 7.mode_op:_____

8.pdtime1: _____ 9.pdtime2: _____ 10.window:_____

11.rexmit:_____ 12.dcedte:_____ 13.pcsd_stat: _____

14.config: _____

cpblx3 (2/4)

15.ldtel sid tlim telno

1) _____

2) _____

3) _____

4) _____

19.security:_____ 20.link_id:_____

21.feid_len: _____ 22.neid_len: _____ 23.password_len: _____

24.feid.fend

1) _____ 4) _____ 7) _____ 10) _____ 13) _____

2) _____ 5) _____ 8) _____ 11) _____ 14) _____

3) _____ 6) _____ 9) _____ 12) _____ 15) _____

cpblx3 (3/4)

35.neid.nend

1) ___ 4) ___ 7) ___ 10) ___ 13) ___
2) ___ 5) ___ 8) ___ 11) ___ 14) ___
3) ___ 6) ___ 9) ___ 12) ___ 15) ___

46.pswrd.pwr

1) ___ 4) ___ 7) ___ 10) ___ 13) ___
2) ___ 5) ___ 8) ___ 11) ___ 14) ___
3) ___ 6) ___ 9) ___ 12) ___ 15) ___

57.perm_state: ___ 58.outscrsiz: ___ 59.outpktsiz: ___

60.inscrsiz: ___ 61.inpktsiz: ___

cpblx3 (4/4)

62.hup:_ 63.spckt: ___ 64.x3wsize: ___ 65.x3psize: ___

66.x3pvcn.ldinno

1) ___ 6) ___ 11) ___ 16) ___ 21) ___ 26) ___ 31) ___
2) ___ 7) ___ 12) ___ 17) ___ 22) ___ 27) ___ 32) ___
3) ___ 8) ___ 13) ___ 18) ___ 23) ___ 28) ___
4) ___ 9) ___ 14) ___ 19) ___ 24) ___ 29) ___
5) ___ 10) ___ 15) ___ 20) ___ 25) ___ 30) ___

81.time_ti: ___ 82.time_tj: ___ 83.time_tf: ___ 84.time_t1: ___

85.time_td: ___ 86.time_ts: ___ 87.time_ack: ___

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is a key field.

Legal values: alphanumeric characters

2. speed: (SPEED)

Data rate of associated communications equipment.

Legal values: 110, 300, 1200, 1800, 2400, 4800, 9600, 56000, 64000, null

3. duplex: (DUPLEX)

Full or half-duplex.

Legal values: full, half, null

4. ds_type: (DSTYPE)

PC modem type.

Legal values:	201CDDD	209A	NOPORT
	201CPL	212A	PL201C
	202/108	2024A	RS449
	208A	2048A	ITAPAC
	703COU	703CCUL	703CCUH
	703CDO	RS232	RS422
	RS423	208ADDD	CCITT
	208B	NODS	null

Enter RS449 when the PC modem is a 500B data set.

⇒ NOTE:

Multiple values may be considered equivalent in the database. Any legal value may be entered, but the value may not be displayed by Recent Change and Verify. The following table shows which values are displayed for the set of equivalent values.

Displayed Value	Equivalent Values
703COU	703COU 212A
703CCUL	703CCUL 208A
703CCUH	703CCUH 208B
703CDO	703CDO 209A
RS232	RS232 RS449
RS422	RS422 RS423 NODS NULL

5. carrier: (CARRIER)

Continuous or switched carrier.

Legal values: c, s, null

6. line_access: (LINEACCESS)

Type of line access.

Legal values: direct, private, noACU, ACU, null

7. mode_op: (MODEOP)

Mode operation. Block or send/receive mode of operation.

Legal values: b, s, null

8. pdtime1: (PDTIME1) Required

Maximum time to wait for level 2 protocol acknowledgment.

Values entered represent tenths of a second.

Legal values: 0 – 255, HEX

9. pdtime2: (PDTIME2) Required

Maximum time to allow data link to be idle.

Values entered represent tenths of a second.

Legal values: 0 – 255, HEX

10. window: (WINDOW) Required

Maximum number of I frames a station may have outstanding.

Legal values: 0 – 7

11. rexmit: (REXMIT) Required

Maximum number of retransmissions of an individual frame.

Legal values: 0 – 7

12. dcedte: (DCEDTE) Required

Data terminal or Data Circuit-terminating Equipment (DCE).

Legal values: 0 (data terminal equipment)
1 (data circuit-terminating equipment)

13. pcsd_stat: (PCSDSTAT)

Normal status for this pcsd.

Legal values: active (active device of duplex pair)
standby (standby device of duplex pair)
null

14. config: (CONFIG)

Simplex, duplex, or multiple configuration.

Legal values: duplex (one of a duplex pair)
null
multiple
simplex [simplex Peripheral Controller Subdevice
(PCSD)]

15. ldtel (LDTEL)

System identification (id), time limit, telephone number.

ldtel.sid (*LDTEL.SID[ROW])

System identifier.

Legal values: 0 – 65535, HEX, null

⇒ NOTE:

A value of 65535 in row 1 will deactivate the Communications Protocol Handler's (CPH) packet retransmission facility. This will result in no packets being retransmitted. Any other value will result in the default action of retransmitting a packet up to two times by CPH.

ldtel.tlim (*LDTEL.TLIM[ROW])

Time limit (seconds).

Legal values: 0 – 255, HEX, null

ldtel.telno (*LDTEL.TELNO[ROW])

Telephone number.

Legal values: alphanumeric or null

19. security: (SECURITY)

Check security information.

Legal values: y, n, null

20. link_id: (LINKID)

Link identification number.

Legal values: 0 – 7

Default value: 0

21. feid_len: (FEIDLEN)

Length of far-end system identifier.

Legal values: 0 – 15

Default value: 0

22. neid_len: (NEIDLEN)

Length of near-end system identifier.

Legal values: 0 – 15

Default value: 0

23. password_len: (PASSWORDLEN)

Length of password.

Legal values: 0 – 15

Default value: 0

24. feid.fend (*FEID.FEND[ROW])

Far-end system identifier.

Legal values: 0 – 255, HEX, null

Default value: 0

35. neid.nend (*NEID.NEND[ROW])

Near-end system identifier.

Legal values: 0 – 255, HEX, null

Default value: 0

46. pswrd.pwr (*PSWRD.PWRD[ROW])

Password.

Legal values: 0 – 255, HEX, null

Default value: 0

57. perm_state: (PERMSTATE) Required

Dial-up permission state.

Legal values: allow (allow dial-up connection)
condalw (if duplex and primary not active,
allow dial-up connection)
inhibit (inhibit dial-up connection)

58. outscrsiz: (OUTSCRSIZ) Required

Output buffer area size (words).

Legal values: 0 – 65535, HEX

⇒ NOTE:

If increasing the size of this buffer, the growth/degrowth procedures for the SDL must be followed. Failure to follow the procedures will cause the IOP's to initialize.

59. outpktsiz: (OUTPKTSIZ) Required

Data portion of output buffer (bytes).

Legal values: 0 – 640, 0x0 – 0x280, null

60. inscrlsiz: (INSCRSIZ) Required

Input buffer area size (words).

Legal values: 0 – 65535, HEX

⇒ NOTE:

If increasing the size of this buffer, the growth/degrowth procedures for the SDL must be followed. Failure to follow the procedures will cause the IOP's to initialize.

61. inpktsiz: (INPKTSIZ) Required

Data portion of input buffer (bytes).

Legal values: 0 – 640, 0x0 – 0x280, null

62. hup: (HUP)

Hang up on last close.

Legal values: y, n, null

63. spckt: (SPCKT) Required
Link data packet size.
Legal values: 0 – 255
64. x3wsize: (X3WSIZE) Required
Level 3 protocol window size.
Legal values: 0 – 7, 0x0 – 0x7, null
65. x3psize: (X3PSIZE) Required
Level 3 protocol packet size.
Legal values: 0 – 637, 0x0 – 0x27D, null
66. x3pvcn.lidunno (*X3PVCN.LDINNO[ROW])
Logical channel number corresponding to partition number.
Legal values: 0 – 4095, 0x0 – 0xFFFF, null
81. time_ti (TIMETI)
Interrupt packet timer.
Legal values: 180 – 600 seconds
Default value: 180
82. time_tj: (TIMETJ)
Reject packet timer.
Legal values: 60 – 600 seconds
Default value: 60
83. time_tf: (TIMETF)
Flow control packet timer.
Legal values: 10 – 600 seconds
Default value: 60
84. time_tl: (TIMETL)
Unacknowledged data packet timer.
Legal values: 60 – 600 seconds
Default value: 60

85. time_td: (TIMETD)

Reset request timer.

Legal values: 180 – 600 seconds

Default value: 180

86. time_ts: (TIMETS)

Restart request timer.

Legal values: 180 – 600 seconds

Default value: 180

87. time_ack: (TIMEACK)

Acknowledgment timer.

Legal values: 5 – 300 seconds

Default value: 5

cpphnone

This form gives the options for the X.25 data link handler with only level 2 protocols.

Form Layout

cpphnone (1/4)
(Recent Change and Verify)

Communications Protocol Option Block

1.option_name: _____

2.speed: _____ 3.duplex: _____ 4.ds_type: _____

5.carrier:_ 6.line_access: _____ 7.mode_op:_

8.pdtime1: _____ 9.pdtime2: _____ 10.window:_

11.rexmit:_ 12.dcedte:_ 13.pcsd_stat: _____

14.config: _____

cpphnone (2/4)

15.ldtel sid tlim telno

1) _____
2) _____
3) _____
4) _____

19.security:_ 20.link_id:_

21.feid_len: _ 22.neid_len: _ 23.password_len: _

24.feid.fend

1) _____ 4) _____ 7) _____ 10) _____ 13) _____
2) _____ 5) _____ 8) _____ 11) _____ 14) _____
3) _____ 6) _____ 9) _____ 12) _____ 15) _____

cpphnone (3/4)

35.neid.nend

1) ___ 4) ___ 7) ___ 10) ___ 13) ___
2) ___ 5) ___ 8) ___ 11) ___ 14) ___
3) ___ 6) ___ 9) ___ 12) ___ 15) ___

46.pswrd.pwr

1) ___ 4) ___ 7) ___ 10) ___ 13) ___
2) ___ 5) ___ 8) ___ 11) ___ 14) ___
3) ___ 6) ___ 9) ___ 12) ___ 15) ___

57.perm_state: ___ 58.outscrsiz: ___ 59.outpktsiz: ___

60.inscrsiz: ___ 61.inpktsiz: ___

cpphnone (4/4)

62.hup:_63.spckt: ___ 64.opktsiz: ___

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is a key field.

Legal values: alphanumeric characters

2. speed: (SPEED)

Data rate of associated communications equipment.

Legal values: 110, 300, 1200, 1800, 2400, 4800, 9600, 56000, 64000, null

3. duplex: (DUPLEX)

Full or half-duplex.

Legal values: full, half, null

4. ds_type: (DSTYPE)

PC modem type.

Legal values: 201CDDD 209A NOPORT
 201CPL 212A PL201C
 202/108 2024A RS449
 208A 2048A ITAPAC
 703COU 703CCUL 703CCUH
 703CDO RS232 RS422
 RS423 208ADDD CCITT
 208B NODS null

Enter RS449 when the PC modem is a 500B data set.

⇒ NOTE:

Multiple values may be considered equivalent in the database. Any legal value may be entered, but the value may not be displayed by Recent Change and Verify. The following table shows which values are displayed for the set of equivalent values.

Displayed Value	Equivalent Values			
703COU	703COU	212A		
703CCUL	703CCUL	208A		
703CCUH	703CCUH	208B		
703CDO	703CDO	209A		
RS232	RS232	RS449		
RS422	RS422	RS423	NODS	NULL

5. carrier: (CARRIER)

Continuous or switched carrier.

Legal values: c, s, null

6. line_access: (LINEACCESS)

Type of line access.

Legal values: direct, private, noACU, ACU, null

7. mode_op: (MODEOP)

Mode operation. Block or send/receive mode of operation.

Legal values: b, s, null

8. pdtime1: (PDTIME1) Required

Maximum time to wait for level 2 protocol acknowledgment.

Values entered represent tenths of a second.

Legal values: 0 – 255, HEX

9. pdtime2: (PDTIME2) Required

Maximum time to allow data link to be idle.

Values entered represent tenths of a second.

Legal values: 0 – 255, HEX

10. window: (WINDOW) Required

Maximum number of I frames a station may have outstanding.

Legal values: 0 – 7

11. rexmit: (REXMIT) Required

Maximum number of retransmissions of an individual frame.

Legal values: 0 – 7

12. dcedte: (DCEDTE) Required

Data terminal or data circuit-terminating equipment.

Legal values: 0 (data terminal equipment)
1 (data circuit-terminating equipment)

13. pcsd_stat: (PCSDSTAT)

Normal status for this pcsd.

Legal values: active (active device of duplex pair)
standby (standby device of duplex pair)
null

14. config: (CONFIG)

Simplex, duplex, or multiple configuration.

Legal values: duplex (one of a duplex pair)
multiple
null
simplex (simplex PCSD)

15. ldotel (LDTEL)

System identification (id), time limit, telephone number.

ldtel.sid (*LDTEL.SID[ROW])

System identifier.

Legal values: 0 – 65535, HEX, null

ldtel.tlim (*LDTEL.TLIM[ROW])

Time limit.

Legal values: 0 – 255, HEX, null

ldtel.telno (*LDTEL.TELNO[ROW])

Telephone number.

Legal values: alphanumeric characters, null

19. security: (SECURITY)

Check security information.

Legal values: y, n, null

20. link_id: (LINKID)

Link identification number.

Legal values: 0 – 7

Default value: 0

21. feid_len: (FEIDLEN)

Length of far-end system identifier.

Legal values: 0 – 15

Default value: 0

22. neid_len: (NEIDLEN)

Length of near-end system identifier.

Legal values: 0 – 15

Default value: 0

23. password_len: (PASSWORDLEN)

Length of password.

Legal values: 0 – 15

Default value: 0

24. feid.fend (*FEID.FEND[ROW])

Far-end system identifier.

Legal values: 0 – 255, HEX, null

35. neid.nend (*NEID.NEND[ROW])

Near-end system identifier.

Legal values: 0 – 255, HEX, null

46. pswrd.pwr (*PSWRD.PWRD[ROW])

Password.

Legal values: 0 – 255, HEX, null

57. perm_state: (PERMSTATE) Required

Dial-up permission state.

Legal values:	allow	(allow dial-up connection)
	condalw	(if duplex and primary not active, allow dial-up connection)
	inhibit	(inhibit dial-up connection)

58. outscrsiz: (OUTSCRSIZ) Required

Output buffer area size (words).

Legal values: 0 – 65535, HEX

59. outpktsiz: (OUTPKTSIZ) Required

Data portion of output buffer (bytes).

Legal values: 0 – 640, 0x0 – 0x280

60. inscrsiz: (INSCRSIZ) Required

Input buffer area size (words).

Legal values: 0 – 65535, HEX

61. inpktisiz: (INPKTSIZ) Required

Data portion of input buffer (bytes).

Legal values: 0 – 640, 0x0 – 0x280

62. hup: (HUP)

Hang up on last close.

Legal values: y, n, null

63. spckt: (SPCKT) Required

Link data packet size.

Legal values: 0 – 255, HEX

64. opktsiz: (OPKTSIZ) Required

Number of bytes in output packet.

Legal values: 0 – 65535, HEX

dbinfo

This form obtains various information about the database from the database.

Form Layout

```
dbinfo      (1/6)
Database Information Form (Execute Only)
1.dbinfo_opf: _____
*           ** INDEX **           *
*                                     tab (>) to: *
* 1) UCB List Function              2         *
*                                     *         *
* 2) IOP Device List Function       8         *
*                                     *         *
* 3) Pointer List Function          14        *
*                                     *         *
* 4) Form Rid Function              21        *
*                                     *         *
```

```
dbinfo      (2/6)
2.ucb_list:_
3.ucb_start
   complex_name: _____ complex_number: ___
   unit_name: _____ unit_number: ___
8.iop_list:_
9.iop_start
   complex_name: _____ complex_number: ___
   unit_name: _____ unit_number: ___
```

dbinfo (3/6)

14.pointer_list: _

15.form_type: _____

16.form_inst_key (1 to 4 part key)

keyfield1: _____

keyfield2: ____

keyfield3: _____

keyfield4: ____

dbinfo (4/6)

21.get_form_rid: _

22.form_info:

type_of_form 1) _____

keyfld1: _____

keyfld2: _____

keyfld3: _____

keyfld4: ____

type_of_form 2) _____

keyfld1: _____

keyfld2: _____

keyfld3: _____

keyfld4: ____

dbinfo (5/6)

22.form_info (cont.)

type_of_form 3) _____

keyfld1: _____

keyfld2: _____

keyfld3: _____

keyfld4: _____

type_of_form 4) _____

keyfld1: _____

keyfld2: _____

keyfld3: _____

keyfld4: _____

dbinfo (6/6)

22.form_info (cont.)

type_of_form 5) _____

keyfld1: _____

keyfld2: _____

keyfld3: _____

keyfld4: _____

Definition of Fields

1. dbinfo_opf: (DBINFOOPF) Required

dbinfo output filename. This is the key field.

Legal value: printable characters

2. ucb_list: (UCBLIST)

ucb_list indicates whether a list of ucbs is desired.

Legal values: y, n, null

3. ucb_start (UCBSTART)

ucb_start.complex_name: (*UCBSTART.COMPLEXNAME)

The first part of a four-part ucb identifier.

ucb_start.complex_number: (*UCBSTART.COMPLEXNUMBER)

The second part of a four-part ucb identifier.

ucb_start.unit_name: (*UCBSTART.UNITNAME)

The third part of a four-part ucb identifier.

ucb_start.unit_number: (*UCBSTART.UNITNUMBER)

The fourth part of a four-part ucb identifier.

8. iop_list: (IOPLIST)

iop_list indicates whether an IOP list is desired.

Legal values: y, n, null

9. iop_start (IOPSTART)

(See Field 3 description.)

14. pointer_list: (POINTERLIST)

The pointer_list indicates whether a list of all forms pointing to a form is desired.

Legal values: y, n, null

15. form_type: (FORMTYPE)

Form type of record being pointed to.

Legal values: aphopt cpphnone eaiopt gfsappl logdev pphopt
 audrec dciopt errthld gfscom mdct tphopt
 cdopt device ftaminopt gfsinit mhdopt ttop
 chopt dfcopt ftamopt gfsresp nphopt ucb
 ciopt dst ftamuser gfssec objectid null
 cpblx3 duopt getty

16. form_inst_key: (FORMINSTKEY)

form_inst_key.keyfield1: (*FORMINSTKEY.KEYFIELD1) Key field 1.

form_inst_key.keyfield2: (*FORMINSTKEY.KEYFIELD2) Key field 2.

form_inst_key.keyfield3: (*FORMINSTKEY.KEYFIELD3) Key field 3.

form_inst_key.keyfield4: (*FORMINSTKEY.KEYFIELD4) Key field 4.

21. get_form_rid: (GETFORMRID)

This field asks if you want the Record Identifier (RID) of a specified form.

Legal values: y, n, null

22. form_info

form_info.type_of_form: (FORMINFO.TYPEOFFORM)

Type of form for which RID is desired.

Legal values:

aftopt	dciopt	mhdopt
aphopt	device	nphopt
aphuser	dfcopt	objectid
auderr	dst	pphopt
audinst	duopt	pphuser
audrec	eaiopt	runa
authdef	ecdorg	scsdbody
btihdev	errthld	scsdopt
btparm	ftaminopt	simcntl
bxslinfo	ftamopt	sphopt
bxslloc	ftamuser	splinfo
bxslrmt	getty	tphopt
cdopt	gfsappl	tsapinfo
cftinfo	gfscom	ttopt
chopt	gfsinit	ucb
ciopt	gfsresp	ularp
classdef	gfssec	workhrs
cpblx3	logdev	null
cpophone	mdct	

form_info.keyfld 1: (FORMINFO.KEYFLD1)

form_info.keyfld 2: (FORMINFO.KEYFLD2)

form_info.keyfld 3: (FORMINFO.KEYFLD3)

form_info.keyfld 4: (FORMINFO.KEYFLD4)

Form Checks

```
case (ucb_list):

  when {"y"}: do;
    ucb_start.unit_name must_not_be unknown
      error: "unit_name must be specified when UCB_list is desired";
    ucb_start.unit_number must_not_be unknown
      error: "unit_number must be specified when UCB_list is desired";
  end;

  when {"n", unknown}: do;
    ucb_start.complex_name must_be unknown
      error: "complex_name must be null";
    ucb_start.complex_number must_be unknown
      error: "complex_number must be null";
    ucb_start.unit_name must_be unknown
      error: "unit_name must be null";
    ucb_start.unit_number must_be unknown
      error: "unit_number must be null";
  end;
end;

case (iop_list);

  when {"y"}: do;
    iop_start.unit_name must_not_be unknown
      error: "unit_name must be specified when IOP_list is desired";
    iop_start.unit_number must_not_be unknown
      error: "unit_number must be specified when IOP_list is desired";
  end;

  when {"n", unknown}: do;
    iop_start.complex_name must_be unknown
      error: "complex_name must be null";
    iop_start.complex_number must_be unknown
      error: "complex_number must be null";
    iop_start.unit_name must_be unknown
      error: "unit_name must be null";
    iop_start.unit_number must_be unknown
      error: "unit_number must be null";
  end;
end;

case (pointer_list):

  when {"y"}: do
    form_type must_not_be unknown
      error: "form_type must be specified when POINTER_list
        is desired";
  end;
```

```
when {"n", unknown}: do
  form_type must_be unknown
  error: "form_type must be null";
  form_inst_key.keyfield1 must_be unknown
  error: "keyfield1 must be null";
  form_inst_key.keyfield2 must_be unknown
  error: "keyfield2 must be null";
  form_inst_key.keyfield3 must_be unknown
  error: "keyfield3 must be null";
  form_inst_key.keyfield4 must_be unknown
  error: "keyfield4 must be null";
end;
end;

case (get_form_rid);

when {"y"}: do;
  count (select type_of_form from form_info)

  must_not_equal 0
  error: "When get_form_rid is y, there must
  be at least one type_of_form
  for every form_info element:"

do;
  if type_of_form is known and type_of_form equals "ucb"
  then do;
    keyfld3 must_not_equal unknown
    error: "When type_of_form is ucb,
    keyfld3 must not be null";
    keyfld4 must_not_equal unknown
    error: "When type_of_form is ucb,
    keyfld4 must not be null";

  end;
  else do;
    if keyfld1 is unknown
    then type_of_form must_not_be known
    error: "When get_form_rid is y,
    keyfld1 must not be null";
    if type_of_form is unknown
    then keyfld1 must_not_be known
    error: "When get_form_rid is y,
    type_of_form must not be null";

  end;
end;
end;
end;
```

```
when {"n", unknown}: do;
  for every form_info element:
  do;
    type_of_form must_be unknown
      error: "When get_form_rid is n,
            type_of_form must be null" ;
    keyfld1 must_be unknown
      error: "When get_form_rid is n,
            keyfld1 must be null" ;
    keyfld2 must_be unknown
      error: "When get_form_rid is n,
            keyfld2 must be null" ;
    keyfld3 must_be unknown
      error: "When get_form_rid is n,
            keyfld3 must be null" ;
    keyfld4 must_be unknown
      error: "When get_form_rid is n,
            keyfld4 must be null" ;
  end;
end;
end;
```

dciopt

This form contains options that control operation of Dual Serial Channel (DSCH) Computer Interconnect (DCI) devices. Every DCI *ucb* record must have a *dciopt* record.

This form will be used by the DCI driver beginning with *UNIX*® Real-Time Reliable (RTR) Operating System Release 21.17 and replaces the *chopt* form used for the DCI in previous releases. Therefore, all existing DCI *chopt* forms must be removed from the Equipment Configuration Database (ECD) and *dciopt* forms must be inserted before the DCI can be used with Release 21.17 and later. The following table shows the mapping of *chopt* fields used for DCI with the corresponding fields in the *dciopt* form.

chopt field	dciopt field
1.option_name	1.option_name
no field	2.adj_prCSR_id
4.poll	3.min_poll_intv
5.time_out	4.max_poll_intv
6.picconf	5.auto_restore
no field	6.dcitype

Form Layout

DCI Option Block Record	dciopt	(1/1) (Recent Change and Verify)
1.option_name:_____		2.adj_prCSR_id:_____
3.min_poll_intv:_____		4.max_poll_intv:_____
5.auto_restore:_____		6.dcitype:_____

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This value is the same as field 57 (option_name) on the *ucb* form of the corresponding DCI. This is a key field.

Legal values: 1 – 8 alphanumeric characters

2. adj_prcsr_id: (ADJPRCSRID)

The id of the adjunct processor to which the 3B20D/3B21D computer is attached via the DCI cable.

Legal values: 0 - 3

Default value: 0

⇒ NOTE:

A DCI device is a logical unit defined in a *ucb* form in the ECD; it is not the adjunct processor.

3. min_poll_intv: (MINPOLLINTV)

The minimum polling period (in milliseconds). The adjunct's DCI driver will begin checking the queue every min_poll_intv milliseconds, increasing this in 10-millisecond increments until either a job is queued or it reaches max_poll_intv (field 4).

Legal values: 10 - 250

Default value: 10

4. max_poll_intv: (MAXPOLLINTV)

The maximum polling period (in milliseconds). This is the longest period the adjunct's DCI driver will wait before checking the queue for jobs.

Legal values: 10 - 250

Default value: 250

5. auto_restore: (AUTORESTORE)

The time (in seconds) before the 3B20D/3B21D computer DCI driver attempts to restore a DCI device previously removed by the driver because of non-3B computer impacting errors. The restore itself may take longer than the time specified in this field, but it will not begin until after auto_restore seconds have elapsed.

Legal values: 0 - 300

Default value: 120

6. dcitype: (DCITYPE)

This field determines the type of the DCI unit. A value of "1" indicates the DCI supports the High Availability-Operations and Maintenance Platform (HA-OMP) feature. DCI units that are not used in an HA-OMP configuration must use the default value ("0").

Legal values: 0 - 1

Default value: 0

device

This form defines a spooler output device.

Form Layout

```
device (1/3)
(Recent Change and Verify)

Device Definition Record

1.logical_devname:_____
2.path:_____
3.seqn:_      4.logfile:_      5.temp:_      6.priority:_
7.maxq:_____      8.blksize:_____
9.linelen:_____      10.logflimit:_____
11.time_stamp:_____      12.multiple_msg:_
```

```
device (2/3)

13.msg_head
  head1  head2  head3  head4  head5  head6  head7  head8
  _____
  head9  head10 head11 head12 head13 head14 head15 head16
  _____

30.msg_trailer
  tr1    tr2    tr3    tr4    tr5    tr6    tr7    tr8
  _____
  tr9    tr10   tr11   tr12   tr13   tr14   tr15   tr16
  _____
```

	device	(3/3)
47.printer_page:___	48.paging_limit:___	49.protocol:_____
50.dialogue:_	51.alt_device:_____	52.page_per_rpt:_
53.language:___		

Definition of Fields

1. logical_devname: (LOGICALDEVNAME) Required

Logical device name. This is the key field. Legal references to an output device are the logical name or a class specifying the logical name.

Legal values: alphanumeric characters

2. path: (PATH)

Device pathname. This is the pathname that the *UNIX*® system uses to reference the output device (that is, */dev/rop0*). For log files, this specifies all but the last character of the pathname.

Legal values: printable characters, null

3. seqn: (SEQN)

Enable sequence numbers. If the option is enabled, then output messages sent to this device will contain a sequence number indicating the order in which the spooler received the output message request.

Legal values: y, n, null

Default value: n

4. logfile: (LOGFILE)

Log file/regular file. A flag specifying whether the output device is to be treated as a log file or a regular file. The entries sent to a log file contain header information to aid utilities in processing log files. This flag also enables split files. When a log file grows beyond the log file overflow limit, it is closed and another log file is opened.

The names of the two files are formed by appending an American Standard Code for Information Interchange (ASCII) one or zero to the end of the device name.

Regular files are any spooler output device other than log files.

Legal values: y, n

Default value: n

5. temp: (TEMP)

Temporary Spooler Output Process (SOP). If an SOP encounters an output error and this flag specifies a "temporary" SOP, then the SOP will die. Messages sent to this device will be lost until the Coordinator of Spooler Output Processes (CSOP) receives an SOP restart request for the device. The restart request comes from a getty that knows the status of the output device.

An SOP that is "permanent" will not die on an output error. Processing will continue normally except that the current output message will be lost.

Legal values: y, n, null

Default value: n

6. priority: (PRIORITY)

Priority order queue. This option causes the output queue for a device to be kept in priority order rather than chronological order.

Legal values: y, n, null

Default value: n

7. maxq: (MAXQ)

This field specifies the maximum number of output messages that can be queued for this device.

Legal values: 0 – 32767

Default value: 50

8. blksize: (BLKSIZE)

This field specifies the size of an output buffer used by the write function to output to the device. (Normally, log files = 512, Switching Control Center (SCC) links = 1000, other devices = 512.)

Legal values: 0 – 32767

Default value: 512

9. linelen: (LINELEN)

Forced line length – 0 to disable. If this number is not zero, lines greater in length than the specified limit will be broken with a new line code. Forced line length has no effect on log files. (Log files = 0, SCC links = 80, most devices = 80.)

Legal values: 0 – 32767

Default value: 80

10. logflimit (LOGFLIMIT)

Log file overflow limit. If the output device is a log file, then this specifies the overflow limit for a log file half. (Log files = 15000, SCC and others = 0.)

Legal values: 0 – MAXINT

Default value: 15000

11. time_stamp: (TIMESTAMP)

Time stamp flag. If this field is set to msgon or msgoff, it will override the requested time stamp of an output message for the specified device. If the field is set to "dontcare," the requested time stamp will be used.

The option is overridden if the time stamp option in the miscellaneous spooler information record (splrinfo) is set to msgon or msgoff.

Legal values: dontcare, msgoff, msgon

Default value: dontcare

12. multiple_msg: (MULTIPLEMSG)

Multiple messages per write. If this flag is set, an SOP will put as many messages into an output buffer as possible before writing the buffer out to its device. This option only has an effect if the output queue for an SOP has more than one output message in it. An output buffer will only contain a partial message if the output message is larger than block size.

Legal values: y, n

Default value: n

13. msg_head.head1 — msg_head.head16:
(*MSGHEAD.HEAD1 – MSGHEAD.HEAD16)

Output message header. This is a 0- to 15-character string that will precede every output message on the device. To conform to Man-Machine Language (MML) output standards, this string should consist of a new line code. MML requires a blank line between output messages.

Legal values: 0 – 255, ESCAPE, HEX characters, QCHAR, null

Default value: x0

30. msg_trailer.tr1 — msg_trailer.tr16: (*MSGTRAILER.TR1 — MSGTRAILER.TR16)

Output message separator. This is a 0- to 15-character string that will follow every output message on the device.

Legal values: 0 – 255, ESCAPE, HEX characters, QCHAR, null

Default value: x0

47. printer_page: (PRINTERPAGE) Required

Size of a logical page, for example, the standard value for a Read-Only Printer (ROP) is 66.

Legal values: 0 – 255

Default value: 0

48. paging_limit: (PAGINGLIMIT) Required

Used to determine whether or not output paging will be performed on this device. If the value of this field is nonzero and the value of printer_page field (field 47) is greater than or equal to 10 and this device is not a logfile field (field 4), then paging will be performed on this device.

Legal values: 6 – 255

Default value: 6

49. protocol: (PROTOCOL)

Protocol used by the output device.

Legal values: no_prot, aom_prot, sccs_prot, aphprot, or null

Default value: no_prot

50. dialogue: (DIALOGUE) Required

Indicates whether or not this device is capable of supporting a dialogue shell.

Legal values. y or n

Default value: n

51. alt_device: (ALTDEVICE)

Alternative output device logical name.

If this device is designated as a dialogue device (refer to field 50 - dialogue), then CSOP will route spooler output to the alternative device when alternative routing is required. Alternative routing is required when spooler output intended for this device cannot be linked to the active dialogue session on the device.

Note that the language of the device (refer to field 53 - language) also factors into alternative routing. When alternative routing is required, CSOP will attempt to route the spooler output to an alternative device that has the same language as this device. If the language of this device does not match the language of the alternative device, then CSOP will attempt to route the spooler output to the alternative device of this device's alternative device and so on. In the case where a language match is found between this device and an alternative device in the chain of alternative devices, then CSOP will route the the spooler output to the first alternative device in the chain where the language match occurred. Otherwise, CSOP will route the spooler output to the first alternative device.

Legal values: printable characters or null

52. page_per_rpt: (PAGEPERRPT)

Indicates whether this device is capable of supporting a page per report.

Legal values: y or n

53. language: (LANGUAGE)

Indicates whether the language for this device is English or the national language.

Legal values: ENGL, NATL

Default value: ENGL

Form Checks

```
alt_device must_not_equal logical_devname  
error: "alt_device cannot be the same as logical_devname";
```

dfcopt

This form contains options which control operation of the Disk File Controller (DFC).

This form is not implemented in all releases.

Form Layout

dfcopt		(1/2)
Disk File Controller Option Block Record		(Recent Change and Verify)
1.option_name:_____	2.elevator:_____	
3.readretry:_____	4.rexflag:_____	
5.eirflag:_____	6.eirbufsz:_____	
7.bqsize:_____	8.hqsize:_____	
9.spqsize:_____	10.rspsize:_____	

dfcopt		(2/2)
11.timeoutval:_____	12.rdelay:_____	
13.wdelay:_____	14.dbsize:_____	
15.bitopts:_____		

Definition of Fields

1. option_name: Required

Option block name. This is a key field.

Legal value: 1 – 8 alphanumeric characters

2. elevator:

Allow job optimization using an elevator algorithm. The elevator algorithm sorts jobs based on disk drive head movement. If allowed (y), the algorithm will be enabled for all Moving Head Disks (MHDs) controlled by this DFC, unless it is disabled for a specific MHD by the elevator field in an *mhdopt* form. If disallowed (n), the elevator algorithm will not be enabled for any MHD controlled by this DFC.

This option is not implemented in all releases.

Legal value: y, n, null

3. readretry:

Allow the DFC to automatically retry a read job after a failure. If allowed (y), the read job retries will be enabled for all MHDs controlled by this DFC, unless it is disabled for a specific MHD by the readretry field in an *mhdopt* form. If disallowed (n), read retries will not be enabled for any MHD controlled by this DFC.

This option is not implemented in all releases.

Legal values: y, n, null

4. rexflag:

Allow the DFC to routinely exercise an MHD by submitting read jobs to an MHD to verify the media and its operating condition. If allowed (y), the routine exercise jobs will be enabled for all MHDs controlled by this DFC, unless it is disabled for a specific MHD by the rexflag field in an *mhdopt* form. If disallowed (n), routine exercise jobs will not be enabled for any MHD controlled by this DFC.

This option is not implemented in all releases.

Legal values: y, n, null

5. eirflag:

Control the print format of the DFC Enhanced Information Report (EIR). An EIR usually accompanies errors reported by the DFC and provides additional information useful in resolving the error.

Legal values: s A short EIR report will be printed
 l A long EIR report will be printed
 n No EIR report will be printed

This option is not implemented in all releases.

Default value: s

6. eirbufsz:

Size of the DFC EIR buffer (in number of computer words) allocated in the disk driver.

This option is not implemented in all releases.

Legal values: 0 – 2048 decimal, null

7. bqsize:

The number of usable entries allowed in the DFC base priority job queue.

This option is not implemented in all releases.

Legal values: 0 – 255 decimal, null

8. hqsize:

The number of usable entries allowed in the DFC high priority job queue.

This option is not implemented in all releases.

Legal values: 0 – 255 decimal, null

9. spqsize:

The number of usable entries allowed in the DFC special priority job queue.

This option is not implemented in all releases.

Legal values: 0 – 255 decimal, null

10. rspsize:

The number of usable entries allowed in the DFC job response queue.

This option is not implemented in all releases.

Legal values: 0 – 255 decimal, null

11. timeoutval:

DFC time-out value, in seconds. Specifies the maximum time that can elapse without any job responses returned by the DFC. Applies only when jobs are outstanding to the DFC. This value does not affect the job response rate, only the period considered for job time-outs.

This option is not implemented in all releases.

Legal values: 1 – 1800 decimal, null

12. rdelay:

Delay period between successive data transfer requests, in the direction of DFC to Dual Serial Channel (DSCH), in microseconds. Controls the rate at which the DFC requests services of the DSCH.

This option is not implemented in all releases.

Legal values: 0 – 1000 decimal, null

13. wdelay:

Delay period between successive data transfer requests, in the direction of DSCH to DFC, in microseconds. Controls the rate at which the DFC requests services of the DSCH.

This option is not implemented in all releases.

Legal values: 0 – 1000 decimal, null

14. dbsize:

The size of the internal DFC data buffer allocated to buffering of data between an MHD and the 3B20D/3B21D computers, in 512-byte blocks.

This option is not implemented in all releases.

Legal values: 1 – 256 decimal, null

15. bitopts:

A DFC option word, 32 bits wide. Provides 32 separate switches (on/off) to control various options within the DFC or disk driver. None of these bits are currently defined.

This option is not implemented in all releases.

Legal values: 0x0 – 0xFFFFFFFF HEX, null

dst

This form gives the information unique to daylight saving time instance.

Form Layout

Daylight Saving Time Options		dst	(1/1)
		(Recent Change and Verify)	
1.option_name:_____			
2.start_time:			
3.month:___		4.day_of_week:_	5.week_of_month:_
6.day_of_month:___		7.hour:___	8.minute:___
9.end_time:			
10.month:___		11.day_of_week:_	12.week_of_month:_
13.day_of_month:___		14.hour:___	15.minute___
16.years:		17.year_start:_____	18.year_end:_____

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is a key field.

Legal values: alphanumeric characters

2. start_time: (STARTTIME)

Time when daylight saving time starts.

3. start_time.month: (STARTTIME.MONTH) Required

Month in which daylight saving time starts.

Legal values: 1 – 12

4. start_time.day_of_week: (STARTTIME.DAYOFWEEK)

Day of the week on which daylight saving time starts.

Legal values: 1 – 7 or null

5. start_time.week_of_month: (STARTTIME.WEEKOFMONTH)

Week of the month on which daylight saving time starts.

Legal values: 1 – 5 or null

6. start_time.day_of_month: (STARTTIME.DAYOFMONTH)

Day of the month on which daylight saving time starts.

Legal values: 1 – 31 or null

7. start_time.hour: (STARTTIME.HOUR) Required

Hour of the day on which daylight saving time starts.

Legal values: 0 – 23

8. start_time.minute: (STARTTIME.MINUTE) Required

Minute of the day on which daylight saving time starts.

Legal values: 0 – 59

9. end_time: (ENDTIME)

Time when daylight saving time ends.

10. end_time.month: (ENDTIME.MONTH) Required

Month in which the daylight saving time ends.

Legal values: 1 – 12

11. end_time.day_of_week: (ENDTIME.DAYOFWEEK)

Day of the week on which daylight saving time ends.

Legal values: 1 – 7 or null

12. end_time.week_of_month: (ENDTIME.WEEKOFMONTH)

Week of the month on which daylight saving time ends.

Legal values: 1 – 5

13. end_time.day_of_month: (ENDTIME.DAYOFMONTH)

Day of the month on which daylight saving time ends.

Legal values: 1 – 31 or null

14. end_time.hour: (ENDTIME.HOUR) Required

Hour of the day on which daylight saving time ends.

Legal values: 0 – 23

15. end_time.minute: (ENDTIME.MINUTE) Required

Minute of the hour on which daylight saving time ends.

Legal values: 0 – 59

16. years:

Start and end year for dst.

17. years.year_start:

Year in which daylight saving time starts.

Legal values: 1910 – 2037 or null

18. years.year_end:

Year in which daylight saving time ends.

Legal values: 1910 – 2037 or null

Form Checks

```
if start_time.day_of_week is unknown
  then start_time.day_of_month must be known
  error: "start day_of_month must be set when start day of week is unknown";

if end_time.day_of_week is unknown
  then end_time.day_of_month must be known
  error: "end day_of_month must be set when end day_of_week is unknown";

if start_time.day_of_week is known
  then start_time.week_of_month must be known
  error: "start week_of_month must be set when start day_of week is set";

if start_time.day_of_week is unknown
  then start_time.week_of_month must be unknown
  error: "start week_of_month must not be set when start day_of week is not set";

if end_time.day_of_week is unknown
  then end_time.week_of_month must be unknown
  error: "end week_of_month must not be set when end day_of week is not set";

if end_time.day_of_week is known
  then end_time.week_of_month must be known
  error: "end week_of_month must be set when end day_of week is set ";

if start_time.day_of_week is known
  then start_time.day_of_month must be unknown
  error: "start day_of_month must be set when start day_of week is set ";

if end_time.day_of_week is known
  then end_time.day_of_month must be unknown
  error: "end day_of_month must not be set when end day_of week is set ";

case (start_time.month);
  when {2}:
    do;
      if start_time.day_of_month is known
        then start_time.day_of_month must_not_be_greater_than 29
        error: "February has at most 29 days";
      end;
    end;

  when {4,6,9,11}:
    do;
      if start_time.day_of_month is known
        then start_time.day_of_month must_be_less_than 31
        error: "There are only 30 days in the months of
          April, June, September, and November";
      end;
    end;
end

case (end_time.month);
  when {2}:
    do;
      if end_time.day_of month is known
        then end_time.day_of_month must_not_be_greater_than 29
        error: "February has at most 29 days";
      end;
    end;
```

```
when {4,6,9,11}:
  do;
    if end_time.day_of_month is known
      then end_time.day_of_month must_be_less_than 31
        error: "There are only 30 days in the months of
          April, June, September, and November";
    end;
  end;

if years.year_start is unknown
  then years.year_end must be unknown
    error: "year_end must not be set when year_start is not set";

if years.year_start is known
  then if years.year_end is known
    then years.year_end must_be_greater_than years.year_start
      error: "year_start must be less than year_end";
```

duopt

This form contains options for the Direct User Interface (DUI) handler.

Form Layout

duopt		(1/1)
Direct User Interface Handler Options		(Recent Change and Verify)
1.option_name: _____	2.duiport: ____	
3.duin1: __	4.duit1: __	
5.duit2: __	6.duin2:_	
7.duihdlc:_	8.duiwindow: _	

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is the key field.

Legal values: alphanumeric characters

2. duiport: (DUIPORT) Required

CCITT = high, RS449 = low-speed interface.

3. duin1: (DUIN1) Required

X.25 N1 parameter.

Legal values: 0 – 255

4. duit1: (DUIT1) Required

X.25 T1 parameter.

Legal values: 1 – 256

5. duit2: (DUIT2) Required

X.25 T2 parameter.

Legal values: 1 – 256

6. duin2: (DUIN2) Required

X.25 N2 parameter.

Legal values: 2 – 7

7. duihdlc: (DUIHDLC) Required

High-level Data Link Control (HDLC) select.

Legal values: 0, 1

8. duiwindow: (DUIWINDOW) Required

X.25 window size.

Legal values: 1 – 31

duoptdyn

This form contains verify-only data from the Data User Interface (DUI) handler option block.

Form Layout

Duopt Dynamic Data	duoptdyn	(1/1)	(Review Only)
1.option_name: _____			
2.data_rate: _	3.duplex: ____		
4.poll_rate: __	5.dma:_		
6.invert:_	7.t_size: _____		
8.t_addr: _____			

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is the key field.

Legal values: alphanumeric characters

2. data_rate:

Data rate.

Legal values: 0 – 15

3. duplex:

Duplex.

Legal values: full, half, null

4. poll_rate:

Polling rate.

Legal values: 0 – 255

5. dma:

Direct Memory Access (DMA) job number.

Legal values: 0 – 3

6. invert:

Invert address field.

Legal value: 0, 1

7. t_size:

DUI table size in bytes.

Legal values: 0 – 65535

8. t_addr:

DUI table address.

Legal values: 0 – 65535

eaiopt

This form gives the information unique to the Emergency Action Interface (EAI) terminal.

Form Layout

eaiopt (1/3)
(Recent Change and Verify)

Emergency Action Interface Option

1.option_name:_____	2.short_timeout:_____	
3.long_timeout:_____	4.cmon:_____	
5.sysname:_____	6.systype:_____	
7.genid:_____	8.eaiinit_hex:_	
9.start_save_time:___	10.end_save_time:___	
11.time_zone		
12.std_time:___	13.min_gmt:___	14.dst_time:_____

eaiopt (2/3)

15.dlg_req_char:___	16.dyn_req_char:___	17.datestamp_layout:_____
18.dlg_header:	_____	
19.dlg_trailer:	_____	
20.sep_user_code:_	21.user_code_term:___	22.password_term:___
23.passcard_t/o:_	24.passcard_reads:_	25.break_attempts:___
26.min_psswd_chg:___	27.max_psswd_chg:___	28.disable_attempts:_

eaiopt (3/3)

29.virtual_switch:_____

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is the key field.

Legal values: alphanumeric characters

2. short_timeout: (SHORTTIMEOUT) Required

Short time-out cycle length. Updates minutes and seconds fields of date and 24-hour clock.

Legal values: 0 – MAXINT, HEX, null

Default value: 5000

3. long_timeout: (LONGTIMEOUT) Required

Long time-out length for sending messages to the Craft Interface Integrity Monitor (CMON). Updates date and hours fields of date and 24-hour clock.

Legal values: 0 – MAXINT, HEX, null

Default value: 30000

4. cmon: (CMON) Required

Time-out response from CMON.

Legal values: 0 – MAXINT, HEX, null

Default value: 10000

5. sysname: (SYSNAME)

System name.

Legal values: printable characters, null

6. systype: (SYSTYPE)

System type.

Legal values: printable characters, null

7. genid: (GENID)

Generic identification (id).

Legal values: printable characters, null

8. eaiinit_hex: (EAIINITHEX)

Initial hexadecimal digit for displaying EAI status on board.

Legal values: c handler can access at least one MTTYPC
d EAI for on-line Central Control (CC) is accessible from
functioning MTTYPC
e EAI receives all well response from CMON
f one of the above false
null

9. start_save_time: (STARTSAVETIME)

Day of the year on which daylight saving time starts.

Legal values: 1 – 366, null

10. end_save_time: (ENDSAVETIME)

Day of the year on which daylight saving time ends. This field is replaced by the *dst* form.

Legal values: 1 – 366, null

11. time_zone: (TIMEZONE)

Time zone.

12. time_zone.std_time: (*TIMEZONE.STDTIME) Required

Standard time to use.

Legal values: three or four alphabetic characters

Default value: GMT

13. time_zone.min_gmt: (*TIMEZONE.MINSGMT) Optional

Minutes to adjust time from Greenwich Mean Time (GMT).

Legal values: -720 to 720, null

14. time_zone.dst_time: (*TIMEZONE.DSTTIME)

Daylight standard time to use.

Legal values: 3 - 5 alphabetic characters, null

15. dlg_req_char: (DLGREQCHAR)

Character used to request initiation of a dialogue session.

Legal values: 0 – 255, QCHAR, HEX2D, ESCAPE, or null

Default value: x07

16. dyn_req_char: (DYNREQCHAR)

Character used to request dialogue with terminal authority.

Legal values: 0 – 255, QCHAR, HEX2D, ESCAPE, or null

Default value: x23

17. datestamp_layout: (DATESTAMPLAYOUT) Required

Layout of date stamp, including “m,” “d,” and “y” for month, day, and year plus optional separators; for example, “/” and “:”.

Legal values: printable characters

18. dlg_header: (DLGHEADER)

Layout of dialogue header. This field may include any printable character plus control strings that the spooler replaces with designated variables including:

Control String	Description
%s	system name
%d	date stamp (in format specified in this record)
%t	time
%i	identity (person or terminal)
%n	dialogue number

Legal values: printable characters or null

19. dlg_trailer: (DLGTRAILER)

Layout of dialogue trailer. See description of dialogue header.

Legal values: printable characters or null

20. sep_user_code: (SEPUSERCODE)

Separate user code used.

Legal values: y, n

Default value: n

21. user_code_term: (USERCODETERM)

User code termination character.

Legal values: \r or semicolon

Default value: \r

22. password_term: (PASSWORDTERM)

Password termination character.

Legal values: \r or semicolon

Default value: \r

23. passcard_t/o: (PASSCARDT/O)

Passcard reception time-out.

Legal values: 1 – 9

Default value: 5

24. passcard_reads: (PASSCARDREADS)

Number of passcard read attempts.

Legal values: 1 – 9

Default value: 3

25. break_attempts: (BREAKATTEMPTS)

Password break attempt threshold.

Legal values: 0 – 10, or null

Default value: 3



CAUTION:

*The three fields `min_psswd_chg`, `max_psswd_chg`, and `dsable_attempt` are for the Password Enhancement Feature only. If all three fields are null, the feature is turned off and cannot be turned on by using recent change. Customers who have purchased the password enhancement feature are warned: **DO NOT ENTER "null" IN THESE FIELDS OR THE FEATURE WILL BE LOST FOREVER.***

26. `min_psswd_chg`: (MINPSSWDCHG)

Minimum time (in weeks) before password change is required.

Legal values: 0 – 99

Default value: 1

27. `max_psswd_chg`: (MAXPSSWDCHG)

Maximum time (in weeks) before password change is required.

Legal values: 0 – 99

Default value: 13

28. `dsable_attempt`: (DSABLEATTEMPT)

Number of bad login attempts allowed before disabling the user.

Legal values: 0 – 9

Default value: 3

29. `virtual_switch`: (VIRTUALSWITCH)

Switch Platform Identifier.

Legal values: DRM, VCDX, or null

Form Checks

```
case (time_zone.std_time);

  when{"GMT"}: do;
    (time_zone.min_gmt must_be unknown) or (time_zone.min_gmt equals 0)
    error: "when GMT is standard time, min_gmt must be either null
           or 0";
  end;

  when{"gmt"}: do;
    (time_zone.min_gmt must_be unknown) or (time_zone.min_gmt equals 0)
    error: "when gmt is standard time, min_gmt must be either null
           or 0";
  end;

  otherwise:
    time_zone.min_gmt must_be known
    error: "when standard time is not GMT, min_gmt must not be null";

end;

dyn_req_char must_not_equal dig_req_char";
  error:"the dyn_req_char must be different than the dig_req_char";

if min_psswd_chg is unknown
then max_psswd_chg must_be unknown
  and dsable_attempt must_be unknown
  error: "min_psswd_chg cannot be null unless password enhancement
         is turned off";

if max_psswd_chg is unknown
then min_psswd_chg must_be unknown
  error: "max_psswd_chg cannot be null unless password enhancement
         is turned off";

if dsable_attempt is unknown
then min_psswd_chg must_be unknown
  and max_psswd_chg must_be unknown
  error: "dsable_attempt cannot be null unless password enhancement
         is turned off";

max_psswd_chg must_be_less_than_min_psswd_chg
  error."max_psswd_chg cannot be less than min_psswd_chg";
```

ecdorg

This form is the starting point in accessing information in the Equipment Configuration Database (ECD). It specifies information about the relationship of various devices.

Form Layout

```
Header Information For Database          ecdorg      (1/3)
                                         (Recent Change and Verify)

1.kecdorg:_____          2.model:_____

3.cu0          complex_name  complex_number  unit_name  unit_number
4)_____     5)_____     6)_____     7)_____

8.cu1          9)_____     10)_____     11)_____     12)_____

13.cu0_mas0   14)_____     15)_____     16)_____     17)_____

18.cu0_mas1   19)_____     20)_____     21)_____     22)_____

23.cu1_mas0   24)_____     25)_____     26)_____     27)_____
```

```
                                         ecdorg      (2/3)

28.cu1_mas1   complex_name  complex_number  unit_name  unit_number
29)_____     30)_____     31)_____     32)_____

33.cache0     34)_____     35)_____     36)_____     37)_____

38.cache1     39)_____     40)_____     41)_____     42)_____

43.bdev0      44)_____     45)_____     46)_____     47)_____

48.bdev1      49)_____     50)_____     51)_____     52)_____
```

ecdorg (3/3)

root and swap devices

53.root_log_name: _____ 54.root_dcn: _____ 55.root_part: _____

56.swap_log_name: _____ 57.swap_dcn: _____ 58.swap_part: _____

59.limp_mode: _____

Definition of Fields

1. kecdorg: (KECDORG) Required

This is the key field.

Legal values: ECDORG

Default value: ECDORG

2. model: (MODEL) Required

3B computer model type.

Legal values: 3B20D, 3B20DCR, 3B20DNS, 3B21D

3. cu0: (CU0)

Link to ucb for cu0; cu0 is a structure which includes:

4) cu0.complex_name: (*CU0.COMPLEXNAME)

First part of four-part ucb identifier.

5) cu0.complex_number: (*CU0.COMPLEXNUMBER)

Second part of four-part ucb identifier.

6) cu0.unit_name: (*CU0.UNITNAME)

Third part of four-part ucb identifier.

7) cu0.unit_number: (*CU0.UNITNUMBER)

Fourth part of four-part ucb identifier.

8. cu1: (CU1)

Link to ucb for CU1; CU1 is a structure which includes the following:

9) See 4 – 7 in field cu0.

13. cu0_mas0: (CU0MAS0)

Links to mass storage controllers for CU0.

14) See 4 – 7 in field cu0.

18. cu0_mas1: (CU0MAS1)

Links to mass storage controllers for CU0.

19) See 4 – 7 in field cu0.

23. cu1_mas0: (CU1MAS0)

Links to mass storage controllers for CU1.

24) See 4 – 7 in field cu0.

28. cu1_mas1: (CU1MAS1)

Links to mass storage controllers for CU1.

29) See 4 – 7 in field cu0.

33. cache0: (CACHE0)

Link to cache for CU0.

34) See 4 – 7 in field cu0.

38. cache1: (CACHE1)

Link to cache for CU1.

39) See 4 – 7 in field cu0.

43. bdev0: (BDEV0)

Identifies the boot device Moving Head Disk (MHD) on Disk File Controller (DFC) 0.

44) bdev0 is a structure (see field numbers 4 – 7 of this form).

48. bdev1: (BDEV1)

Identifies the boot device (MHD) on DFC 1.

49) bdev1 is a structure (see field numbers 4 – 7 of this form).

53. root_log_name: (ROOTLOGNAME)

Logical name of root device.

Legal values: /dev/..., /dgn/..., null

54. root_dcn: (ROOTDCN) Required

Root device control number.

Legal values: 0 – 32767, 0x0 – 0x7FFF

Default value: 4

55. root_part: (ROOTPART) Required

Root partition number.

Legal values: 0 – 32767, 0x0 – 0x7FFF

Default value: 4

56. swap_log_name: (SWAPLOGNAME)

Logical name of swap device.

Legal values: /dev/..., /dgn/..., null

57. swap_dcn: (SWAPDCN) Required

Swap device control number.

Legal values: 0 – 32767, 0x0 – 0x7FFF

Default value: 4

58. swap_part: (SWAPPART) Required

Swap partition number.

Legal values: 0 – 32767, 0x0 – 0x7FFF

Default value: 6

59. limp_mode: (LIMPMODE)

Disk Independent Operation (DIOP) feature level.

Legal values: NODIOP – DIOP is not enabled
 UCLDIOP – DIOP is enabled

Default value: NODIOP

errcount

This review-only form holds the current error counts for a particular *ucb*. The *errcount* and *errthld* forms are used by the **kconfig** library to determine the marking of devices as faulty. The *errcount* form keeps the actual count of errors within a particular interval of time. The *errthld* forms keep the limits, both time interval and error count, to be reached before a specific device is marked faulty.

Form Layout

				errcount	(1/3)			
Error Count For The UCB				(Review Only)				
1.complex_name: _____		2.complex_number: ____						
3.unit_name: _____		4.unit_number: ____						
5.ecflt: _____		6.ecnum: ____						
7.errcount_max								
short	long	s_time	l_time	short	long	s_time	l_time	
1) _____	_____	_____	_____	6) _____	_____	_____	_____	
2) _____	_____	_____	_____	7) _____	_____	_____	_____	
3) _____	_____	_____	_____	8) _____	_____	_____	_____	
4) _____	_____	_____	_____	9) _____	_____	_____	_____	
5) _____	_____	_____	_____	10) _____	_____	_____	_____	

7.errcount_max(cont)				errcount (2/3)			
short	long	s_time	l_time	short	long	s_time	l_time
11) _____	_____	_____	_____	23) _____	_____	_____	_____
12) _____	_____	_____	_____	24) _____	_____	_____	_____
13) _____	_____	_____	_____	25) _____	_____	_____	_____
14) _____	_____	_____	_____	26) _____	_____	_____	_____
15) _____	_____	_____	_____	27) _____	_____	_____	_____
16) _____	_____	_____	_____	28) _____	_____	_____	_____
17) _____	_____	_____	_____	29) _____	_____	_____	_____
18) _____	_____	_____	_____	30) _____	_____	_____	_____
19) _____	_____	_____	_____	31) _____	_____	_____	_____
20) _____	_____	_____	_____	32) _____	_____	_____	_____
21) _____	_____	_____	_____	33) _____	_____	_____	_____
22) _____	_____	_____	_____	34) _____	_____	_____	_____

7.errcount_max(cont)				errcount (3/3)			
short	long	s_time	l_time	short	long	s_time	l_time
35) _____	_____	_____	_____	45) _____	_____	_____	_____
36) _____	_____	_____	_____	46) _____	_____	_____	_____
37) _____	_____	_____	_____	47) _____	_____	_____	_____
38) _____	_____	_____	_____	48) _____	_____	_____	_____
39) _____	_____	_____	_____	49) _____	_____	_____	_____
40) _____	_____	_____	_____	50) _____	_____	_____	_____
41) _____	_____	_____	_____	51) _____	_____	_____	_____
42) _____	_____	_____	_____	52) _____	_____	_____	_____
43) _____	_____	_____	_____	53) _____	_____	_____	_____
44) _____	_____	_____	_____	54) _____	_____	_____	_____

Definition of Fields

1. complex_name: (COMPLEXNAME) Optional

First part of a four-part **ucb** identifier. This is a key field.

2. complex_number: (COMPLEXNUMBER) Optional

Second part of a four-part **ucb** identifier. This is a key field.

3. unit_name: (UNITNAME) Required

Third part of a four-part **ucb** identifier. This is a key field.

4. unit_number: (UNITNUMBER) Required

Fourth part of a four-part **ucb** identifier. This is a key field.

5. ecflt:

Faulty flag for use by configuration control (CONFIG). Indicates the time the device went faulty.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

6. ecnun:

Number of error counters used.

Legal values: 0 – 255, HEX, null

7. errcount_max

Error counts, short and long, indicate the number of hardware faults this ucb has suffered. Time stamps, **s_time** and **l_time**, indicate the time stamp value when the short and long error counts were last initialized to zero. This field represents an array of 54 short- and long-term counters.

errcount_max.short

Number of errors since short-term time stamp. If the short-term counter reaches the threshold, this unit is marked faulty.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

errcount_max.long

Number of errors since long-term time stamp. If the long-term counter reaches the threshold, this unit is marked faulty.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

errcount_max.s_time

Time since error collection started for short-term threshold.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

errcount_max.l_time

Time when error collection started for long-term threshold.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

errthld

This form associates hardware error numbers with error counters. It also specifies thresholds to be used with error counters. The numbers supplied for grp_errval are the numbers of the counter and thresholds (one-to-one correspondence for counters and thresholds). The hardware error number obtains the position in grp_errval.

Form Layout

Error Threshold Values		errthld	(1/3)
			(Recent Change and Verify)
1.error_table: _____			
2.short_timer: _____		3.long_timer: _____	
4.err_values			
errval1 errval2 errval3 errval4 errval5 errval6 errval7 errval8 errval9			
1)	_____	_____	_____
2)	_____	_____	_____
3)	_____	_____	_____
4)	_____	_____	_____
5)	_____	_____	_____
6)	_____	_____	_____

14.thr_table		errthld (2/3)						
	sthr	lthr	sthr	lthr	sthr	lthr		
1)	_____	_____	14)	_____	_____	27)	_____	_____
2)	_____	_____	15)	_____	_____	28)	_____	_____
3)	_____	_____	16)	_____	_____	29)	_____	_____
4)	_____	_____	17)	_____	_____	30)	_____	_____
5)	_____	_____	18)	_____	_____	31)	_____	_____
6)	_____	_____	19)	_____	_____	32)	_____	_____
7)	_____	_____	20)	_____	_____	33)	_____	_____
8)	_____	_____	21)	_____	_____	34)	_____	_____
9)	_____	_____	22)	_____	_____	35)	_____	_____
10)	_____	_____	23)	_____	_____	36)	_____	_____
11)	_____	_____	24)	_____	_____	37)	_____	_____
12)	_____	_____	25)	_____	_____	38)	_____	_____
13)	_____	_____	26)	_____	_____	39)	_____	_____

14.thr_table(cont)		errthld (3/3)						
	sthr	lthr	sthr	lthr	sthr	lthr		
40)	_____	_____	45)	_____	_____	50)	_____	_____
41)	_____	_____	46)	_____	_____	51)	_____	_____
42)	_____	_____	47)	_____	_____	52)	_____	_____
43)	_____	_____	48)	_____	_____	53)	_____	_____
44)	_____	_____	49)	_____	_____	54)	_____	_____

⇒ NOTE:

If updates are made to both the *errthld* and *ucb* forms in the same transaction block, the *errthld* update **must** be done first.

Definition of Fields

1. error_table: (ERRORTABLE) Required

Error threshold record name. The symbolic name representing this error threshold table instance. This is a key field.

Legal values: alphanumeric characters

2. short_timer: (SHORTTIMER) Required

Short-term timer for error threshold. This indicates the time interval to initialize the short-term counter to zero.

Legal values: 0 – MAXINT, 0x0 – 0x7FFFFFFF

3. long_timer: (LONGTIMER) Required

Long-term timer for error threshold. This indicates the time interval to initialize the long-term counter to zero.

Legal values: 0 – MAXINT, 0x0 – 0x7FFFFFFF

4. err_values.errval1 — err_values.errval9:
(*ERRVALUES.ERRVAL1[ROW] — ERRVALUES.ERRVAL9[ROW])

Number of threshold to be associated with this hardware error. This field is an array of a maximum of 54 error types. The values in this field are used to index into `errcount_max` field in the `errcount` record. Error types may share the same error thresholds.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

14. thr_table: (THRTABLE)

Threshold table. Indicates threshold values for long- and short-term counters. This field is an array of 54 error threshold values. For each error counter, there are corresponding short and long error thresholds.

thr_table.sthr (*THRTABLE.STHR[ROW])

Maximum number of errors to be tolerated in short-time interval.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

thr_table.lthr (*THRTABLE.LTHR[ROW])

Maximum number of errors to be tolerated in long-time interval.

Legal values: 0 – 65534, 0x0 – 0xFFFFE, null

ftaminopt

This form is used to enter keys to one or more *ftamuser* records. It is populated for File Transfer Access and Management (FTAM) applications that will act as an initiator. Additional information about the remote applications with which these applications want to communicate is defined. Multiple applications may share the same *ftaminopt* record.

Form Layout

```
ftaminopt          (1/1)
FTAM Initiator Option Record  (Recent Change and Verify)

1.ftaminoptname: _____

2.remote_application_information:

1)_____  5)_____  9)_____  13)_____
2)_____  6)_____  10)_____ 14)_____
3)_____  7)_____  11)_____ 15)_____
4)_____  8)_____  12)_____ 16)_____
```

Definition of Fields

1.ftaminoptname: (FTAMINOPTNAME) Required

The FTAM initiator option name.

Legal values: 1 – 8 characters

Default value: no default

2.remote_application_information: (REMOTEAAPPLICATIONINFORMATION)

Sixteen occurrences of a character string representing the key to an *ftamuser* record.

Legal values: 1 – 8 characters

Default value: null (blank)

⇒ NOTE:

There must be at least one occurrence of an *ftamuser* record associated with an *ftaminopt* record.

Form Checks

There must be at least one entry in the remote_application_information field.

ftamopt

This form is used to define File Transfer Access and Management (FTAM) protocol-specific information necessary for certain applications. The data items stored in this form are accessed on each FTAM association establishment.

Form Layout

FTAM Option Record	ftamopt	(1/1) (Recent Change and Verify)
1.ftamoptname: _____	2.fsl_version: _____	
3.implementation_info: _____		
4.ftam_qos: _____	5.ckpt_window: _____	
6.pres_req: _____	7.ctxt_name: _____	
8.ftaminoptname: _____	9.fpdu_buffer_size: _____	

Definition of Fields

1.ftamoptname: (FTAMOPTNAME) Required

The option name pertaining to the application specific FTAM information. The record is accessed in Recent Change and Verify (RC/V) by way of the ftamoptname field. However, at run time of the Data Transfer Process (DTP), the value for the key to the *ftamopt* form is passed from the *gfscom* record. This is a key field.

Legal values: 1 – 8 characters

Default value: no default

2.fsl_version: (FSLVERSION)

The FTAM Service Library (FSL) version to be used. For example, this field may be used to interwork with systems that have different levels of defect reports included.

Legal values: VER1 (indicates FSL version 1)

Default value: VER1

3.implementation_info: (IMPLEMENTATIONINFO)

This is the implementation information field. It is the indicator of the level of FTAM profile support. This string is exchanged on an F-INITIALIZE for information only.

Legal values: 0 – 64 printable characters

Default value: env41204(rev);AFT11

4.ftam_qos: (FTAMQOS)

The level of error susceptibility. The value of this field is dependent on the FTAM functional units required for the application which are specified in the General File Server (GFS) Equipment Configuration Database (ECD) records. If functional unit restart or recovery is required, the ftam_qos field should be at least 1. Class 1 recovery supports recovery from damaged data transfer regime only. Class 2 recovery means recovery from select or open regimes. Class 3 recovery means recovery from loss of an association. The value of this field must not be NORECOV if the functional unit restart or recovery (fields 26 and 27 on the *gfscom* form) is required.

Legal values:	NORECOV	No Recovery
	CLASS1	Class 1 Recovery
	CLASS2	Class 2 Recovery
	CLASS3	Class 3 Recovery

Default value: NORECOV

5.ckpt_window: (CKPTWINDOW)

The maximum checkpoint window supported. This value is negotiated on the F_INITIALIZE. This field depends on the availability of the restart functional unit (on *gfscom* record) which depends on the session minor synchronize and resynchronize functional units.

Legal values: 1 – 127

Default value: 1

6.pres_req: (PRESREQ)

The presentation requirements. There are three special values: kernel, context management, and context management and restoration.

Legal values: KERNEL, CTXTMGT (context management), or CTXTMRST (context management and restoration)

Default value: KERNEL

7.ctxt_name: (CXTXTNAME) Required

The application context name of the FTAM dialogue. This field provides the key to an *objectid* record instance.

Legal values: 1 – 8 characters

Default value: FTAMCXTXT

8.ftaminoptname: (FTAMINOPTNAME)

The name of the *ftaminopt* record associated with this application. This field is used when an application wants to act as an initiator.

Legal values: 0 – 8 characters

Default value: null (blank)

9.fpdubuffer_size: (FPDUBUFFERSIZE)

The buffer size for sending and receiving FPDUs (FTAM protocol data units) in 1K increments. This is used primarily for control packets.

Legal values: 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9216, or 10240

Default value: 7168 (7K)

Form Checks

The following checks are performed on this form:

```
/*
The following cross-checks are being performed since the
underlying protocol does not support any other options
than these. Support of other values will require FTAM
protocol development.
*/

(ftam_qos = "NORECOV")
    error: "Only 'No Recovery' quality of service is supported";
(ckpt_window = 1)
    error: "Only a checkpoint window of 1 is supported";
(pres_req = "KERNEL")
    error: "Only 'KERNEL' is supported for the presentation
requirements";
(ctxt_name = "FTAMCXTXT")
    error: "Only context_name 'FTAMCXTXT' with object_identifiers
1,0,8571,1,1 is supported";
```

ftamuser

This form is used to define FTAM protocol-specific information that is required only for local applications that will act as an initiator. This form provides information about remote applications with which a local application wants to communicate.

Form Layout

```
ftamuser      (1/3)
FTAM User Record      (Recent Change and Verify)

1.ftamusername: _____

remote_application_entity_title:
  2.rmt_appl_title: _____      3.rmt_aeq: _____

4.rmt_psap_sel      rmt1  rmt2  rmt3  rmt4
      _      _      _      _

9.rmt_ssap_sel: _____

10.rmt_tsap_addr: _____
```

```
ftamuser      (2/3)

11.qos_priority: _____

12.qos_max_out: _____      13.qos_avg_out: _____

14.qos_max_in: _____      15.qos_avg_in: _____

remote_vfs_password:
  16.vfs_pwd_chk: _____      17.vfs_pwd_type: _

18.vfs_pwd_octet_option
oct1  oct2  oct3  oct4  oct5  oct6  oct7  oct8
_      _      _      _      _      _      _      _

oct9  oct10  oct11  oct12  oct13  oct14  oct15  oct16
_      _      _      _      _      _      _      _

35.vfs_str_opt: _____
```

```
ftamuser      (3/3)
remote_creation_password:
 36.crt_pwd_chk:_          37.crt_pwd_type:_
 38.crt_pwd_octet_option

oct1 oct2 oct3 oct4 oct5 oct6 oct7 oct8
_ _ _ _ _ _ _ _
oct9 oct10 oct11 oct12 oct13 oct14 oct15 oct16
_ _ _ _ _ _ _ _
55.crt_str_opt:_____
```

Definition of Fields

1.ftamusername: (FTAMUSERNAME) Required

The FTAM user name that is specified on the *ftaminopt* record for a particular remote application.

Legal values: 1 – 8 characters

Default value: no default

2.rmt_appl_title: (RMTAPPLTITLE)

The remote application process title. This specifies the key to an *objectid* record. This field cannot be null (blank).

Legal values: 1 – 8 characters

Default value: no default

3.rmt_aeq: (RMTAEQ)

The remote application entity qualifier. The combination of the *rmt_appl_title* and the *rmt_aeq* fields is the remote application entity title.

Legal values: 0 – (MAXINT-1) (2147483646)

Default value: 0

4.rmt_psap_sel.rmt1 – rmt_psap_sel.rmt4:
(RMTPSAPSEL.RMT1– RMTPSAPSEL.RMT4)

The presentation selector of the remote entity. This is an array (1 to 4 hexadecimal characters) that specifies the presentation selector of the remote entity. A blank field in the middle of the address translates to 0.

Legal values: 1 – 4 hexadecimal characters, or null (blank)

Default value: null (blank)

9.rmt_ssap_sel: (RMTSSAPSEL)

The session selector of the remote entity. The value of this field must be present on an existing *tsapinfo* form.

Legal values: 1 – 16 printable characters, or null (blank)

Default value: null (blank)

10.rmt_tsap_addr: (RMTTSAPADDR) Required

The transport address of the remote entity. The value of this field must be present on an existing *tsapinfo* form.

Legal values: 1 – 44 decimal characters

Default value: no default

11.qos_priority: (QOSPRRIORITY) Required

The quality of service priority for the connection.

Legal values: 0 – 64

Default value: 64

12.qos_max_out: (QOSMAXOUT)

The quality of service maximum output value.

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000, or null (blank)

Default value: 75

13.qos_avg_out: (QOSAVGOUT)

The quality of service average acceptable output value.

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000, or null (blank)

Default value: 75

14.qos_max_in: (QOSMAXIN)

The quality of service maximum input value.

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000,
or null (blank)

Default value: 75

15.qos_avg_in: (QOSAVGIN)

The quality of service average acceptable input value.

Legal values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000,
or null (blank)

Default value: 75

16.vfs_pwd_chk: (VFSPWDNULL)

Specifies if the remote Virtual File Store (VFS) password is checked.

Legal values: y (yes) or n (no)

Default value: y

17.vfs_pwd_type: (VFSPWDTYPE)

This field specifies what type of remote VFS password is used.

Legal values: o (octet string password), s (graphic string password), or n
(vfs_pwd_type is not used)

Default value: n

18.vfs_pwd_octet_option.oct1 – vfs_pwd_octet_option.oct16:
(VFSOCTOPT.OCT1 – VFSOCTOPT.OCT16)

This is an array of 1 to 16 hexadecimal characters specifying the remote
VFS password in octet format.

Legal values: 1 – 2 hexadecimal characters, or null (blank)

Default value: null (blank)

35.vfs_str_opt: (RMTVFSPWD)

The password for the remote VFS password string.

Legal values: 0 – 16 alphanumeric characters

Default value: null string

36.crt_pwd_chk: (CRTPWDNULL)

Specifies if the remote file creation password is checked.

Legal values: y (yes) or n (no)

Default value: y

37.crt_pwd_type: (CRTPWDTYPE)

This field specifies what type of remote file creation password is used.

Legal values: o (octet string password)
 s (graphic string password)
 n (cert_pwd_type field is not used)

Default value: n

38.crt_pwd_octet_option.oct1 – crt_pwd_octet_option.oct16:
(CRTOCTOPT.OPT1 – CRTOCTOPT.OCT16)

This is an array of 1 to 16 hexadecimal characters specifying the remote file creation password in octet format.

Legal values: 1 – 2 hexadecimal characters, or null (blank)

Default value: null (blank)

55.crt_str_opt: (RMTCRTPWD)

The password for the remote file creation password string.

Legal values: 0 – 16 alphanumeric characters

Default value: null string

Form Checks

The following form checks are performed on this form:

```
case (vfs_pwd_null);
  when {"n"}: do;
    vfs_pwd_type must_be "n"
    error: "When vfs_pwd_chk is 'n', remote password must be unknown";

    vfs_oct_opt.oct1 must_be unknown
    error: "When vfs_pwd_chk is 'n', octet password must be unknown";
  end;

  when {"y"}: do;
    vfs_pwd_type must_not_be "n"
    error: "When vfs_pwd_chk is set to 'y', the VFS password must be 's' or 'o'";
  end;
end;

case (crt_pwd_null)
```

```
when {"n"}: do;
  crt_pwd_type must_be "n"
  error: "When crt_pwd_chk is 'n', create password must be unknown";
  crt_opt.opt1 must_be unknown
  error: "When crt_pwd_chk is 'n', octet password must be unknown";
end;

when {"y"}: do;
  crt_pwd_type must_not_be "n"
  error: "When crt_pwd_chk is set to 'y', the create password type must be 's' or
  'o'";
end;
end;

case (vfs_pwd_type);
when {"o"}: do;
  rmt_vfs_pwd must_be unknown
  error: "When VFS password type is 'o', remote VFS password must be unknown";
  vfs_opt.opt1 must_be known
  error: "When VFS password type is 'o', octet password must be known";
end;

when {"s"}: do;
  vfs_opt.opt1 must_be unknown
  error: "When VFS password type is 's', octet password must be unknown";
end;

when {"n"}: do;
  rmt_vfs_pwd must_be unknown
  error: "When VFS password type is 'n', remote VFS password must be unknown";
  vfs_opt.opt1 must_be unknown
  error: "When VFS password type is 'n', octet password must be unknown";
end;
end;

case (crt_pwd_type)
when {"o"}: do;
  rmt_crt_pwd must_be unknown
  error: "When create password type is 'o', remote create password must be
  unknown";

  crt_opt.opt1 must_be known
  error: "When crt_pwd_type is 'o', octet password must be known";
end;

when {"s"}: do;
  crt_opt.opt1 must_be unknown
  error: "When create password type is 's', octet password must be unknown";
  rmt_crt_pwd must_be known
  error: "When create password type is 's', remote create password must be known";
end;

when {"n"}: do;
  rmt_crt_pwd must_be unknown
  error: "When create password type is 'n,
remote create password must be unknown";
  crt_opt.opt1 must_be unknown
  error: "When create password type is 'n', octet password must be unknown";
end;
end;
```

getty

This form specifies the shell that is to be displayed on a particular device.

Form Layout

Getty Record	getty	(1/1)	
		(Recent Change and Verify)	
1.gettyrec:_____	2.gettyname:_____		
3.getty_dir:_____			
4.shlname:_____			
5.spl:_	6.prompt:_	7.pagename:_____	8.portid:___
9.dlg_tmr:___	10.auth_chk:___	11.cmd_log_	12.split_scrn:___
13.paging_timer:___			

Definition of Fields

1. gettyrec: (GETTYREC) Required

Getty record id (getty and channel id). This is a key field.

Legal values: printable characters

2. gettyname: (GETTYNAME) Required

Getty name (for example: pdsgetty, cdgetty, splgetty, getty, shlgetty, ciagetty).

Legal values: printable characters

3. getty_dir: (GETTYDIR)

Getty directory. Directory to execute shell.

Legal values: printable characters, null

⇒ NOTE:

Should not be a mount point other than “/”.

4. shlname: (SHLNAME)

Shell name.

Legal values: printable, null

5. spl: (SPL)

Spooler.

Legal values: y, n, null

Default value: y

6. prompt: (PROMPT) Required

Shell prompt.

Legal values: y, n

Default value: n

7. pagename: (PAGENAME)

Name of page to be displayed on the control display terminal after a system boot or craft initialization.

Legal values: alphanumeric or null

8. portid: (PORTID)

Port identification number of page to be displayed.

Legal values: 0 – 255 or null [Null will be mapped to port 18 (PT_NONE) in the database, and port 18 will appear as null when the form is reviewed.]

9. dlg_tmr: (DLGTMR) Required

Dialogue time-out value. A value of 0 means that the dialogue session will never time out.

Legal values: 0 – 30 minutes

Default value: 0

10. auth_chk: (AUTHCHK) Required

Defines the type of authority check to be performed by a dialogue shell invoked by this *getty* form, either personal (p), terminal (t), passcard authority check (c), dynamic authority check (d), or none (n). If personal authority checking is specified, a password will have to be entered to initiate a dialogue session. (Refer to AUD:PATH input message in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.)

Legal values: p, t, c, d, u, or n

Default value: n

11. cmd_log: (CMDLOG)

Specifies whether or not input commands will be logged by the dialogue shell invoked by this *getty* form.

Legal values: y, n, or null

Default value: n

12. split_scrn: (SPLITSCRN)

Specifies whether or not the dialogue shell invoked by this *getty* form will have a split-screen capability (that is, separate input and output windows).

Legal values: y, n, or null

Default value: n

13. paging_timer: (PAGINGTIMER) Required

If the split_scrn field (field 12) is set to "y," any output message larger than the terminal's output window will cause the dialogue shell to print one page of output and prompt the user for continuation; for example, "Continue (y/n) ?". This field specifies the length of time, in seconds, that the dialogue shell will wait before automatically continuing with the next page of output. (See *device* form, field 47, for information on defining the terminal's output window size.)

Legal values: 0 – 600 seconds

Default value: 120

Form Checks

```
if gettyname is "cdgetty"
  then if pagename is known
    then portid must_be known and portid must_not_be 18
    error: "if the cdgetty pagename is set, the portid must not be
      null or 18 (PT_NONE)";
  else portid must_be 18 or portid must_be unknown
  error: "if the cdgetty pagename is null, the portid must be
    null or 18 (PT_NONE)";
```

gfsappl

This form is used to provide information that is needed by General File Transfer (GFT) applications for population information and the defined filename space.

Form Layout

gfsappl (1/4)		
GFS Application Element Definition (Recent Change and Verify)		
1.applname: _____	2.vfs_id: _____	3.max_files: _____
4.max_file_size: _____	5.max_sessions: _____	6.max_dupl_age: _____
7.max_ids: _____	8.max_dtp: _____	9.max_dialogues: _____
10.appl_status: _____	11.appl_importance: _____	12.ap_type: _____
13.ap_subtype: _____	14.auto_deletion: _____	15.ready_deletable: _____
16.max_robustness: _____	17.auto_file_gen: _____	18.real_time_xfer: _____

gfsappl (2/4)		
19.path_name_prefix	pathname_prefix	
1) _____	9) _____	
2) _____	10) _____	
3) _____	11) _____	
4) _____	12) _____	
5) _____	13) _____	
6) _____	14) _____	
7) _____	15) _____	
8) _____		
24.warning_thresh: _____	25.critical_thresh: _____	26.emergency_thresh: _____
27.atd_thresh: _____	28.atd_active: _____	

gfsappl (3/4)

29.device_list: logical_devname

1) _____	12) _____	23) _____
2) _____	13) _____	24) _____
3) _____	14) _____	25) _____
4) _____	15) _____	26) _____
5) _____	16) _____	27) _____
6) _____	17) _____	28) _____
7) _____	18) _____	29) _____
8) _____	19) _____	30) _____
9) _____	20) _____	31) _____
10) _____	21) _____	32) _____
11) _____	22) _____	

gfsappl (4/4)

36.dtp_common_id: _____ 37.dtp_responder_id: _____

38.dtp_initiator_id: _____ 39.ftamusername: _____

40.adld_thresh: ____ 41.adld_active: _

42.alw_prefix_chnge: _ 43.max_fgn: _____

Definition of Fields

1.applname: (APPLNAME) Required

Internal application name. This is a key field.

Legal values: 1 – 8 characters

Default value: no default

2.vfs_id: (VFSID)

Required field: This field is required if the field dpt_responder_id field has a non-null value [that is, this application acts as a General File Server (GFS) responder]. When the value is non-null, the value must exist as an attach_id field on some *pphuser* form.

Virtual file store identifier.

Legal values: 1 – MAXINT or null (blank)

Default value: no default

3.max_files: (MAXFILES)

The maximum number of files to be supported by the application.

Legal values: 1 – 4500

Default value: no default

4.max_file_size: (MAXFILESIZE)

Specifies the number of bytes of disk space allocated by monitor for the data portion of a new file (must be a multiple of 512 bytes). The real file consists of a variable size data portion and a fixed size file header.

Legal values: 2048 – 33552896

Default value: no default

5.max_sessions: (MAXSESSIONS)

The maximum number of completed session records managed by the application.

Legal values: 1 – 32

Default value: 1

6.max_dupl_age: (MAXDUPLAGE)

The maximum time a file is used for duplicate file checking (in hours). A value of 0 indicates duplicate file checking will not be done.

Legal values: 0 – 255

Default value: 0

7.max_ids: (MAXIDS)

The maximum number of file identifier records managed.

Legal values: 1 – 20

Default value: 4

8.max_dtp: (MAXDTP)

The maximum number of simultaneous data transfer processes to allow.

Legal values: 1 – 16

Default value: 1

9.max_dialogues: (MAXDIALOGUES)

The number of simultaneous files that are allowed to be populated.

Legal values: 0 – 15

Default value: no default

10.appl_status: (APPLSTATUS)

Specifies whether this application can use the file system. This allows for temporarily disabling the application's use of the file system when the file system must be taken down for repair.

Legal values: e (enabled) or d (disabled)

Default value: no default

11.appl_importance: (APPLIMPORTANCE)

How critical is the information to the customer (used in determining what data should be purged when there is insufficient memory). When updating this field from "noncritical" to "critical," there is a possibility that future files allocated may not be restored correctly after a GFS monitor reboot or background audit. The prior actions occur if the internal GFS database is destroyed. Once this field has been set, it should not be changed.

Legal values: c (critical) or n (noncritical)

Default value: no default

12.ap_type: (APTYPE) Required

The type of population which will be used by the application. Stream means that GFS will transfer data basically as it is and in the order in which it is generated. Explicit population means that the GFS takes care of space allocation, file creation, actual transfer of data to disk, and monitoring when the file is full. Explicit attachment means that GFS registers existing files and performs some partial management. Implicit attachment means that GFS registers a node (any files under a node are available), but no management or security services are provided. Non-GFS is used to denote an application that uses this form for the logical device list but is not administered by GFS.

Legal values:

stream	
ep	(explicit population)
ea	(explicit attachment)
ia	(implicit attachment)
nongfs	(non-GFS)

Default value: no default

13.ap_subtype: (APSUBTYPE) Required

The type of storage that is used by the application.

Legal values: fp (file partition) or fs (file system)

Default value: fp

14.auto_deletion: (AUTODELETION)

Specifies if files in the "safe" are automatically deleted.

Legal values: y (yes) or n (no)

Default value: no default

15.ready_deletable: (READYDELETABLE)

Specifies if existing files in the "ready" are state deletable.

Legal values: y (yes) or n (no)

Default value: no default

16.max_robustness: (MAXROBUSTNESS)

Specifies if files should be protected from accidental deletion.

Legal values: y (yes) or n (no)

Default value: no default

17.auto_file_gen: (AUTOFILEGEN)

Specifies if there should be automatic generation of file numbers to append to filenames for explicitly populated file partitions.

Legal values: y (yes) or n (no)

Default value: no default

18.real_time_xfer: (REALTIMEXFER)

Specifies if data should be transferred immediately upon generation.

Legal values: y (yes) or n (no)

Default value: no default

19.path_name_prefix: (PATHNAMEPREFIX)

For explicitly populated file systems, this is the mount point. For implicit attachment, this is the base directory.

Legal values: 0 – 32 characters

Default value: no default

24.warning_thresh: (WARNINGTHRESH)

Threshold percentage of storage and/or inodes used which triggers a warning message to the technician. A value of zero means that this checking is disabled.

Legal values: 0 – 99

Default value: no default

25.critical_thresh: (CRITICALTHRESH)

Threshold percentage of storage and/or inodes used which triggers a critical message to the technician. A value of zero means that this checking is disabled.

Legal values: 0 – 99

Default value: no default

26.emergency_thresh: (EMERGENCYTHRESH)

Threshold percentage of storage and/or inodes used which triggers an emergency message to the technician. A value of zero means that this checking is disabled.

Legal values: 0 – 99

Default value: no default

27.atd_thresh: (ATDTHRESH)

Threshold percentage of storage and/or inodes used which triggers an automatic dump to tape of data. A value of zero means that this checking is disabled. When there is a non-zero value specified, then the *tape_pathname* attribute on the *gfsinit* form must be non-null.

Legal values: 0 – 99

Default value: no default

28.atd_active:(ATDACTIVE)

This field specifies whether automatic tape dump is active.

Legal values: y or n

Default value: n

29.device_list: (DEVICETHRESH)

List of 1 to 32 *logdev_names*. Used to specify the location of the file system or the locations for raw file partitions. The set of logical devices across all *gfsappl* forms must be unique (that is, a device may only be specified once).

Legal values: /dev/ followed by 1 – 15 characters

Default value: no default

36.dtp_common_id: (DTPCOMMONID)

Specifies the name linking the form which contains the application information.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

37.dtp_responder_id: (DTPRESPONDERID)

Specifies the name linking the form which contains the application information associated with a responder data link Data Transfer Process (DTP).

Legal values: 1 – 8 alphanumeric characters

Default value: no default

38.dtp_initiator_id: (DTPINITIATORID)

The name linking the form which contains the application information associated with tape DTPs and data link initiators.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

39.ftamusername:(FTAMUSERNAME)

This field specifies the *ftamuser* record associated with the automatic data link.

Legal Values: alphanumeric, unknown

Default value: unknown

40.adld_thresh:(ADLDTHRESH)

This field specifies the percentage of disk space for the Automatic Data Link Dump.

Legal values: 0 – 99, unknown

Default value: 0

41.adld_active:(ADLDACTIVE)

This field specifies if the Automatic Data Link Dump is active.

Legal Values: y or n

Default value: n

42.alw_prefix_chnge:(ALWPREFIXCHNGE)

This field specifies if a user is allowed to change the *pathname_prefix* field. If this field is set to "n," the Recent Change and Verify (RC/V) code will not allow the user to change the *pathname_prefix* field. If the field is set to "y," changes to the *pathname_prefix* field are allowed. However, the RC/V code will not allow duplicate or overlapping *pathname_prefix* changes. For example, if */usr* already exists as a *pathname_prefix*, the RC/V code will not allow */usr/tmp* to be entered as a *pathname_prefix*. Once the field is set to "n," the RC/V code prevents the user from changing it to "y."

Legal values: y or n

Default value: y

43.max_fgn:(MAXFGN)

This field specifies the maximum file generation number.

Legal values: 0 – 9999

Default value: 9999

Form Checks

The following checks are required for this form:

The ap_type “nongfs” must have an ap_subtype of “fp” because this is used for Automatic Message Accounting (AMA). There must also be at least one logical device specified. All other fields are to be ignored for “nongfs.” The following table shows attributes that are meaningful only for certain values of ap_type and ap_subtype. “X” denotes that the attribute in the left-hand column is meaningful only for the given ap_type/ap_subtype pairing.

Attribute	Valid for ap_type/ap_subtype pairing					
	st/fp*	st/fs†	ep/fp‡	ep/fs§	ea/fs**	ia/fs††
ap_status		X		X	X	X
auto_deletion_of_safe	X	X	X	X	X	
automatic_file_generated_numbers			X	X		
max_filesize	X	X	X	X		
max_num_dialogue	X	X	X	X		
max_robustness		X		X	X	X
ready_deletable	X	X	X	X		
warning_threshold	X	X	X	X		
emergency_threshold	X	X	X	X		
critical_threshold	X	X	X	X		
automatic_tape_dump_threshold	X	X	X	X		
* stream/file partition. † stream/file system. ‡ explicit population/file partition. § explicit population/file system. ** explicit attachment/file system. †† implicit attachment/file system.						

In addition, the following are required:

- The attribute `vfs_id` must be specified, if either of the link fields `dtp_responder` or `dtp_initiator` are non-null on this form. In the case of a `dtp_initiator`, there must be a data link process.
- `Auto_file_gen_num` must be set to "y" if the `ap_subtype` is set to "stream."
- `Pathname_prefix` must be null (blank) if `ap_type` is set to "stream."
- `Pathname_prefix` must be null (blank) if `ap_type` is set to "ep" and `ap_subtype` is set to "fp."
- `Pathname_prefix` must be specified if `ap_type` is set to "ia" or "ea."
- For the non-zero valued attributes `warning_threshold`, `critical_threshold`, and `emergency_threshold`, the following relationship must exist: `warning_threshold < critical_threshold < emergency_threshold`.
- If `ap_subtype` is set to "fs," there must be only one `logdev_name` specified. This applies only when the `ap_type` is "nongfs," "ep," or "stream."
- If `ap_subtype` is set to "fs," the number of `logdev_names` specified must be 0. This applies only when the `ap_type` is "ea" or "ia."
- If `ap_subtype` is set to "fp," there must be at least one `logdev_name` specified. This applies only when the `ap_type` is "stream," "ep," or "nongfs."
- If `add_active` is "y" `ftamusername` must be known.
- If `atd_active` is "y" `dtp_initiator_id` must be known.
- If either `dtp_responder` or `dtp_initiator` is specified, the attribute `dtp_common` must be specified.
- If `ap_type` is set to "stream" or "ep" and `max_files` is known, then `(max_fgn + 1)` must be greater than or equal to `max_files`.
- If `ap_type` is set to "stream" or "ep" and `max_files` is unknown, then `max_fgn` must be 9999.

The following are further restrictions on attributes:

- `max_ids` must be 4.
- `ap_subtype` must be "fp" if `ap_type` is "ep."
- `ap_type` must be "stream," "ep," "nongfs," or "ea."
- If `appl_importance` is "c," `ready_deletable` must be "n."

gfscom

This form is used to provide additional General File Transfer (GFT) information not contained on the *gfsappl* form, such as content type, functionality supported by the application, timer values, and a link to the *ftamopt* record.

Form Layout

	gfscom	(1/4)
GFS Application Common Definition		(Recent Change and Verify)
1.dtp_common_id: _____		
2.default_order: __ 3.protocol_version: _____ 4.dl_optname: _____		
5.recovery_mode: __ 6.inactivity_timer: _____ 7.file_slct_timer: _____		
service_class	(y/n)	attr_group (y/n)
9)unconstrained:	-	15)storage: -
10)management:	-	16)security: -
11)transfer:	-	17)private: -
12)tran_and_mgmt:	-	
13)access:	-	

	gfscom	(2/4)	
func_units	(y/n)	fa_actions (y/n)	fadu_id_group (y/n)
19)read:	-	33)read: -	44)traversal: -
20)write:	-	34)insert: -	45)r_traversal: -
21)file_access:	-	35)replace: -	46)random: -
22)lim_f_mgmt:	-	36)extend: -	
23)enh_f_mgmt:	-	37)erase: -	
24)grouping:	-	38)read_attr: -	
25)fadu_locking:	-	39)chg_attr: -	
26)recovery:	-	40)delete_file: -	
27)r_data_xfer:	-	41)passthrough: -	
28)lim_fs_mgmt:	-	42)link: -	
29)enh_fs_mgmt:	-		
30)obj_manip:	-		
31)grp_manip:	-		

gfscom (3/4)					
document_type	selected	string_sig	string_len	char_set	eol
FTAM-1	48)___	49)____	50)_____	51)___	52)___
FTAM-2	54)___	55)____	56)_____	57)___	58)___
FTAM-3	60)___	61)____	62)_____		
FTAM-4	64)___	65)____	66)_____		

gfscom (4/4)	
content_type_list	
68.constraint_set:	69.abstract_syntax:
1)_____	_____
2)_____	_____
3)_____	_____
4)_____	_____
5)_____	_____
6)_____	_____
7)_____	_____
8)_____	_____

Definition of Fields

1.dtp_common_id: (DTPCOMMONID) Required

The name from the *gfsappl* form to which this information is linked. This is a key field.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

2.default_order: (DEFAULTORDER)

The default order in which data is transmitted.

Legal values: n (newest first) or o (oldest first)

Default value: no default

3.protocol_version: (PROTOCOLVERSION)

Specifies which protocol the data link uses. When set to "ftam88," the attributes digital_sig_key and encryption_key on the *gfsresp* form must be null.

Legal values: ftam88 or ftam89a

Default value: ftam88

4.dl_optname: (DLOPTNAME)

Specifies a link to the data link option record.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

5.recovery_mode: (RECOVERYMODE)

Specifies the recovery mode for data transfer operations.

Legal values: n (none), s (at_start_of_file), c (at_any_active_checkpoint)

Default value: s

6.inactivity_timer: (INACTIVITYTIMER)

Specifies the time in seconds that a Data Transfer Process (DTP) can exist without doing anything. A value of zero disables checking.

Legal values: 0 – 86400

Default value: 1800

7.file_slct_timer: (FILESLCTTIMER)

Specifies the time in seconds that a DTP can have a file selected without doing anything. A value of zero disables checking.

Legal values: 0 – 86400

Default value: 600

service_class:

There are five service classes three of which divide and group the 13 functional units. These three service classes are access, management, and transfer. The relationship between service class and functional unit is shown in the following table. The service class designations are as follows:

- T – Transfer service class
- A – Access service class
- M – Management service class
- TM – Transfer and management service class
- U – Unconstrained service class.

The last two service classes are super sets of the first three. The notations used within the table are as follows:

- M – This is a mandatory functional unit for this service class.
- O – This is an optional functional unit for this service class.
- N – Read, write, or both needs to be a functional unit for this service class.

Functional Unit	Service Classes				
	T	A	M	TM	U
Read	N	M		N	O
Write	N	M		N	O
File_Access		M			O
Limited_File_Mgmt	O	O	M	M	O
Enhanced_File_Mgmt	O	O	O	O	O
Grouping	M	O	M	M	O
FADU_Locking		O			O
Recovery	O	O		O	O
Restart_Data_Transfer	O	O		O	O
Limited_Filestore_Mgmt	O	O	O	O	O
Enhanced_Filestore_Mgmt	O	O	O	O	O
Object_Manipulation	O	O	O	O	O
Group_Manipulation	O	O	O	O	O

9.unconstrained: (UNCONSTRAINED)

Specifies if all functional units are available to be selected by an initiator.

Legal values: y or n

Default value: n

10.management: (MANAGEMENT)

Specifies if all functional units associated with the management service class are available to be selected by an initiator.

Legal values: y (yes) or n (no)

Default value: n

11.transfer: (TRANSFER)

Specifies if all functional units associated with the transfer service class are available to be selected by an initiator.

Legal values: y (yes) or n (no)

Default value: y

12.tran_and_mgmt: (TRANANDMGMT)

Specifies if all functional units associated with the transfer and management service class are available to be selected by an initiator.

Legal values: y (yes) or n (no)

Default value: n

13.access: (ACCESS)

Specifies if all functional units associated with the access service class are available to be selected by an initiator.

Legal values: y (yes) or n (no)

Default value: n

attr_group:

Groups various file attributes according to their functionality. The three groups are storage, security, and private.

15.storage: (STORAGE)

Specifies if read/write to the definitions of the file attributes (storage account, date and time of creation, date and time of last modification, date and time of last read access, date and time of last attribute modification, identity of creator, identity of last modifier, identity of last reader, identity of last attribute modifier, file availability, filesize, and future filesize) and access attributes (current account, current concurrency control, and current locking style) are allowed.

Legal values: y (yes) or n (no)

Default value: y

16.security: (SECURITY)

Specifies if read/write to the definitions of the file attributes (access control and legal qualifications) and the activity attributes (current access passwords and active legal qualification) are allowed.

Legal values: y (yes) or n (no)

Default value: n

17.private: (PRIVATE)

Specifies if read/write to an attribute that is outside the scope of Open System Interconnection (OSI) standardization is allowed.

Legal values: y (yes) or n (no)

Default value: n

func_units:

FTAM consists of many individually defined services. For ease of structuring these services into useful sets, they are grouped into 13 functional units. These functional units are read, write, file access, limited file management, enhanced file management, grouping, FADU locking, recovery, restart data transfer, limited filestore management, enhanced filestore management, object manipulation, and group manipulation.

19.read: (READ)

Specifies if support of the transfer of data from responder to initiator is provided.

Legal values: y (yes) or n (no)

Default value: y

20.write: (WRITE)

Specifies if support of the transfer of data from initiator to responder is provided.

Legal values: y (yes) or n (no)

Default value: n

21.file_access: (FILEACCESS)

Specifies if any FADU in the file access structure is allowed to be identified for file access and is allowed to be removed from the file access structure.

Legal values: y (yes) or n (no)

Default value: n

22.lim_f_mgmt: (LIMFMGMT)

Specifies if file management for the creation and deletion of files and for the interrogation of the attributes of a file is provided.

Legal values: y (yes) or n (no)

Default value: n

23.enh_f_mgmt: (ENHFMGMT)

Specifies if the capability to modify the attributes of a file is provided.

Legal values: y (yes) or n (no)

Default value: n

24.grouping: (GROUPING)

Specifies if several regimes are allowed to be established in one exchange by forming several independent primitives into a group for procedural purposes.

Legal values: y (yes) or n (no)

Default value: y

25.fadu_locking: (FADULOCKING)

Specifies if the invocation of concurrency control locks on a FADU basis is allowed in addition to a file basis.

Legal values: y (yes) or n (no)

Default value: n

26.recovery: (RECOVERY)

Specifies if the initiator is allowed to recreate an open regime which has been destroyed by some failure.

Legal values: y (yes) or n (no)

Default value: n

27.r_data_xfer: (DATAXFER)

Specifies if data transfer service is allowed to be interrupted and restarted at a negotiated point.

Legal values: y (yes) or n (no)

Default value: n

28.lim_fs_mgmt: (LIMFSMGMT)

Specifies if filestore management for the interrogation of the attributes of a filestore is provided.

Legal values: y (yes) or n (no)

Default value: n

29.enh_fs_mgmt: (ENHFSMGMT)

Specifies if filestore management for the modification of the attributes of a filestore is provided.

Legal values: y (yes) or n (no)

Default value: n

30.obj_manip: (OBJMANIP)

Specifies if the functionality object manipulation is provided.

Legal values: y (yes) or n (no)

Default value: n

31.grp_manip: (GRPMANIP)

Specifies if functionality of group manipulation is provided.

Legal values: y (yes) or n (no)

Default value: n

fa_actions:

The file access actions are possible actions that may be performed on a file. These actions are read, insert, replace, extend, erase, read attribute, change attribute, delete file, passthrough, and link. The file_access_actions on this form are a super set of those on the *gfssec* form. This means that given a *gfsappl* populated form, the dtp_common attribute identified an instance of this form. The attribute dtp_responder identified an instance of a responder form, which contains two lists: file_access_list and path_access_list. Those two lists may contain links to the *gfssec* form for this application. For all *gfssec* forms linked to this application, the union of the file_access_actions that are set to "y" must be set to "y" on this form.

33.read: (READ)

Specifies if an initiator is allowed to locate and read an FADU.

Legal values: y (yes) or n (no)

Default value: y

34.insert: (INSERT)

Specifies if an initiator is allowed to create a new FADU and insert it into a position in the file specified in the constraint set.

Legal values: y (yes) or n (no)

Default value: n

35.replace: (REPLACE)

Specifies if an initiator is allowed to replace the contents in an existing DU or FADU.

Legal values: y (yes) or n (no)

Default value: n

36.extend: (EXTEND)

Specifies if an initiator is allowed to add data to the end of the DU associated with the root node of the currently located FADU.

Legal values: y (yes) or n (no)

Default value: n

37.erase: (ERASE)

Specifies if an initiator is allowed to erase the FADU.

Legal values: y (yes) or n (no)

Default value: n

38.read_attr: (READATTR)

Specifies if an initiator is allowed to interrogate the values of requested attributes.

Legal values: y (yes) or n (no)

Default value: n

39.chg_attr: (CHGATTR)

Specifies if an initiator is allowed to change existing file attributes.

Legal values: y (yes) or n (no)

Default value: n

40.delete_file: (DELETEFILE)

Specifies if an initiator is allowed to delete and deselect the file.

Legal values: y (yes) or n (no)

Default value: n

41.passthrough: (PASSTHROUGH)

Specifies if an initiator is allowed the use of a file-directory object name within a pathname.

Legal values: y (yes) or n (no)

Default value: n

42.link: (LINK)

Specifies if an initiator is allowed to create reference links to the given object.

Legal values: y (yes) or n (no)

Default value: n

fadu_id_group:

A file access data unit group has many possible methods for accessing each unit. Three methods are traversal, reverse traversal, and random.

44.traversal: (TRAVERSAL)

Specifies if a preorder traversal of a hierarchical subtree is used for identifying FADUs.

Legal values: y (yes) or n (no)

Default value: y

45.r_traversal: (RTRAVERSAL)

Specifies if a reverse preorder traversal of a hierarchical subtree is used for identifying FADUs.

Legal values: y (yes) or n (no)

Default value: y

46.random: (RANDOM)

Specifies if an identifier, a path of FADU identifiers, or a node-number is used to identify FADUs.

Legal values: y (yes) or n (no)

Default value: n

document_type:

Document type is one format for specifying members of a content type list.
Four document types are available:

- FTAM-1 : unstructured text file
- FTAM-2 : sequential text file
- FTAM-3 : unstructured binary file
- FTAM-4 : sequential binary file

FTAM-1

48.selected: (SELECTED)

Specifies if a document type of FTAM-1 applies.

Legal values: y (yes) or n (no)

Default value: n

49.string_sig: (STRINGSIG)

Data format of the document_type.

Legal values: va (variable), fi (fixed), or ns (not_significant)

Default value: ns

50.string_len: (STRINGLEN)

Maximum size of data unit being transferred. Zero indicates buffer size restricted by associated Input/Output (I/O) buffer size. The size is given in the number of bytes. For best performance, use string lengths greater than or equal to 512 bytes.

Legal values: 0 – 130048

Default value: 1024

51.char_set: (CHARSET)

Character set used for this document type.

Legal values: prt (PrintableString) tel (TeletexString)
vid (VideotextString) ia5 (Ia5String)
gra (GraphicString) vis (VisibleString)
gen (GeneralString)

Default value: ia5

52.eol: (EOL)

Specifies if there is end-of-line processing.

Legal values: y (yes) or n (no)

Default value: n

FTAM-2

54.selected: (SELECTED)

Specifies if a document type of FTAM-2 applies.

Legal values: y (yes) or n (no)

Default value: n

55.string_sig: (STRINGSIG)

Data format of the document type.

Legal values: va (variable), fi (fixed), or ns (not_significant)

Default value: ns

56.string_len: (STRINGLEN)

Maximum size of data unit being transferred. Zero indicates buffer size restricted by associated I/O buffer size. The size is given in the number of bytes. For best performance, use string lengths greater than or equal to 512 bytes.

Legal values: 0 – 130048

Default value: 1024

57.char_set: (CHARSET)

Character set for this document type.

Legal values: prt (PrintableString) tel (TeletexString)
vid (VideotextString) ia5 (Ia5String)
gra (GraphicString) vis (VisibleString)
gen (GeneralString)

Default value: ia5

58.eol: (EOL)

Specifies if there is end-of-line processing.

Legal values: y (yes) or n (no)

Default value: n

FTAM-3

60.selected: (SELECTED)

Specifies if a document type of FTAM-3 applies.

Legal values: y (yes) or n (no)

Default value: n

61.string_sig: (STRINGSIG)

Data format of the document.

Legal values: va (variable), fi (fixed), or ns (not_significant)

Default value: ns

62.string_len: (STRINGLEN)

Maximum size of data unit being transferred. Zero indicates buffer size restricted by associated I/O buffer size. The size is given in the number of bytes. For best performance, use string lengths greater than or equal to 512 bytes.

Legal values: 0 – 130048

Default value: 8192

FTAM-4

64.selected: (SELECTED)

Specifies if a document type of FTAM-4 applies.

Legal values: y (yes) or n (no)

Default value: n

65.string_sig: (STRINGSIG)

Data format of the document type.

Legal values: va (variable), fi (fixed), or ns (not_significant)

Default value: ns

66.string_len: (STRINGLEN)

Maximum size of data unit being transferred. Zero indicates buffer size restricted by associated I/O buffer size. The size is given in the number of bytes. For best performance, use string lengths greater than or equal to 512 bytes.

Legal values: 0 – 130048

Default value: 8192

content_type_list:

List of 1 – 8 constraint set/abstract syntax pairs.

68.constraint_set: (CONSTRAINTSET)

A set of restrictions and refinements of a general file model which specifies a less general model tailored to the needs of a particular class of applications, which is given an identifier.

Legal values:

us (unstructured)	sf (sequential_flat)
of (ordered_flat)	ofu (ordered_flat_unique)
oh (ordered_hierarchical)	gh (general_hierarchical)
ghu (general_hierarchical_unique)	

Default value: sf

69.abstract_syntax: (ABSTRACTSYNTAX)

Abstract syntax is an object identifier that uniquely identifies the abstract syntax of the file data and is used to reference the abstract syntax in the FTAM protocol.

Legal values: 0 – 8 characters specifying an object id

Default value: null (blank)

Form Checks

The following checks are required for this form:

```
/*
The following checks are made because these fields
exist on the forms but are not currently supported
*/
(count {select * from con_abst} must_be 0)
    error: "Constraint set/abstract syntax not currently supported";

(protocol_version = "ftam88")
    error: "FTAM-88 only supported currently";

(service_class.unconstrained = "n")
    error: "Service class unconstrained not currently supported";
(service_class.access = "n")
    error: "Service class access not currently supported";

(func_units.fu_file_access = "n")
    error: "Functional unit file access not currently supported";
(func_units.fu_enh_f_mgmt = "n")
    error: "Functional unit enhanced file management not currently
supported";
(func_units.fu_fadu_locking = "n")
    error: "Functional unit FADU locking not currently supported";
(func_units.fu_recovery = "n")
    error: "Functional unit recovery not currently supported";
(func_units.fu_r_data_xfer = "n")
    error: "Functional unit restart data transfer not currently
supported";
(func_units.fu_lim_fs_mgmt = "n")
    error: "Functional unit limited filestore management not currently
supported";
(func_units.fu_enh_fs_mgmt = "n")
    error: "Functional unit enhanced filestore management not currently
supported";
(func_units.fu_obj_manip = "n")
    error: "Functional unit object manipulation not currently supported";
(func_units.fu_grp_manip = "n")
    error: "Functional unit group manipulation not currently supported";

(fa_actions.fa_insert = "n")
    error: "File access action insert not currently supported";
```

```
(fa_actions.fa_erase = "n")
    error: "File access action erase not currently supported";
(fa_actions.fa_chg_attr = "n")
    error: "File access action change attribute not currently
    supported";
(fa_actions.fa_passthrough = "n")
    error: "File access action passthrough not currently supported";
(fa_actions.fa_link = "n")
    error: "File access action link not currently supported";

(fadu_id_group.random = "n")
    error: "FADU identity group random access not supported";

(recovery_mode in {"n", "s"})
    error: "Recovery mode 'at any active checkpoint' not currently
    supported";

(ftam_1_doc.ftam1_selected = "y") or (ftam_3_doc.ftam3_selected = "y")
    error: "Document type FTAM-1 or FTAM-3 must be selected";

(ftam_2_doc.ftam2_selected = "n")
    error: "Documented type FTAM-2 not currently supported";

(ftam_4_doc.ftam4_selected = "n")
    error: "Documented type FTAM-4 not currently supported";

if (func_units.fu_enh_f_mgmt = "y")
then (func_units.fu_lim_f_mgmt = "y")
    error: "Enhanced file management requires limited file
    management";

if (func_units.fu_enh_fs_mgmt = "y")
then (func_units.fu_lim_fs_mgmt = "y")
    error: "Enhanced file store management requires limited
    file store management";

/*
The following checks are valid even when the above checks
are removed at some point in time.
*/
inactivity_timer >= file_slct_timer
    error: "inactivity_timer must be greater than or equal to file_slct_timer";
(ftam_1_doc.string_length = 0) or
(ftam_1_doc.string_length >= 512)
    warning: "For best performance - use string lengths >= 512 bytes";

(ftam_2_doc.string_length = 0) or
(ftam_2_doc.string_length >= 512)
    warning: "For best performance - use string lengths >= 512 bytes";

(ftam_3_doc.string_length = 0) or
(ftam_3_doc.string_length >= 512)
    warning: "For best performance - use string lengths >= 512 bytes";

(ftam_4_doc.string_length = 0) or
(ftam_4_doc.string_length >= 512)
    warning: "For best performance - use string lengths >= 512 bytes";
```

```
(service_class.unconstrained = "y") or
(service_class.access = "y") or
(service_class.tran_and_mgmt = "y") or
(service_class.management = "y") or
(service_class.transfer = "y")
  error: "At least one of the service classes must be set to yes";

if (func_units.fu_write = "n")
  then do;
    (fa_actions.fa_insert = "n")
    error: "File access action insert needs functional unit write";
    (fa_actions.fa_replace = "n")
    error: "File access action replace needs functional unit write";
    (fa_actions.fa_extend = "n")
    error: "File access action extend needs functional unit write";
    (fa_actions.fa_erase = "n")
    error: "File access action erase needs functional unit write";
  end;

if (recovery_mode = "c")
  then do;
    (func_units.fu_recovery = "y")
    error: "Functional unit recovery needed for recovery mode
    'at any active checkpoint'";
    (func_units.fu_r_data_xfer = "y")
    error: "Functional unit restart data transfer needed for
    recovery mode 'c'";
  end;

if (protocol_version = "ftam88")
  then do;
    (func_units.fu_lim_fs_mgmt = "n")
    error: "Functional unit limited filestore management only
    valid for FTAM89a";

    (func_units.fu_enh_fs_mgmt = "n")
    error: "Functional unit enhanced filestore management only
    valid for FTAM89a";

    (func_units.fu_obj_manip = "n")
    error: "Functional unit object manipulation only valid for
    FTAM89a";
    (func_units.fu_grp_manip = "n")
    error: "Functional unit group manipulation only valid for
    FTAM89a";
    (fa_actions.fa_passthrough = "n")
    error: "File access action passthrough only valid for
    FTAM89a";
    (fa_actions.fa_link = "n")
    error: "File access action link only valid for FTAM89a";
  end;
```

```
if (service_class.transfer = "y")
then do;
  (func_units.fu_read = "y") or (func_units.fu_write = "y")
error: "Transfer service class must have functional unit
  read or write";
  (func_units.fu_file_access = "n")
error: "Transfer service class cannot have functional unit
  file access";
  (func_units.fu_grouping = "y")
error: "Transfer service class must have functional unit
  grouping";
  (func_units.fu_fadu_locking = "n")
error: "Transfer service class can not have functional unit
  FADU locking";
end;

if (service_class.access = "y")
then do;
  (func_units.fu_read = "y")
error: "Access service class must have functional unit read";
  (func_units.fu_write = "y")
error: "Access service class must have functional unit write";
  (func_units.fu_file_access = "y")
error: "Access service class must have functional unit file
  access";
end;

if (service_class.management = "y")
then do;
  (func_units.fu_read = "n")
error: "Management service class can not have functional
  unit read";
  (func_units.fu_write = "n")
error: "Management service class can not have functional
  unit write";
  (func_units.fu_file_access = "n")
error: "Management service class can not have functional
  unit file access";
  (func_units.fu_lim_f_mgmt = "y")
error: "Management service class must have functional unit
  limited file management";
  (func_units.fu_grouping = "y")
error: "Management service class must have functional unit
  grouping";
  (func_units.fu_fadu_locking = "n")
error: "Management service class can not have functional
  unit FADU locking";
  (func_units.fu_recovery = "n")
error: "Management service class can not have functional
  unit recovery";
  (func_units.fu_r_data_xfer = "n")
error: "Management service class can not have functional
  unit restart data transfer";
end;
```

```
if (service_class.tran_and_mgmt = "y")
then do;
  (func_units.fu_read = "y") or (func_units.fu_write = "y")
error: "Transfer/Mgmt service class must have functional
      unit read or write";
  (func_units.fu_file_access = "n")
error: "Transfer/Mgmt service class can not have functional
      unit file access";
  (func_units.fu_lim_f_mgmt = "y")
error: "Transfer/Mgmt service class must have functional
      unit limited file management";
  (func_units.fu_grouping = "y")
error: "Transfer/Mgmt service class must have functional
      unit grouping";
  (func_units.fu_fadu_locking = "n")
error: "Transfer/Mgmt service class can not have functional
      unit FADU locking";
end;
```

gfsinit

This form is used to provide initiator-specific information on General File Transfer (GFT) tape and data link processes along with the buffer usage for these.

Form Layout

GFS Initiator Definition			gfsinit	(1/2)
			(Recent Change and Verify)	
1.dtp_initiator_id: _____				
2.tape_process				
3.tape_pathname: _____				
4.tape_layout: _____		5.tape_standard: _____		
buffer_type	number_of_buffers	buffer_size		
send_input	6)_____	7)_____		
send_output	8)_____	9)_____		
10.bsl_sel: _____		11.apl_sel: _____		12.dtl_sel: _____

			gfsinit	(2/2)
13.data_link_process				
14.dl_pathname: _____				
buffer_type	number_of_buffers	buffer_size		
send_input	15)_____	16)_____		
send_output	17)_____	18)_____		
receive_input	19)_____	20)_____		
receive_output	21)_____	22)_____		
23.bsl_sel: _____		24.apl_sel: _____		25.dtl_sel: _____
26.file_append:_		27.file_overwrite:_		28.snd_switch_id:_

Definition of Fields

1.dtp_initiator_id: (DTPINITIATORID) Required

The name from the *gfsappl* form to which this information is linked. This is a key field.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

2.tape_process: (TAPEPROCESS)

Fields 3 – 14 are concerned with initiator applications using a tape process.

3.tape_pathname: (TAPEPATHNAME)

Tape process path.

Legal values: 0 – 64 alphanumeric characters

Default value: no default

4.tape_layout: (TAPELAYOUT)

Tape layout allowing for default tape writer.

Legal values: none, iso (ISO001), seq (sequential), or rep (repeated_prefix)

Default value: iso

5.tape_standard: (TAPESTANDARD)

Tape standard allowing for default tape writer.

Legal values: NONE, ISO1, ISO3, ISO4, IBM83, or IBM87

Default value: ISO4

buffer_type.send_input:

For the tape process, specifies the input buffer information needed for the send buffer.

6.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of input buffers for the send buffer.

Legal values: 0 – 127

Default value: 4

7.buffer.size: (BUFFERSIZE)

Specifies the input buffer size for the send buffer.

Legal values: 0 – 130048

Default value: 6144

buffer_type.send_output:

For the tape process, specifies the output buffer information needed for the send buffer.

8.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of output buffers for the send buffer.

Legal values: 0 – 127

Default value: 4

9.buffer.size: (BUFFERSIZE)

Specifies the output buffer size for the send buffer.

Legal values: 0 – 130048

Default value: 6144

10.bsl_sel: (BSLSEL)

Specifies the version of the data transfer bulk storage library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x0001

11.apl_sel: (APLSEL)

Specifies the version of the data transfer application library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x0001

12.dtl_sel: (DTLSEL)

Specifies the version of the data transfer library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x0001

13.data_link_process: (DLPROCESS)

Fields 3 – 14 are concerned with initiator applications using a data link process.

14.dl_pathname: (DLPATHNAME)

Data link process path.

Legal values: 1 – 64 alphanumeric characters or null

Default value: no default

buffer_type.send_input:

For the data link process, specifies the input buffer information needed for the send buffer.

15.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of input buffers for the send buffer.

Legal values: 0 – 127

Default value: 0

16.buffer_size: (BUFFERSIZE)

Specifies the input buffer size (multiple of 512) bytes for the send buffer.

Legal values: 0 – 130048

Default value: 0

buffer_type.send_output:

For the data link process, specifies the output buffer information needed for the send buffer.

17.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of output buffers for the send buffer.

Legal values: 0 – 127

Default value: 0

18.buffer_size: (BUFFERSIZE)

Specifies the output buffer size (multiple of 512) bytes for the send buffer.

Legal values: 0 – 130048

Default value: 0

buffer_type.receive_input:

For the data link process, specifies the input buffer information needed for the receive buffer.

19.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of input buffers for the receive buffer.

Legal values: 0 – 127

Default value: 0

20.buffer_size: (BUFFERSIZE)

Specifies the input buffer size (multiple of 512) bytes for the receive buffer.

Legal values: 0 – 130048

Default value: 0

buffer_type.receive_output:

For the data link process, specifies the output buffer information needed for the receive buffer.

21.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of output buffers for the receive buffer.

Legal values: 0 – 127

Default value: 0

22.buffer_size: (BUFFERSIZE)

Specifies the output buffer size (multiple of 512) bytes for the receive buffer.

Legal values: 0 – 130048

Default value: 0

23.bl_ssel: (BLSSEL)

Specifies the version of the data transfer bulk storage library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x0001

24.apl_sel: (APLSEL)

Specifies the version of the data transfer application library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x0001

25.dtl_sel: (DTLSEL)

Specifies the version of the data transfer library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x0001

26.file_append:(FILEAPPEND)

This field specifies the requested action on the f_create, f_open, and f_write primitives. When the 5ESS® switch is an initiator and the file_append option is set to “y,” the file will be appended to the existing file.

Legal values: y or n

Default value: n

27.file_overwrite:(FILEOVERWRITE)

This field specifies the requested action on the f_create, f_open, and f_write primitives. When the 5ESS switch is an initiator and the file_overwrite option is set to “y,” the file will be overwritten to the existing file.

Legal values: y or n

Default value: n

28.snd_switch_id:(SNDSWITCHID)

This field specifies switch id for remote connections. When the 5ESS switch is an initiator and the snd_switch_id option is set to “y,” the switch id will be added to the initiator id.

Legal values: y or n

Default value: n

Form Checks

The following checks are performed on this form:

```
if (file_overwrite is "y")
    then (file_append must_be "n")
        error: "when file_overwrite is set to 'y', file_append must be 'n'";

if (tape_process.sl_num_out_buf != 0)
then do;
    (tape_process.sl_num_in_buf != 0)
```

```
error: "Tape input send buffer needed when there is an
output send buffer";

(tape_process.sl_out_buf_size != 0)
error: "Tape output send buffer size must be specified";

(tape_process.sl_out_buf_size <= 6144)
error: "Tape output send buffer size should be less than
or equal to 6144";
end;
else do;
(tape_process.sl_in_buf_size <= 6144)
error: "Tape input send buffer size should be less than or
equal to 6144";
end;

if (tape_process.sl_num_in_buf != 0)
then (tape_process.sl_in_buf_size != 0)
error: "Tape input send buffer size must be specified";

(dl_process.dl_pathname must_be known) or
(tape_process.tape_pathname must_be known)
error: "At least one of the two initiator processes must exist";

if (dl_process.sl_num_out_buf != 0)
then do;
(dl_process.sl_num_in_buf != 0)
error: "Data link input send buffer needed when there is
an output send buffer";

(dl_process.sl_out_buf_size != 0)
error: "Data link output send buffer size must be specified";
end;

if (dl_process.sl_num_out_buf is 0)
then (dl_process.sl_out_buf_size must_be 0)
error: "Send output buffer size must be zero when num_ out_buf is zero";

if (dl_process.sl_num_in_buf != 0)
then (dl_process.sl_in_buf_size != 0)
error: "Data link input send buffer size must be specified";
if (dl_process.sl_num_in_buf is 0)
then (dl_process.sl_in_buf_size must_be 0)
error: "Send input buffer size must be zero when num_in_buf is zero";

if (dl_process.rl_num_out_buf != 0)
then do;
(dl_process.rl_num_in_buf != 0)
error: "Data link input receive buffer needed when there
is an output receive buffer";

(dl_process.rl_out_buf_size != 0)
error: "Data link output receive buffer size must be specified";
end;
```

```
if (dl_process.rl_num_out_buf is 0)
then (dl_process.rl_out_buf_size must_be 0)
  error: "Receive output buffer size must be zero when receive
  num_out_buf is zero";

if (dl_process.rl_num_in_buf != 0)
then (dl_process.rl_in_buf_size != 0)
  error: "Data link input receive buffer size must be specified";
if (dl_process.rl_num_in_buf is 0)
then (dl_process.rl_in_buf_size must_be 0)
  error: "Receive input buffer size must be zero when receive num_in_buf
  is zero";

if (dl_process.dl_pathname is known)
then ((dl_process.sl_num_in_buf != 0) or (dl_process.rl_num_in_buf != 0))
  error: "There must be at least one send or receive input buffer";

/* Check if the receive input buffer is greater than 7K */
/* octets (7 * 1024 = 7168).  If not, print a warning */
/* message.  Note: this check uses negative logic. */

(dl_process.rl_in_buf_size <= 0) or
(dl_process.rl_in_buf_size >= 7168)
  warning: "The receive input buffer size has been set lower than the required
  7K octets";
```

gfsresp

This form is used to provide links to the *gfssec* form defining security information and information regarding permitted actions, buffer usage, and the pathname processing that is allowed for General File Transfer (GFT) applications in a responder role.

Form Layout

GFS Responder Definition	gfsresp	(1/4) (Recent Change and Verify)
1.dtp_responder_id: _____		
2.password_encoding: ____		
3.digital_signature_key: _____		
4.encrypted_key: _____		
5.vfs_pwd_chk: ____	6.vfs_password: _____	
7.create_pwd_chk: ____	8.create_password: _____	

	gfsresp	(2/4)
9.file_access_list:	11.path_access_list:	
1) _____	1) _____	
2) _____	2) _____	
3) _____	3) _____	
4) _____	4) _____	
5) _____	5) _____	
6) _____	6) _____	
7) _____	7) _____	
8) _____	8) _____	
9) _____	9) _____	
10) _____	10) _____	

gfsresp (3/4)

13.process_creation_type: ____

data_link_process

15.dl_pathname: _____

buffer_type	number_of_buffers	buffer_size
send_input	16) _____	17) _____
send_output	18) _____	19) _____
receive_input	20) _____	21) _____
receive_output	22) _____	23) _____

24.bsl_sel: _____ 25.apl_sel: _____ 26.dtl_sel: _____

gfsresp (4/4)

27.prcss_portname: ____

28.prcss_pathname: _____

Definition of Fields

1.dtp_responder_id: (DTPRESPONDERID) Required

The name from the *gfsappl* form to which this information is linked. This is a key field.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

2.password_encoding: (PASSWORDENCODING)

Password encoding type to be performed.

Legal values: n (no_encryption) or b (bellcore)

Default value: n

3.digital_signature_key: (DIGITALSIGNATUREKEY)

The key used to encrypt the digital signature key.

Legal values: 1 – 64 printable characters

Default value: null string

4. encryption_key: (ENCRPTIONKEY)

The key used to encrypt other user data.

Legal values: 1 – 64 printable characters

Default value: null string

5. vfs_pwd_chk: (VFSPWDNULL)

Specifies if the virtual file system password is checked.

Legal values: y or n

Default value: y

6. vfs_password: (VFSPASSWORD)

Virtual file system password.

Legal values: 0 – 16 printable characters

Default value: null string

7. create_pwd_chk: (CREATEPWDNULL)

Specifies if the create password is checked.

Legal values: y or n

Default value: n

8. create_password: (CREATEPASSWORD)

Password allowing one to create objects in the filestore.

Legal values: 0 – 16 printable characters

Default value: null string

9. file_access_list: (FILEACCESSLIST)

List of links to the *gfssec* low-level form which is used for specifying the file access actions, concurrency control, and users.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

11.path_access_list: (PATHACCESSLIST)

List of links to the *gfssec* low-level form which is used for specifying the file access actions, concurrency control, and users through a pathname.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

13.process_creation_type: (PROCESSCREATIONTYPE)

Specifies the mechanism by which the responder process is created.

Legal values: p (pcreated) or e (e_usr1_poll)

Default value: p

14.data_link_process: (DLPROCESS)

The data link responder process.

15.dl_pathname: (DLPATHNAME)

Data link path. This must be null if the process_creation_type is "p."

Legal values: 1 – 64 alphanumeric characters or null

Default value: no default

send_buffer.input_buffer:

For the data link process, specifies the input buffer information needed for the send buffer.

16.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of input buffers for the send buffers.

Legal values: 0 – 127

Default value: 4

17.buffer_size: (BUFFERSIZE)

Specifies the input buffer size (multiple of 512) in bytes for the send buffer.

Legal values: 0 – 130048

Default value: 8192

send_buffer.output_buffer:

For the data link process, specifies the output buffer information needed for the send buffer.

18.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of output buffers for the send buffer.

Legal values: 0 – 127

Default value: 4

19.buffer_size: (BUFFERSIZE)

Specifies the output buffer size (multiple of 512) in bytes for the send buffer.

Legal values: 0 – 130048

Default value: 8192

receive_buffer.input_buffer

20.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of input buffers for the receive buffer.

Legal values: 0 – 127

Default value: 4

21.buffer_size: (BUFFERSIZE)

Specifies the input buffer size (multiple of 512) in bytes for the receive buffer.

Legal values: 0 – 130048

Default value: 8192

receive_buffer.output_buffer:

22.number_of_buffers: (NUMBEROFBUFFERS)

Specifies the number of output buffers for the receive buffer.

Legal values: 0 – 127

Default value: 4

23.buffer.size: (BUFFERSIZE)

Specifies the output buffer size (multiple of 512) in bytes for the receive buffer.

Legal values: 0 – 130048

Default value: 8192

24.bsl_sel: (BSLSEL)

Specifies the version of the bulk storage library, where bulk storage refers to storage for billing data and raw data.

Legal values: 0x0000 – 0xFFFF

Default value: 0x1

25.apl_sel: (APLSEL)

Specifies the version of the application library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x1

26.dtl_sel: (DTLSEL)

Specifies the version of the data transfer library.

Legal values: 0x0000 – 0xFFFF

Default value: 0x1

27.prcss_portname:(PRCSSPORTNAME)

This field specifies a process port name. A notify message is sent to the process after arrival of a file and/or closing a connection.

Legal values: 50 – 250, unknown

Default value: unknown

28.prcss_pathname:(PRCSSPATHNAME)

This field specifies the process path to be created after the arrival of a file and/or closing a connection.

Legal values: alphanumeric characters, unknown

Default value: unknown

Form Checks

The following checks are performed on this form:

```
/*
The following checks are done since these fields exist
on the forms but are not totally supported
*/
(password_encode = "n")
    error: "Only 'no encryption' password encoding type is supported";

(proc_creat_type = "p")
    error: "Only 'pcreated' process creation type is supported";

count {select * from path_access_list} = 0
    error: "Path access list currently not supported";

/*
The following checks must always be made, regardless
of whether the above checks continue to exist.
*/

if prcss_pathname is known
then prcss_portname must_be unknown
    error:"When prcss_pathname is set, prcss_portname
must not be set";

if prcss_portname is known
then prcss_pathname must_be unknown
    error:"When prcss_portname is set, prcss_pathname
must not be set";

if (password_encode = "n")
then do;
(digital_sig_key must_be unknown)
    error: "Digital signature key invalid for 'no encryption'";
(encryption_key must_be unknown)
    error: "Encryption key invalid for 'no encryption'";
end;
else do;
(digital_sig_key must_be known)
    error: "Digital signature key needed for 'bellcore'";

(encryption_key must_be known)
    error: "Encryption key needed for 'bellcore'";

(vfs_pwd_null must_be "n")
    error: "Vfs password not allowed for 'bellcore'";
end;

if (proc_creat_type = "p")
then
(dl_process.dl_pathname must_be unknown)
    error: "Data link pathname must be null for 'PCREATED'";
else
(dl_process.dl_pathname must_be known)
    error: "Data link pathname must be non-null for 'E_USR1_POLL'";
```

```
if (dl_process.sl_num_out_buf != 0)
then do;
  (dl_process.sl_num_in_buf != 0)
  error: "Data link input send buffer needed when there is
        an output send buffer";

  (dl_process.sl_out_buf_size != 0)
  error: "Data link output send buffer size must be specified";
end;

if (dl_process.sl_num_in_buf != 0)
then (dl_process.sl_in_buf_size != 0)
  error: "Data link input send buffer size must be specified";
if (dl_process.sl_num_in_buf is 0)
then (dl_process.sl_in_buf_size must_be 0)
  error: "Send input buffer size must be zero when num_in_buf
        is zero";
if (dl_process.sl_num_out_buf is 0)
then (dl_process.sl_out_buf_size must_be 0)
  error: "Send output buffer size must be zero when num_out_buf
        is zero";

if (dl_process.rl_num_out_buf != 0)
then do;
  (dl_process.rl_num_in_buf != 0)
  error: "Data link input receive buffer needed when there
        is an output receive buffer";

  (dl_process.rl_out_buf_size != 0)
  error: "Data link output receive buffer size must be
        specified";
end;

if (dl_process.rl_num_out_buf is 0)
then (dl_process.rl_out_buf_size must_be 0)
  error: "Receive output buffer size must be zero when
        num_out_buf is zero";

if (dl_process.rl_num_in_buf != 0)
then (dl_process.rl_in_buf_size != 0)
  error: "Data link input receive buffer size must be specified";

(dl_process.sl_num_in_buf != 0) or
(dl_process.rl_num_in_buf != 0)
  error: "There must be at least one send or receive input buffer";

if (dl_process.rl_num_in_buf is 0)
then (dl_process.rl_in_buf_size must_be 0)
  error: "Receive input buffer size must be zero when num_in_buf
        is zero";
if (vfs_pwd_null = "n")
then (vfs_password must_be unknown)
  error: "If vfs_pwd_chk is 'n', then vfs_password must be null";

if (creat_pwd_null = "n")
then (creat_password must_be unknown)
  error: "If creat_pwd_chk is 'n', then creat_password must be null";

((count {select * from path_access_list}) +
```

```
(count {select * from file_access_list})) <= 10
    error: "Number of path and file access list elements must be <= 10";
/* Check if the receive input buffer size is not 0. */
/* If it's not 0, then verify it is greater than or */
/* equal to 7K octets (7 * 1024 = 7168). If not, */
/* print a warning message. */

if (dl_process.rl_in_buf_size != 0)
then (dl_process.rl_in_buf_size >= 7168)
    warning: "The receive buffer size has been set lower than
            the required 7K octets";
```

gfssec

This form is used to provide the necessary security information for a General File Transfer (GFT) application in a responder role.

Form Layout

```

                                     gfssec          (1/2)
GFS Security/Access Definition      (Recent Change and Verify)

1.access_instance: _____

2.identity: _____

remote_application_entity_title

4.process_title: _____

5.qualifier: _____
```

```

                                     gfssec          (2/2)

file_access_actions

                password_information          concurrency_access

read:           permitted  pwd?      password  no   exc  shr  nr
insert:        7)_       8)_       9)_____ 10)_ 11)_ 12)_ 13)_
replace:       15)_      16)_      17)_____ 18)_ 19)_ 20)_ 21)_
extend:        23)_      24)_      25)_____ 26)_ 27)_ 28)_ 29)_
erase:         31)_      32)_      33)_____ 34)_ 35)_ 36)_ 37)_
read_attr:     39)_      40)_      41)_____ 42)_ 43)_ 44)_ 45)_
chg_attr:      47)_      48)_      49)_____ 50)_ 51)_ 52)_ 53)_
delete_file:   55)_      56)_      57)_____ 58)_ 59)_ 60)_ 61)_
passthrough:  63)_      64)_      65)_____ 66)_ 67)_ 68)_ 69)_
link:          71)_      72)_      73)_____ 74)_ 75)_ 76)_ 77)_
               79)_      80)_      81)_____ 82)_ 83)_ 84)_ 85)_
```

Definition of Fields

1.access_instance: (ACCESSINSTANCE) Required

The name from the *gfsresp* form to which this information is linked. This is a key field.

Legal values: 1 – 8 alphanumeric characters

Default value: no default

2.identity: (IDENTITY)

The user_identity is a graphical string and could mean for example that different individuals at a remote site may have their own passwords that allow them different degrees of access.

Legal values: 0 – 24 characters

Default value: no default

remote_application_entity_title:

An identifier, naming the calling application entity, composed of the next two fields.

4.process_title: (PROCESSTITLE)

This is the name/site of the calling entity.

Legal values: 0 – 8 characters

Default value: null

5.qualifier: (QUALIFIER)

A qualifier placed on the name of the calling application entity.

Legal values: 0 – MAXINT

Default value: 0

file_access_actions:

For the read, insert, replace, extend, erase, read_attr, chg_attr, delete_file, passthrough, and link fields, the following subfields must be specified:

7, 15, 23, 31, 39, 47, 55, 63, 71, and 79.

password_information.permitted: Specifies if this file access action is permitted for field 'x'.

Legal values: y (yes) or n (no)

Default value: n

8, 16, 24, 32, 40, 48, 56, 64, 72, and 80.

password_information.pwd?: Is the password checked for this file_access_action?

Legal values: y (yes) or n (no)

Default value: n

9, 17, 25, 33, 41, 49, 57, 65, 73, and 81.

password_information.password: If pwd? is "y," then a password must be supplied. Although the password must be supplied, it may be a null string.

Legal values: 0 – 16 characters

Default value: null

10, 18, 26, 34, 42, 50, 58, 66, 74, and 82.

concurrency_access.no (not_required): If set to yes, implies no concurrency controls need to be in effect.

Legal values: y (yes) or n (no)

Default value: no

11, 19, 27, 35, 43, 51, 59, 67, 75, and 83.

concurrency_access.exc (exclusive): If set to yes, implies shared access is allowed.

Legal values: y (yes) or n (no)

Default value: no

12, 21, 28, 36, 44, 52, 60, 68, 76, and 84.

concurrency_access.shr (shared): If set to yes, implies exclusive access is allowed.

Legal values: y (yes) or n (no)

Default value: no

13, 21, 29, 37, 45, 53, 61, 69, 77, and 85.

concurrency_access.nr (no access): If set to yes, implies no access is allowed.

Legal values: y (yes) or n (no)

Default value: no

Form Checks

The following checks are needed for this form:

```
/*
The following checks are done since these fields exist
on the forms but are not totally supported
*/
(fa_insert.permitted = "n")
    error: "File access action 'insert' not currently supported";

(fa_erase.permitted = "n")
    error: "File access action 'erase' not currently supported";

(fa_chg_attr.permitted = "n")
    error: "File access action 'change attribute' not currently supported";

(fa_passthrough.permitted = "n")
    error: "File access action 'passthrough' not currently supported";

(fa_link.permitted = "n")
    error: "File access action 'link' not currently supported";

/*
The following checks must always be made, regardless
of whether the above checks continue to exist. Note
that if any concurrency check is set to 'y' that this
implies that the Concurrency Controls are being used;
otherwise, if they are all set to 'n', then it is regarded
as optional and no further checks are done.
*/

if (fa_read.cl_no_access          = "y") or
(fa_read.cl_exclusive            = "y") or
(fa_read.cl_shared               = "y") or
(fa_read.cl_not_required         = "y") or
(fa_insert.cl_no_access          = "y") or
(fa_insert.cl_exclusive          = "y") or
(fa_insert.cl_shared             = "y") or
(fa_insert.cl_not_required       = "y") or
(fa_replace.cl_no_access         = "y") or
(fa_replace.cl_exclusive         = "y") or
(fa_replace.cl_shared            = "y") or
(fa_replace.cl_not_required      = "y") or
(fa_extend.cl_no_access          = "y") or
(fa_extend.cl_exclusive          = "y") or
(fa_extend.cl_shared             = "y") or
(fa_extend.cl_not_required       = "y") or
(fa_erase.cl_no_access           = "y") or
(fa_erase.cl_exclusive           = "y") or
(fa_erase.cl_shared              = "y") or
(fa_erase.cl_not_required        = "y") or
(fa_read_attr.cl_no_access       = "y") or
(fa_read_attr.cl_exclusive       = "y") or
(fa_read_attr.cl_shared          = "y") or
(fa_read_attr.cl_not_required    = "y") or
(fa_chg_attr.cl_no_access        = "y") or
```

```
(fa_chg_attr.cl_exclusive      = "y") or
(fa_chg_attr.cl_shared        = "y") or
(fa_chg_attr.cl_not_required  = "y") or
(fa_delete_file.cl_no_access  = "y") or
(fa_delete_file.cl_exclusive  = "y") or
(fa_delete_file.cl_shared     = "y") or
(fa_delete_file.cl_not_required = "y") or
(fa_passthrough.cl_no_access  = "y") or
(fa_passthrough.cl_exclusive  = "y") or
(fa_passthrough.cl_shared     = "y") or
(fa_passthrough.cl_not_required = "y") or
(fa_link.cl_no_access         = "y") or
(fa_link.cl_exclusive        = "y") or
(fa_link.cl_shared           = "y") or
(fa_link.cl_not_required     = "y")
then do;
(fa_read.cl_no_access        = "y") or
(fa_read.cl_exclusive       = "y") or
(fa_read.cl_shared          = "y") or
(fa_read.cl_not_required    = "y")
  error: "At least one concurrency lock must be specified
    for 'read'";

(fa_insert.cl_no_access      = "y") or
(fa_insert.cl_exclusive     = "y") or
(fa_insert.cl_shared        = "y") or
(fa_insert.cl_not_required  = "y")
  error: "At least one concurrency lock must be specified
    for 'insert'";

(fa_replace.cl_no_access    = "y") or
(fa_replace.cl_exclusive   = "y") or
(fa_replace.cl_shared       = "y") or
(fa_replace.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
    for 'replace'";

(fa_extend.cl_no_access    = "y") or
(fa_extend.cl_exclusive   = "y") or
(fa_extend.cl_shared       = "y") or
(fa_extend.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
    for 'extend'";

(fa_erase.cl_no_access     = "y") or
(fa_erase.cl_exclusive     = "y") or
(fa_erase.cl_shared        = "y") or
(fa_erase.cl_not_required  = "y")
  error: "At least one concurrency lock must be specified
    for 'erase'";

(fa_read_attr.cl_no_access = "y") or
(fa_read_attr.cl_exclusive = "y") or
(fa_read_attr.cl_shared    = "y") or
(fa_read_attr.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
    for 'read attribute'";

(fa_chg_attr.cl_no_access  = "y") or
```

```
(fa_chg_attr.cl_exclusive = "y") or
(fa_chg_attr.cl_shared = "y") or
(fa_chg_attr.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
        for 'change attribute'";

(fa_delete_file.cl_no_access = "y") or
(fa_delete_file.cl_exclusive = "y") or
(fa_delete_file.cl_shared = "y") or
(fa_delete_file.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
        for 'delete file'";

(fa_passthrough.cl_no_access = "y") or
(fa_passthrough.cl_exclusive = "y") or
(fa_passthrough.cl_shared = "y") or
(fa_passthrough.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
        for 'passthrough'";

(fa_link.cl_no_access = "y") or
(fa_link.cl_exclusive = "y") or
(fa_link.cl_shared = "y") or
(fa_link.cl_not_required = "y")
  error: "At least one concurrency lock must be specified
        for 'link'";

if (fa_read.permitted = "n")
then do;
  (fa_read.cl_exclusive = "n") and (fa_read.cl_shared = "n")
    error: "Invalid concurrency control for 'read' action";
end;

if (fa_insert.permitted = "n")
then do;
  (fa_insert.cl_exclusive = "n") and
  (fa_insert.cl_shared = "n")
    error: "Invalid concurrency control for 'insert' action";
end;

if (fa_replace.permitted = "n")
then do;
  (fa_replace.cl_exclusive = "n") and
  (fa_replace.cl_shared = "n")
    error: "Invalid concurrency control for 'replace' action";
end;

if (fa_extend.permitted = "n")
then do;
  (fa_extend.cl_exclusive = "n") and
  (fa_extend.cl_shared = "n")
    error: "Invalid concurrency control for 'extend' action";
end;

if (fa_erase.permitted = "n")
then do;
  (fa_erase.cl_exclusive = "n") and (fa_erase.cl_shared = "n")
    error: "Invalid concurrency control for 'erase' action";
end;
```

```
if (fa_read_attr.permitted = "n")
then do;
  (fa_read_attr.cl_exclusive = "n") and
  (fa_read_attr.cl_shared = "n")
  error: "Invalid concurrency control for 'read_attr'
  action";
end;

if (fa_chg_attr.permitted = "n")
then do;
  (fa_chg_attr.cl_exclusive = "n") and
  (fa_chg_attr.cl_shared = "n")
  error: "Invalid concurrency control for 'chg_attr' action";
end;

if (fa_delete_file.permitted = "n")
then do;
  (fa_delete_file.cl_exclusive = "n") and
  (fa_delete_file.cl_shared = "n")
  error: "Invalid concurrency control for 'delete_file'
  action";
end;

if (fa_passthrough.permitted = "n")
then do;
  (fa_passthrough.cl_exclusive = "n") and
  (fa_passthrough.cl_shared = "n")
  error: "Invalid concurrency control for 'passthrough'
  action";
end;

if (fa_link.permitted = "n")
then do;
  (fa_link.cl_exclusive = "n") and (fa_link.cl_shared = "n")
  error: "Invalid concurrency control for 'link' action";
end;
end;

if (fa_read.pwd_chk is "n")
then (fa_read.password must_be unknown)
  error: "If 'read' pwd_chk is 'n', then password must be
  unknown";

if (fa_insert.pwd_chk is "n")
then (fa_insert.password must_be unknown)
  error: "If 'insert' pwd_chk is 'n', then password must be
  unknown";

if (fa_replace.pwd_chk is "n")
then (fa_replace.password must_be unknown)
  error: "If 'replace' pwd_chk is 'n', then password must be
  unknown";

if (fa_extend.pwd_chk is "n")
then (fa_extend.password must_be unknown)
  error: "If 'extend' pwd_chk is 'n', then password must be
  unknown";

if (fa_erase.pwd_chk is "n")
```

```
then (fa_erase.password must_be unknown)
  error: "If 'erase' pwd_chk is 'n', then password must be
  unknown";

if (fa_read_attr.pwd_chk is "n")
then (fa_read_attr.password must_be unknown)
  error: "If 'read attribute' pwd_chk is 'n', then password
  must be unknown";

if (fa_chg_attr.pwd_chk is "n")
then (fa_chg_attr.password must_be unknown)
  error: "If 'change attribute' pwd_chk is 'n', then password
  must be unknown";

if (fa_delete_file.pwd_chk is "n")
then (fa_delete_file.password must_be unknown)
  error: "If 'delete file' pwd_chk is 'n', then password
  must be unknown";

if (fa_passthrough.pwd_chk is "n")
then (fa_passthrough.password must_be unknown)
  error: "If 'passthrough' pwd_chk is 'n', then password
  must be unknown";

if (fa_link.pwd_chk is "n")
then (fa_link.password must_be unknown)
  error: "If 'link' pwd_chk is 'n', then password must be
  unknown";
```

klist

This form selectively dumps the keys of key-based forms.

Form Layout

```
Key List                                klist          (1/3)
                                         (Execute Only)

1.temp_name:_____
2.aftopt_form:_   3.aphopt_form:_   4.aphuser_form:_   5.auderr_form:_
6.audhist_form:_  7.audinst_form:_  8.audrec_form:_    9.audstat_form:_
10.authdef_form:_ 11.btihdev_form:_ 12.btparam_form:_  13.bxsinfo_form:_
14.bxsiloc_form:_ 15.bxsrlmt_form:_ 16.cdopt_form:_    17.cftinfo_form:_
18.chopt_form:_   19.ciopt_form:_   20.classdef_form:_ 21.cpblx3_form:_
22.cpphnone_form:_ 23.dciopt_form:_  24.device_form:_   25.dfcopt_form:_
```

```
                                         klist          (2/3)

26.dst_form:_     27.duopt_form:_   28.duoptdyn_form:_ 29.eaiopt_form:_
30.ecdorg_form:_  31.errcount_form:_ 32.errthld_form:_  33.ftaminopt_form:_
34.ftamopt_form:_ 35.ftamuser_form:_ 36.getty_form:_    37.gfsappl_form:_
38.gfscom_form:_  39.gfsinit_form:_  40.gfsresp_form:_  41.gfssec_form:_
42.logdev_form:_  43.mdct_form:_    44.mhdopt_form:_   45.nphopt_form:_
46.objectid_form:_ 47.pphopt_form:_  48.pphuser_form:_  49.runa_form:_
50.scsdbody_form:_ 51.scsdmap_form:_ 52.scsdopt_form:_  53.simcntl_form:_
```

klist (3/3)

54.sphopt_form:_ 55.splrinfo_form:_ 56.tphopt_form:_ 57.tsapinfo_form:_
58.ttopt_form:_ 59.ucb_form:_ 60.ucbdyn_form:_ 61.ularp_form:_
62.workhrs_form:_

Definition of Fields

1. temp_name: (TEMPNAME) Required

The name of the file where keys are to be written. This is the key field.

Legal values: printable characters

2. aftopt_form: (AFTOPTFORM) — 62.workhrs_form: (WORKHOURSFORM)

These fields are used to tell whether to dump keys for the forms indicated by the field name. A “y” indicates that keys should be printed for this form type.

Legal values: y, n, null

logdev

Logical device. This form specifies information about a particular logical device.

Form Layout

Logical Device Information	logdev	(1/1) (Recent Change and Verify)
1.logical_name: _____	2.partition: _	
3.mdct_name: _____	4.channel_id: __	
5.dcn: __	6.driver_type:_	
7.gettyid: _____	8.usr: __	
9.grp: __	10.other: __	

Definition of Fields

1. logical_name: (LOGICALNAME) Required

Logical device name. This is the key field of the form and must begin with */dev/* or */dgn/* and a filename.

Logdev names for Small Computer System Interface (SCSI) tape units residing on the 3B21D computer hardware platform will be in the following form:

/dev/mtnsdb[r]

Where:

- n* Specifies the SCSI tape unit number.
- s* Specifies how the tape blocking factor will be handled.
 - f* Indicates that tape records will be of a fixed size (when writing tape, records will be padded with 0x00 by the Disk File Controller (DFC) as necessary).
 - v* Indicates that tape records will be variable in size.
- d* Specifies the tape writing density.

- l Indicates low density (1,600 bpi).
- h Indicates high density (6,250 bpi).

Digital Audio Tape (DAT) drives do not support multiple density writing. Therefore, the density specification will be ignored when the tape drive is a DAT unit.

- b* Specifies the tape blocking factor (record size).
 - 0 Indicates that the maximum record size is limited only by the system/tape hardware (invalid if the record sizes are fixed).
 - non-0 Indicates record size at multiples of (512 * b) bytes:
 - Minimum record size if record sizes are fixed.
 - Maximum record size if record sizes are variable.

- r* Specifies the rewind/no-rewind upon file close option.

Opened for writing:

- null Indicates that upon device file close, two filemarks will be written on tape and the tape rewound.
- n Indicates that upon device file close, two filemarks will be written on tape and the tape is positioned in between the two filemarks.

Opened for reading:

- null Indicates that upon device file close, tape will be rewound to the Beginning of Tape (BOT).
- n Indicates that upon device file close, the tape will be forward-spaced to the End of Tape (EOT) side of the nearest filemark, unless already there by virtue of reading the tape prior to the device close.

This means that an OPEN and CLOSE sequence with no reads will advance the tape by one filemark.

2. partition: (PARTITION)

Partition number.

Legal values: 0 – 63 – disk
0x0 – 0x86FF – tape
0x0 – 0xFF – DCI (DSCH Computer Interconnect)
0x0 – 0xFF – active DCI Adjunct
0x400 – 0x4FF – standby DCI Adjunct
null

The *logdev* record partition field can be used to control the options of the SCSI tape. Figure 6-1 shows the bit field layout of the SCSI tape partition number. Table 6-2 shows typical *logdev* values.

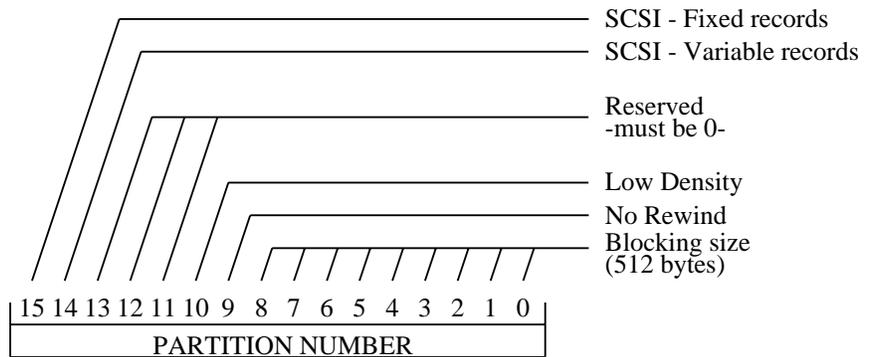


Figure 6-1. SCSI Tape Device (Partition Number Descriptor) — 3B21D Computer Only

Table 6-2. SCSI Tape Device logdev Values

Device Name	Device	Binary Partition Number	User Input	Options
/dev/mt0vl0	0	0100 0100 0000 0000	0x4400	SCSI, 1600 bpi, rewind, variable record size
/dev/mt0vh0	0	0100 0000 0000 0000	0x4000	SCSI, 6250 bpi, rewind, variable record size
/dev/mt0fh4n	0	1000 0010 0000 0100	0x8204	SCSI, 6250 bpi, no-rewind, 2K fixed record size
/dev/mt3fh4n	3	1000 0010 0000 0100	0x8204	SCSI, 6250 bpi, no-rewind, 2K fixed record size
/dev/mt3fh4	3	1000 0000 0000 0100	0x8004	SCSI, 6250 bpi, rewind, 2K fixed record size
/dev/mt0fh256	0	1000 0001 0000 0000	0x8100	SCSI, 6250 bpi, rewind, 128K fixed record size
/dev/mt0fl12	0	1000 0100 0000 1100	0x8400	SCSI, 1600 bpi, rewind, 6K fixed record size
/dev/mt0vl4	0	0100 0100 0000 0100	0x4404	SCSI, 1600 bpi, rewind, 2K-max variable record size
/dev/mt0vl4n	0	0100 0110 0000 0100	0x4604	SCSI, 1600 bpi, no-rewind, 2K-max variable record size
/dev/mt0vl12	0	0100 0100 0000 1100	0x440C	SCSI, 1600 bpi, rewind, 6K-max variable record size
/dev/mt0vl12n	0	0100 0110 0000 1100	0x460C	SCSI, 1600 bpi, no-rewind, 6K-max variable record size
/dev/mt0vh12	0	0100 0000 0000 1100	0x400C	SCSI, 6250 bpi, rewind, 6K-max variable record size
/dev/mt0vh12n	0	0100 0010 0000 1100	0x420C	SCSI, 6250 bpi, no-rewind, 6K-max variable record size
/dev/mt00	0	0000 0000 0000 0000	0x0000	SCSI/IOP, 6250 bpi, rewind, variable record size
/dev/mt04	0	0000 0000 0000 0100	0x0004	SCSI/IOP, 6250 bpi, no-rewind, variable record size
/dev/mt08	0	0000 0000 0000 1000	0x0008	SCSI/IOP, 1600 bpi, rewind, variable record size
/dev/mt0c	0	0000 0000 0000 1100	0x000C	SCSI/IOP, 1600 bpi, no-rewind, variable record size

The partition number for SCSI tape has further restrictions. See "Form Checks".

3. mdct_name: (MDCTNAME)

Name of associated mdct.

Legal values: printable characters, null

4. channel_id: (CHANNELID)

The decimal representation for a unique alphanumeric character identifying this device; for example, a = 97.

Legal values: 32 – 126, null

5. dcn: (DCN)

Device control number of the software device driver.

Legal values: -64 – 29, 255, null

6. driver_type: (DRIVERTYPE) Required

Type of driver.

Legal values: b – block
c – character
i – iop
p – fifo
r – record

7. gettyid: (GETTYID)

Name of the associated getty.

Legal values: printable characters, null

8. usr: (USR)

User access modes.

Legal values: --- r-- --X r-X
-w- rw- -wX rwx

Default value: rwx

9. grp: (GRP)

Group access modes.

Legal values:	---	r--	--X	r-X
	-w-	rw-	-WX	rwX

Default value: r-x

10. other: (OTHER)

Other access modes.

Legal values:	---	r--	--X	r-X
	-w-	rw-	-WX	rwX

Default value: r-x

Form Checks

```
if mdct_name is unknown
  then gettyid must_be unknown
  error: "gettyid must be null when mdct_name is null";

if channel_id is unknown
  then gettyid must_be unknown
  error: "both gettyid and channel_id must be either set or null" ;

if channel_id is known
  then gettyid must_be known
  error: "both gettyid and channel_id must be either set or null" ;

if partition is_in {regex("^0x[0-9a-fA-F]{1,4}$")}
  then do;
  /* insure reserve bits are not set */
  partition must_be_in {
    regex("^0x[0-7]{0,1}[0-9a-fA-F]{1,2}$"),
    regex("^0x[048cC]{0,1}[0-7][0-9a-fA-F]{1,2}$")}
    error: "reserve bits 11, 12, and 13 must
    not be set";

  /* insure fixed and the variable record bit are not both set */
  partition must_be_in {
    regex("^0x[0-9abAB]{0,1}[0-9a-fA-F]{1,3}$")}
    error: "the fixed and the variable record
    bits can not both be set" ;
```

```
/* insure blocking size is less than 0x101 (257) for fixed or variable
   record */
/* This is needed because the blocking size times the block size (512) */
/* cannot be larger than one segment (256 * 512) */
   partition must_be_in {
       regex("^0x[0-3]{0,1}[0-9a-fA-F]{1,3}$"),
       regex("^0x[4-9a-fA-F][02468aAcCeE][0-9a-fA-F]{2}$"),
       regex("^0x[4-9a-fA-F][1-7]0{2}$")}
       error: "blocking size must be less than 257 (0x101)
       for fixed or variable records";

/* insure blocking size is greater than 0 for fixed records */
   partition must_be_in {
       regex("^0x[0-7]{0,1}[0-9a-fA-F]{1,3}$"),
       regex("^0x[89a-fA-F][0-9a-fA-F]{2}[1-9a-fA-F]$")}
       error: "blocking size must be greater than 0
       for fixed records";

end;
```

mdct

This form represents a logical entity in the system. It contains information necessary to link a logical device to one or more physical devices.

Form Layout

mdct				(1/3)
Logical To Physical Device Transaction				(Recent Change and Verify)
1.mdct_name: _____	2.logical_device: _			
3.mdct_type:_	4.mdct_id_char:_			
5.active_devices				
6.complex_name	7.complex_number	8.unit_name	9.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	

mdct				(2/3)
10.backup_devices				
11.complex_name	12.complex_number	13.unit_name	14.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	
15.spare_devices				
16.complex_name	17.complex_number	18.unit_name	19.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	

mdct (3/3)

20.oos_devices
21.complex_name 22.complex_number 23.unit_name 24.unit_number

1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____

25.getty_id: _____

Definition of Fields

1. mdct_name: (MDCTNAME) Required

The symbolic name for an instance of a minor device chain table. This is a key field.

Legal values: printable characters

2. logical_device: (LOGICALDEVICE) Required

The two-character name identifying the complex with which the logical device is associated. The type of UCBs linked to this Minor Device Chain Table (MDCT) (UCBs under the same pseudo node have the same type).

Legal values:

ap	(attached processor community)
cu	(CU community)
dc	(Dual Serial Channel (DSCH) Computer Interconnect)
dk	(disk community)
fi	
io	(input/output community)
ms	(5ESS switch message)
rg	

3. mdct_type: (MDCTTYPE)

This entry tells how the mdct is to be used. It is a one-character field identifying the type of logical device.

Legal values:	a	[For Attached Processor System (APS) specific disk read.]
	d	[Minor Device Chain Table (MDCT) entry is used for diagnostic or physical device access.]
	e	[MDCT entry is for subunits (TN83)].
	f	(MDCT entry is used for moving head disk.)
	j	(MDCT entry is used for DCI adjunct use.)
	l	(MDCT entry is used for synchronous data link.)
	m	(MDCT entry is for magnetic tape unit.)
	null	(All other cases.)
	s	[MDCT entry is for Scanner and Signal Distributor (SCSD).]
	t	(MDCT entry is used for TTY.)
	T	(UCB chain contains top UCBs.)

4. mdct_id_char: (MDCTIDCHAR) Required

This is used to distinguish two entries with the same logical_device and mdct_type. The mdct_id_char, logical_device, and mdct_type in most cases identify a particular logical_device.

Legal values: printable characters

5. active_devices: (ACTIVEDEVICES)

Active ucbs. Identifies by name a maximum of four active physical devices for this logical device.

6. active_devices.complex_name:
(*ACTIVEDEVICES.COMPLEXNAME[ROW])

The first part of a four-part ucb identifier.

7. active_devices.complex_number:
(*ACTIVEDEVICES.COMPLEXNUMBER[ROW])

The second part of a four-part ucb identifier.

8. active_devices.unit_name: (*ACTIVEDEVICES.UNITNAME[ROW])

The third part of a four-part ucb identifier.

9. active_devices.unit_number: (*ACTIVEDEVICES.UNITNUMBER[ROW])

The fourth part of a four-part ucb identifier.

10. backup_devices: (BACKUPDEVICES)

Backup ucbs. Identifies by name a maximum of four backup physical devices for this logical device.

11. See previous fields 6 –9.

15. spare_devices: (SPAREDEVICES)

Spare ucbs. Identifies by name a maximum of four spare physical devices for this logical device.

⇒ NOTE:

This field is not currently used by Recent Change and Verify.

16. See previous fields 6 – 9.

20. oos_devices: (OOSDEVICES)

Out-of-service unit control blocks (UCBs). Identifies a maximum of four out-of-service devices for this logical device.

⇒ NOTE:

This field is not currently used by Recent Change and Verify.

21. See previous fields 6 – 9.

25. getty_id: (GETTYID)

Name of getty record.

Legal values: printable characters, null

Form Checks

```
case (logical_device);

  when {"dk"}: do;
    mdct_type must_be_in {"T","d", "f", "a", "m"}
    error: "mdct_type must be T, d, a, m, or f";
  end;

  when {"io"}: do;
    mdct_type must_be_in {"T", "t", "m", "d", "l", "s", "e"}
    error: "mdct_type must be T,t,m,d,l,s or e";
  end;

  when {"cu", "ap"}: do;
    mdct_type must_be_in {"T", "d"}
    error: "mdct_type must be T or d";
  end;

  when {"dc"}: do;
    mdct_type must_be_in {"T", "d", "j"}
    error: "mdct_type must be T, d or j";
  end;
end;

{select complex_name, complex_number, unit_name, unit_number from active_devices}
union
{select complex_name, complex_number, unit_name, unit_number from backup_devices}
union
{select complex_name, complex_number, unit_name, unit_number from spare_devices}
union
{select complex_name, complex_number, unit_name, unit_number from oos_devices}
must_be_unique
  error: "repeated occurrence of a device, each device must
  be unique";

if mdct_type is "j"
then do;
  count {select * from active_devices} = 1 thru 2
    error: "if mdct_type is 'j', then 1 or 2 active_devices must be specified";
end;
```

mhdopt

This form contains options which control operation of the Moving Head Disk (MHD).

This form is not implemented in all releases.

Form Layout

	mhdopt	(1/1)
Moving Head Disk Option Block Record		(Recent Change and Verify)
1.option_name:_____		2.elevator:_
3.readretry:_		4.rexflag:_
5.vtocwprt:_		6.rstsize:___
7.cmprsize:___		8.fvjbsize:___

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is a key field.

Legal values: 1 – 8 alphanumeric characters

2. elevator: (ELEVATOR)

Enable job optimization using an elevator algorithm. The elevator algorithm sorts jobs based on disk drive head movement. If enabled (y), the algorithm will be utilized for the MHD specifying this option record, unless it is disallowed by the elevator field in a *dfcopt* form. If disabled (n), the elevator algorithm will not be utilized.

This option is not implemented in all releases.

Legal values: y, n, null

3. readretry: (READRETRY)

Enable the Disk File Controller (DFC) to automatically retry a read job after a failure. If enabled (y), the read job retry feature will be used for the MHD specifying this option record, unless it is disallowed by the readretry field in a *dfcopt* form. If disabled (n), read retries will not be used.

This option is not implemented in all releases.

Legal values: y, n, null

4. rexflag: (REXFLAG)

Enable the DFC to routinely exercise the MHD by submitting read jobs to the MHD to verify the media and its operating condition. If enabled (y), the routine exercise jobs will be utilized for the MHD specifying this option record, unless it is disallowed by the rexflag field in a *dfcopt* form. If disabled (n), routine exercise jobs will not be utilized.

This option is not implemented in all releases.

Legal values: y, n, null

5. vtocwprt: (VTOCWPRT)

The Volume Table of Contents (VTOC) of the MHD will be write protected.

This option is not implemented in all releases.

Legal values: y, n, null

6. rstsize: (RSTSIZE)

Job size used by the MHD restore process. The number of 512-byte blocks to be read/written in each disk job.

This option is not implemented in all releases.

Legal values: 1 – 256 decimal, null

7. cmprsize: (CMPRSIZE)

Job size used by the MHD COMPARE/CORRECT process. The number of 512-byte blocks to be read and compared in each disk job.

This option is not implemented in all releases.

Legal values: 1 – 256 decimal, null

8. fvjbsize: (FVJBSIZE)

MHD format/verify job size. The number of 512-byte blocks to be formatted/verified in each operation.

For Storage Module Drive (SMD) MHDs, because of hardware restrictions, the job size used in the format operation will be rounded up to the nearest track boundary.

For Small Computer System Interface (SCSI) MHDs, the job size for a format cannot be specified; the complete disk will be formatted.

This option is not implemented in all releases.

Legal values: 1 – 1024 decimal, null

nphopt

This network protocol handler form contains physical device, switched connection, and packet layer options.

Form Layout

		nphopt	(1/6)
NPH Option Block Record		(Recent Change and Verify)	
1.option_name: _____			
2.speed: _____	3.duplex: _____	4.ds_type: _____	
5.carrier:_____	6.line_access: _____	7.mode_op:_____	
8.pdtimel: _____	9.pdtime2: _____	10.window:_____	
11.rexmit:_____	12.dcedte:_____		

		nphopt	(2/6)	
13.nphldtel	sid	tlim	telno	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	
17.security:_____	18.link_id:_____			
19.feid_len: _____	20.neid_len: _____	21.password_len: _____		
22.feid.fend				
1) _____	4) _____	7) _____	10) _____	13) _____
2) _____	5) _____	8) _____	11) _____	14) _____
3) _____	6) _____	9) _____	12) _____	15) _____

nphopt (3/6)

33.neid.nend

1) ___ 4) ___ 7) ___ 10) ___ 13) ___
2) ___ 5) ___ 8) ___ 11) ___ 14) ___
3) ___ 6) ___ 9) ___ 12) ___ 15) ___

44.pswrd.pwr

1) ___ 4) ___ 7) ___ 10) ___ 13) ___
2) ___ 5) ___ 8) ___ 11) ___ 14) ___
3) ___ 6) ___ 9) ___ 12) ___ 15) ___

55.perm_state: ___ 56.lana: _____ 57.estact_chnls: ___

58.packetid: ___ 59.fast_sel: _ 60.fast_accept: _

nphopt (4/6)

61.connection: ___ 62.num_pvcs: _ 63.losvc_lincome: ___

64.hisvc_lincome: ___ 65.losvc_2way: ___ 66.hisvc_2way: ___

67.losvc_loutgo: ___ 68.hisvc_loutgo: ___ 69.t20_timer: ___

70.t21_timer: ___ 71.t22_timer: ___ 72.t23_timer: ___

73.t24_timer: _ 74.t25_timer: ___ 75.t26_timer: ___

76.r20_counter: _ 77.r22_counter: _ 78.r23_counter: _ 79.nstd_ipsize: ___

80.nstd_opsize: ___ 81.nstd_iwsize: _ 82.nstd_owsize: _

nphopt (5/6)

83.pvcopts

	pnstdipsiz	pnstdopsiz	pnstdiwsiz	pnstdowsiz	user_name	spec_num
1)	—	—	—	—	_____	_____
2)	—	—	—	—	_____	_____
3)	—	—	—	—	_____	_____
4)	—	—	—	—	_____	_____
5)	—	—	—	—	_____	_____
6)	—	—	—	—	_____	_____
7)	—	—	—	—	_____	_____
8)	—	—	—	—	_____	_____
9)	—	—	—	—	_____	_____
10)	—	—	—	—	_____	_____
11)	—	—	—	—	_____	_____

nphopt (6/6)

83.pvcopts(cont)

	pnstdipsiz	pnstdopsiz	pnstdiwsiz	pnstdowsiz	user_name	spec_num
12)	—	—	—	—	_____	_____
13)	—	—	—	—	_____	_____
14)	—	—	—	—	_____	_____
15)	—	—	—	—	_____	_____
16)	—	—	—	—	_____	_____

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. This is a key field.

Legal values: alphanumeric characters

2. speed: (SPEED)

Data rate of associated communications equipment.

Legal values: 110, 300, 1200, 1800, 2400, 4800, 9600, 56000, 64000, null

3. duplex: (DUPLEX)

Full or half-duplex.

Legal values: full, half, null

4. ds_type: (DSTYPE)

PC modem type.

Legal values:	201CDDD	209A	NOPORT
	201CPL	212A	PL201C
	202/108	2024A	RS449
	208A	2048A	ITAPAC
	703COU	703CCUL	703CCUH
	703CDO	RS232	RS422
	RS423	208ADDD	CCITT
	208B	NODS	null

⇒ NOTE:

Multiple values may be considered equivalent in the database. Any legal value may be entered, but the value may not be displayed by Recent Change and Verify. The following table shows which values are displayed for the set of equivalent values.

Displayed Value	Equivalent Values
703COU	703COU 212A
703CCUL	703CCUL 208A
703CCUH	703CCUH 208B
703CDO	703CDO 209A
RS232	RS232 RS449
RS422	RS422 RS423 NODS NULL

5. carrier: (CARRIER)

Continuous or switched carrier.

Legal values: c, s, null

6. line_access: (LINEACCESS)

Type of line access.

Legal values: direct, private, noACU, ACU, null

7. mode_op: (MODEOP)

Mode operation. Block or send/receive mode of operation.

Legal values: b, s, null

8. pdtime1: (PDTIME1) Required

Maximum time to wait for level 2 protocol acknowledgment.

Values entered represent tenths of a second.

Legal values: 0 – 255

9. pdtime2: (PDTIME2) Required

Maximum time to allow data link to be idle.

Values entered represent tenths of a second.

Legal values: 0 – 255

10. window: (WINDOW) Required

Maximum number of I frames a station may have outstanding.

Legal values: 0 – 7

11. rexmit: (REXMIT) Required

Maximum number of retransmissions of an individual frame.

Legal values: 0 – 7

12. dcedte: (DCEDTE) Required

Data terminal or data circuit-terminating equipment.

Legal values: 0 (data terminal equipment)
1 (data circuit-terminating equipment)

13. nphldtel: (NPHLDTEL)

System id, time limit, telephone number.

nphldtel.sid: (*NPHLDTEL.SID[ROW])

System identifier.

Legal values: 0 – 65535, HEX, null

nphldtel.tlim: (*NPHLDTEL.TLIM[ROW])

Time limit.

Legal values: 0 – 255, HEX, null

nphldtel.telno: (*NPHLDTEL.TELNO[ROW])

Telephone number.

Legal values: alphanumeric characters, null

17. security: (SECURITY)

Check security information.

Legal values: y, n, null

18. link_id: (LINKID)

Link identification number.

Legal values: 0 – 7

Default value: 0

19. feid_len: (FEIDLEN)

Length of far-end system identifier.

Legal values: 0 – 15

Default value: 0

20. neid_len: (NEIDLEN)

Length of near-end system identifier.

Legal values: 0 – 15

Default value: 0

21. password_len: (PASSWORDLEN)

Length of password.

Legal values: 0 – 15

Default value: 0

22. feid.fend: (*FEID.FEND[ROW])

Far-end system identifier.

Legal values: 0 – 255, HEX, null

Default value: 0

33. neid.nend: (*NEID.NEND[ROW])

Near-end system identifier.

Legal values: 0 – 255, HEX, null

Default value: 0

44. pswrd.pwr: (*PSWRD.PWRD[ROW])

Password.

Legal values: 0 – 255, HEX, null

Default value: 0

55. perm_state: (PERMSTATE) Required

Dial-up permission state.

Legal values: allow (allow dial-up connection)
 condalw (if duplex and primary not active, then allow
 dial-up connection.)
 inhibit (inhibit dial-up connection)

56. lana: (LANA)

Lana. This is the Local Area Network Address (LANA), the data link's phone number.

Legal values: decimal digits, null

57. estact_chnls: (ESTACTCHNLS) Required

Estimated channels. This is the estimated number of active channels.

Legal values: 0 – 128

Default value: 0

58. packetid: (PACKETID)

Packet layer identifier.

Legal values: ITAPAC, OSINL, BTX25, X25, null

59. fast_sel: (FASTSEL) Required

Fast select facility. This facility applies only to Switched Virtual Circuit (SVC) channels. It allows the user to send one unit of data in the CALL REQUEST packet provided the network supports the fast select feature.

Legal values: y, n

Default value: n

60. fast_accept: (FASTACCEPT) Required

Fast select acceptance. This facility applies only to SVC channels. It allows the user to receive one unit of data in the INCOMING CALL packet.

Legal values: y, n

Default value: n

61. connection: (CONNECTION)

Connection type.

Legal values: DTE, DCE, null

Default value: DTE

62. num_pvcs: (NUMPVCS) Required

Number of Permanent Virtual Circuits (PVC) logical channels.

Legal values: 0 – 16

Default value: 0

63. losvc_1income: (LOSVC1INCOME)

Lowest one-way incoming logical channel number for SVCs.

Legal values: 0 – 4095, null

64. hisvc_1income: (HISVC1INCOME)

Highest one-way incoming logical channel number for SVCs.

Legal values: 0 – 4095, null

65. losvc_2way: (LOSVC2WAY)

Lowest two-way logical channel number for SVCs.

Legal values: 0 – 4095, null

66. hisvc_2way: (HISVC2WAY)

Highest two-way logical channel number for SVCs.

Legal values: 0 – 4095, null

67. losvc_1outgo: (LOSVC1OUTGO)

Lowest one-way outgoing logical number for SVCs.

Legal values: 0 – 4095, null

68. hisvc_1outgo: (HISVC1OUTGO)

Highest one-way outgoing logical channel number for SVCs.

Legal values: 0 – 4095, null

69. t20_timer: (T20TIMER) Required

Restart request response timer.

Legal values: 10 – 300

Default value: 180

70. t21_timer: (T21TIMER) Required

Call request response timer.

Legal values: 10 – 300

Default value: 200

71. t22_timer: (T22TIMER) Required

Reset request response timer.

Legal values: 10 – 300

Default value: 180

72. t23_timer: (T23TIMER) Required

Clear request response timer.

Legal values: 10 – 300

Default value: 180

73. t24_timer: (T24TIMER) Required

Window status transmission timer.

Legal values: 10 – 300

Default value: 60

74. t25_timer: (T25TIMER)

Data packet acknowledgment timer.

Legal values: 10 – 300, null

Default value: 150

75. t26_timer: (T26TIMER) Required

Interrupt response timer.

Legal values: 10 – 300

Default value: 180

76. r20_counter: (R20COUNTER) Required

Restart request retransmission count.

Legal values: 1 – 5

Default value: 1

77. r22_counter: (R22COUNTER) Required

Reset request retransmission count.

Legal values: 1 – 2

Default value: 1

78. r23_counter: (R23COUNTER) Required

Clear request retransmission count.

Legal values: 1 – 10

Default value: 1

79. nstd_ipsize: (NSTDIPSIZE) Required

Input packet size for the SVCs.

Legal values: 128, 256, 512

Default value: 128

80. nstd_opsize: (NSTDOPSIZE) Required

Output packet size for the SVCs.

Legal values: 128, 256, 512

Default value: 128

81. nstd_iwsize: (NSTDIWSIZE) Required

Input window size for the SVCs.

Legal values: 2 – 7

Default value: 2

82. nstd_owsize: (NSTDOWSIZE) Required

Output window size for the SVCs.

Legal values: 2 – 7

Default value: 2

83. pvcopts: (PVCOPTS)

Pvcopts are the options for PVCs.

pvcopts.pnstdipsiz (*PVCOPTS.PNSTDIPSIZ[ROW])

Input packet size.

Legal values: 128, 256, 512, null

pvcopts.pnstdopsiz (*PVCOPTS.PNSTDOPSIZ[ROW])

Output packet size.

Legal values: 128, 256, 512, null

pvcopts.pnstdiwsiz (*PVCOPTS.PNSTDIEWSIZ[ROW])

Input window size.

Legal values: 2 – 7, null

pvcopts.pnstdowsiz (*PVCOPTS.PNSTDOWSIZE[ROW])

Output window size.

Legal values: 2 – 7, null

pvcopts.user_name (*PVCOPTS.USERNAME[ROW])

Application tag name of the higher level user.

Legal values: printable characters, null

pvcopts.spec_num (*PVCOPTS.SPECNUM[ROW])

Special PVC number for the PVC.

Legal values: decimal digits, null

List Element Checks

```
on pvcopts element entry:
do;
  if pnstdipsiz is known then user_name must_be known
    error: "when pnstdipsiz is known user_name must be known";
  if pnstdipsiz is known then spec_num must_be known
    error: "when pnstdipsiz is known spec_num must be known";
  if pnstdopsiz is known then user_name must_be known
    error: "when pnstdopsiz is known user_name must be known";
  if pnstdopsiz is known then spec_num must_be known
    error: "when pnstdopsiz is known spec_num must be known";
  if pnstdiwsiz is known then user_name must_be known
    error: "when pnstdiwsiz is known user_name must be known";
  if pnstdiwsiz is known then spec_num must_be known
    error: "when pnstdiwsiz is known spec_num must be known";
  if pnstdowsiz is known then user_name must_be known
    error: "when pnstdowsiz is known user_name must be known";
  if pnstdowsiz is known then spec_num must_be known
    error: "when pnstdowsiz is known spec_num must be known";

end;
```

Form Checks

```
count {select * from pvcopts} = num_pvcs
  error: "Number of list items in pvcopts must equal the
  number of PVC's in field 62."

if losvc_lincome is known
  then if num_pvcs is_greater_than 0
    then losvc_lincome must_be_greater_than num_pvcs
      error: "losvc_lincome must be greater than num_pvcs";

if losvc_2way is known
  then if hisvc_lincome is known
    then losvc_2way must_be_greater_than hisvc_lincome
      error: "losvc_2way must be greater than hisvc_lincome";
  else if num_pvcs is_greater_than 0
    then losvc_2way must_be_greater_than num_pvcs
      error: "losvc_2way must be greater than num_pvcs";

if losvc_loutgo is known
  then if hisvc_2way is known
    then losvc_loutgo must_be_greater_than hisvc_2way
      error: "losvc_loutgo must be greater than hisvc_2way";
  else if hisvc_lincome is known
    then losvc_loutgo must_be_greater_than hisvc_lincome
```

```
        error: "losvc_loutgo must be greater than hisvc_lincome";
    else if num_pvcs is_greater_than 0
        then losvc_loutgo must_be_greater_than num_pvcs
            error: "losvc_loutgo must be greater than num_pvcs";

if losvc_lincome is known
    then hisvc_lincome must_be known
        error: "hisvc_lincome must be set when losvc_lincome is set";

if losvc_2way is known
    then hisvc_2way must_be known
        error: "hisvc_2way must be set when losvc_2way is set";

if losvc_loutgo is known
    then hisvc_loutgo must_be known
        error: "hisvc_loutgo must be set when losvc_loutgo is set";

if hisvc_lincome is known
    then losvc_lincome must_be known
        error: "losvc_lincome must be set when hisvc_lincome is set";

if hisvc_2way is known
    then losvc_2way must_be known
        error: "losvc_2way must be set when hisvc_2way is set";

if hisvc_loutgo is known
    then losvc_loutgo must_be known
        error: "losvc_loutgo must be set when hisvc_loutgo is set";

if losvc_lincome is known
    then hisvc_lincome must_not_be_less_than losvc_lincome
        error: "hisvc_lincome must not be less than losvc_lincome";

if losvc_2way is known
    then hisvc_2way must_not_be_less_than losvc_2way
        error: "hisvc_2way must not be less than losvc_2way";

if losvc_loutgo is known
    then hisvc_loutgo must_not_be_less_than losvc_loutgo
        error: "hisvc_loutgo must not be less than losvc_loutgo";

if losvc_lincome is unknown
    then num_pvcs + hisvc_2way - losvc_2way + hisvc_loutgo
        - losvc_loutgo <= 127
        error: "sum of channels cannot exceed 127";
else if losvc_2way is unknown
    then num_pvcs + hisvc_lincome - losvc_lincome + hisvc_loutgo -
        losvc_loutgo <= 127
        error: "sum of channels cannot exceed 127";
```

```
else if losvc_loutgo is unknown
  then num_pvcs + hisvc_lincome - losvc_lincome + hisvc_2way
    - losvc_2way <= 127
    error: "sum of channels cannot exceed 127";
else num_pvcs + hisvc_lincome + hisvc_2way + hisvc_loutgo
  - losvc_lincome - losvc_2way - losvc_loutgo <= 127
  error: "sum of channels cannot exceed 127";
if losvc_lincome is unknown
  then if losvc_2way is unknown
    then num_pvcs + hisvc_loutgo - losvc_loutgo <= 127
    error: "sum of channels cannot exceed 127";
if losvc_lincome is unknown
  then if losvc_loutgo is unknown
    then num_pvcs + hisvc_2way - losvc_2way <= 127
    error: "sum of channels cannot exceed 127";
if losvc_2way is unknown
  then if losvc_loutgo is unknown
    then num_pvcs + hisvc_lincome - losvc_lincome <= 127
    error: "sum of channels cannot exceed 127";
```

objectid

This form is used to provide information about certain General File Server (GFS) and File Transfer Access and Management (FTAM) protocol attributes that are defined to be of type objectid. An objectid (object identifier) provides the naming of objects in a hierarchical structure. This hierarchical structure is called the object identifier tree. Each branch identifies an authority permitted to allocate branches in the tree. Each branch of the tree is identified by an integer value; therefore, the *objectid* form is just a set of integer values.

Form Layout

Object Identifier Record	objectid	(1/1)
		(Recent Change and Verify)
1.objectidname: _____		
2.object_identifiers:		
1) _____	4) _____	7) _____
2) _____	5) _____	8) _____
3) _____	6) _____	9) _____
		10) _____
		11) _____
		12) _____

Definition of Fields

1.objectidname: (OBJECTIDNAME) Required

Object identifier name. This is a key field.

Legal values: 1 – 8 characters

2.object_identifiers: (OBJECTIDENTIFIERS)

The set of object identifier component values. Each element is an object identifier component. This field consists of 12 occurrences of a positive integer. This field cannot be null. If an object identifier is unused, the field should be a -1 (MAXINT). These object identifiers must be assigned sequentially.

Legal values: 0 – (MAXINT-1), MAXINT (denoting -1)

Default value: MAXINT (MAXINT is being used to represent -1)

Form Checks

The following form checks are required for this form:

```
objectid: form;

/* The objectid form is designed to provide information about
   certain GFS and FTAM attributes that are defined to be
   of type object identifier. An object identifier provides
   the naming of objects in a hierarchical structure. This
   hierarchical structure is called the object identifier tree.
   Each branch identifies an authority permitted to allocate
   branches in the tree. Each branch of the tree is identified
   by an integer value.
*/

full_name: Object Identifier Record;

key_fields:
{
    objectidnamedomain(objidname)/* Object identifier name */
};

data_fields:
{
    object_idents[12]/* Set of Object Identifiers*/
    {
        obj_id /* Object Identifier */
    }
};

list_breakups:
    object_idents:3,3,3,3;

form_checks:

    count {select * from object_idents} > 0
        error: "At least one object identifier component must be
            specified.";
```

pphopt

This form is used to enter option information for the Associated Control Service Element/Presentation Layer (ACSE/PL) protocol handler.

Form Layout

ACSE/PL Protocol Handler Option Record		pphopt	(1/1)
			(Recent Change and Verify)
1.pphoptname: _____		2.max_assoc: _____	
3.num_psap: _____		4.conn_buff: _____	
5.uabort_buff: _____		6.resync_buff: _____	
7.indic_buff: _____			

Definition of Fields

1.pphoptname: (PPHOPTNAME) Required

The option name pertaining to the ACSE/PL. This is a key field.

Legal values: 1 – 8 characters

Default value: no default

2.max_assoc: (MAXASSOC)

The maximum value of simultaneous associations allowed for all applications associated with this *pphopt* record. If a user (or a remote side) tries to establish an association when the current count is already at the maximum, the association request is refused. There is a one-to-one relationship between an association, a presentation connection, a session connection, and a transportation connection. The sum of max_assoc of all *pphopt* records must not be greater than 200.

Legal values: 0 – 200

Default value: 32

3.num_psap: (NUMPSAP)

The number of Presentation Service Access Points (PSAPs) to allocate.
The sum of num_psap of all *pphopt* records must not be greater than 32.

Legal values: 1 – 32

Default value: 2

4.conn_buff: (CONNBUF)

The number of buffers in the pool of connect Protocol Data Unit (PDU) buffers. This number corresponds to how many connection establishment procedures (combined total of requests and responses) can be done simultaneously.

Legal values: 1 – 200

Default value: 8

5.uabort_buff: (UABORTBUF)

The number of buffers in the pool of user abort PDU buffers. This number corresponds to how many incoming user abort PDUs may be held simultaneously pending receipt by the user plus the number of outgoing user abort PDUs waiting to be sent.

Legal values: 1 – 200

Default value: 12

6.resync_buff: (RESYNCBUFF)

The number of buffers in the pool of incoming resynchronize PDU buffers. This corresponds to the number of incoming resynchronize PDUs which may be held simultaneously pending receipt of the user. It also determines the size of the pool of incoming resynchronize acknowledge PDU buffers.

Legal values: 1 – 200

Default value: 8

7.indic_buff: (INDICBUFF)

The maximum number of received indications per connection that can be held simultaneously pending receipt by the user.

Legal values: 2 – 200

Default value: 10

Form Checks

```
(conn_buff <= max_assoc)
  error: "conn_buff must not be greater than max_assoc";

(uabort_buff <= max_assoc)
  error: "uabort_buff must not be greater than max_assoc";

(resync_buff <= max_assoc)
  error: "resync_buff must not be greater than max_assoc";
```

pphuser

This form is used to enter user information for the Associated Control Service Element/Presentation Layer (ACSE/PL).

Form Layout

pphuser (1/2)
ACSE/PL Protocol Handler User Record (Recent Change and Verify)

1.attach_id: _____ 2.loc_ssap_sel: _____

3.loc_psap_sel loc1 loc2 loc3 loc4

8.pphoptname: _____

9.acse_version: ____ 10.pres_version: ____

11.sess_version: ____

12.mode_select: ____

pphuser (2/2)

13.processnm: _____

14.ext_qos: ____

15.impl_info: _____

Definition of Fields

1.attach_id: (ATTACHID) Required

The application entity qualifier of a local entity. This is a key field.

Legal values: 1 – MAXINT

2.loc_ssap_sel: (LOCSSAPSEL) Required

The session selector of the local entity. Refer to International Standards Organization (ISO) 7498-3 for more details. The loc_ssap_sel must be unique across all *pphuser* records.

⇒ NOTE:

Once a *pphuser* form has been inserted in the ecd, the loc_ssap_sel field cannot be updated using recent change. This field is tied to the attach_id field and cannot be changed without deleting the entire record and reinserting with a new loc_ssap_sel value.

Legal values: 1 – 16 printable characters

3.loc_psap_sel.loc1 – loc4: (LOCPSAPSEL.LOC1 – LOCPSAPSEL.LOC4)

The presentation selector of the local entity. This is an array (1 to 4 hexadecimal characters) that specified the presentation selector of the local entity. A blank field in the middle of the address translates to zero. Refer to ISO 7498-3 for more details. The loc_psap_sel, when non-null, must be unique across all *pphuser* records.

Legal values: 1 – 4 hexadecimal characters, null (blank)

Default value: null (blank)

8.pphoptname: (PPHOPTNAME) Required

The name of the *pphopt* record associated with this *pphuser* record.

Legal values: 1 – 8 characters

9.acse_version: (ACSEVERSION)

The desired implementation version of the ACSE provider. Refer to International Telegraph and Telephone Consultative Committee (CCITT) X.227 for more details.

Legal values: 1 (AP_ACSEVER1)

Default value: 1

10.pres_version: (PRESVERSION)

The desired implementation version of the presentation layer provider.
Refer to CCITT X.226 for more details.

Legal values: 1 (AP_PRESVER1)

Default value: 1

11.sess_version: (SESSVERSION)

The desired implementation version of the session layer provider.

Legal values: a (AP_SESSVERA), or b (AP_SESSVERB)

Default value: b

12.mode_select: (MODESELECT)

The ACSE and presentation service mode. Refer to CCITT X.216 and
X.217 for more details.

Legal values: n (AP_NORMAL)

Default value: n

13.processnm: (PROCESSNM)

The pathname of the process to be created upon receipt of an incoming
association request.

Legal values: 0 – 63 characters

Default value: null (blank)

14.ext_qos: (EXTQOS)

A flag that indicates the desired use of extended control quality of service
(use of transport expedited service to support session layer resynchronize
and abort services). A value of "y" indicates that extended control quality
of service will be proposed to the session layer on connect requests
initiated by this PPH user.

⇒ NOTE:

The session layer will automatically propose extended control of quality of service on outgoing connect requests if the session layer expedited data functional unit is proposed by the session service user during connection establishment, regardless of the value of this field. Refer to CCITT X.215 for more details.

Legal values: y or n

Default value: y

15.impl_info: (IMPLINFO)

Information specific to the implementation of the ACSE protocol that may be exchanged with the remote peer. Refer to CCITT X.227 for more details.

Legal values: 0 – 64 printable characters

Default value: null (blank)

rcvparams

This form contains Recent Change and Verify (RC/V) parameter information. This form cannot be manually invoked. It is invoked automatically, when appropriate, by RC/V.

Form Layout

```
rcvparams (1/1)
RCV ECD Parameter Info (Execute Only)
1.database_name:_____
2.reviewonly:_____ 3.journaling:_____
4.print_file:_____
5.echo_file:_____
6.sysgen:_____ 7.flat_database:_____ 8.hlmode:_____
9.default_dir:_____
```

Definition of Fields

1.database_name: Required

Database name. This is a key field.

Legal values: printable characters

2.reviewonly:

Review only.

Legal values: y, n

Default value: y

3.journaling:

Journaling.

Legal values: y, n

Default value: y

4.print_file:

Print file.

Legal values: printable characters, null

5.echo_file:

Echo filename.

Legal values: printable characters, null

6.sysgen:

Legal values: y, n

Default value: n

7.flat_database:

Flat ECD.

Legal values: y, n

Default value: n

8.hlmode:

Is high-level mode required?

Legal values: y, n

Default value: n

9.default_dir:

Enter default file directory (if other than official).

Legal values: printable, null

runa

This form contains the information for the network protocol handler to select the Data Terminal Equipment (DTE) that can establish a connection with the remote network address.

Form Layout

```
runa (1/1)
Packet Layer Remote User Network (Recent Change and Verify)

1.r_netaddr: _____

2.pcsd_data_link

l_netaddr      l_netaddr      l_netaddr      l_netaddr
1) _____  9) _____  17) _____  25) _____
2) _____  10) _____ 18) _____  26) _____
3) _____  11) _____ 19) _____  27) _____
4) _____  12) _____ 20) _____  28) _____
5) _____  13) _____ 21) _____  29) _____
6) _____  14) _____ 22) _____  30) _____
7) _____  15) _____ 23) _____  31) _____
8) _____  16) _____ 24) _____  32) _____
```

Definition of Fields

1. r_netaddr: (RNETADDR) Required

r_netaddr is the key field for runa. This field is the remote network address.

Legal values: decimal digits

2. pcsd_data_link: (PCSDDATALINK)

pcsd_data_link.l_netaddr: (*PCSDDATALINK.LNETADDR[ROW])

Name of the network protocol option blocks (nphopt) of the DTEs that can be used to access the remote user.

Legal values: printable characters, null

Form Checks

```
count {select l_netaddr from pcsd_data_link} > 0
    error: "a minimum of one l_netaddr must be specified";

{select l_netaddr from pcsd_data_link} must_be_unique
    error: "l_netaddr values must be unique";
```

scsdbody

This form gives information for one logical group of a Scanner and Signal Distributor (SCSD) option block.

Form Layout

SCSD Option Block Body		scsdbody	(1/1)
			(Recent Change and Verify)
1.option_name: _____	2.scsdname: _____	3.dupid:_	4.sc_class: _____
5.complex_name: _____	6.complex_number: _____		
7.unit_name: _____	8.unit_number: _____		
9.sc_list.sc_entry	11.sd_list.sd_entry		
1) _____	1) _____		
2) _____	2) _____		
3) _____	3) _____		
4) _____	4) _____		
5) _____	5) _____		
6) _____	6) _____		

Definition of Fields

1. option_name: (OPTIONNAME) Required

SCSD option block name. This is a key field.

Legal values: alphanumeric characters

2. scsdname: (SCSDNAME) Required

The name of this logical group. This is a key field.

Legal values: alphanumeric characters

3. dupid: (DUPID) Required

Duplex identification number.

Legal values: 0, 1

4. sc_class: (SCCLASS)

The scan class to which this logical group belongs.

Legal values: ALRM (alarm monitor)
FALM (fan alarm)
null
NULL (not monitored by any process)
PDFA (power distributing frame)
PRSW (power switch monitor)
PTSW (port switch monitor)
PWRBKUP (power backup)
USER1
USER2
USER3
USER4
USER5
USER6
USER7
USER8

5. complex_name: (COMPLEXNAME)

The first part of a four-part ucb identifier.

6. complex_number: (COMPLEXNUMBER)

The second part of a four-part ucb identifier.

7. unit_name: (UNITNAME)

The third part of a four-part ucb identifier.

8. unit_number: (UNITNUMBER)

The fourth part of a four-part ucb identifier.

9. sc_list.sc_entry (*SCLIST.SCENTRY[ROW])

Scan points monitored by this logical group.

Legal values: 0 – 47, 0x0 – 0x2F, null

11. sd_list.sd_entry (*SDLIST.SDENTRY[ROW])

Distributor points monitored by this logical group.

Legal values: 0 – 31, 0x0 – 0x1F, null

Form Checks

```
option_name must_not_be blank  
error: "option_name must not be blank";
```

```
scsdname must_not_be blank  
error: "scsdname must not be blank";
```

scsdmap

This is a review-only form containing data about a Scanner and Signal Distributor (SCSD) option block. An SCSD option block describes one SCSD Peripheral Controller (PC) in the Input/Output Processor (IOP) and consists of one *scsdopt* form and multiple *scsdbody* forms, one for each hardware device monitored; for example, Moving Head Disk (MHD), Control Unit (CU), Magnetic Tape (MT), etc.

Form Layout

```
scsdmap (1/3)
SCSD Scan Point Map (Review Only)
1.option_name: _____
2.sc_pnt
  lgroup s_equipage inhibit state  lgroup s_equipage inhibit state
0) _____ 10) _____
1) _____ 11) _____
2) _____ 12) _____
3) _____ 13) _____
4) _____ 14) _____
5) _____ 15) _____
6) _____ 16) _____
7) _____ 17) _____
8) _____ 18) _____
9) _____ 19) _____
```

scsdmap (2/3)

```

2.sc_pnt(cont)
  lgroup s equipage inhibit state    lgroup s equipage inhibit state
20) ___ - - - - 32) ___ - - - -
21) ___ - - - - 33) ___ - - - -
22) ___ - - - - 34) ___ - - - -
23) ___ - - - - 35) ___ - - - -
24) ___ - - - - 36) ___ - - - -
25) ___ - - - - 37) ___ - - - -
26) ___ - - - - 38) ___ - - - -
27) ___ - - - - 39) ___ - - - -
28) ___ - - - - 40) ___ - - - -
29) ___ - - - - 41) ___ - - - -
30) ___ - - - - 42) ___ - - - -
31) ___ - - - - 43) ___ - - - -
    
```

scsdmap (3/3)

```

2.sc_pnt(cont)
  lgroup s equipage inhibit state    lgroup s equipage inhibit state
44) ___ - - - - 46) ___ - - - -
45) ___ - - - - 47) ___ - - - -

33.num_lgrps: ___

34.sc_grpdata
  sc_num sd_num  sc_num sd_num  sc_num sd_num  sc_num sd_num
0) ___ ___  5) ___ ___  10) ___ ___  15) ___ ___
1) ___ ___  6) ___ ___  11) ___ ___  16) ___ ___
2) ___ ___  7) ___ ___  12) ___ ___  17) ___ ___
3) ___ ___  8) ___ ___  13) ___ ___  18) ___ ___
4) ___ ___  9) ___ ___  14) ___ ___  19) ___ ___
    
```

Definition of Fields

1. option_name: (OPTIONNAME) Required

SCSD option block name. This is a key field.

Legal values: alphanumeric characters

2. sc_pnt:

Information about each of the physical scan points.

lgroup

Logical group number (from field 34). Indicates which logical group monitors this physical scan point.

Legal values: 0 – 20, null (scan point not used)

s_equipage

Initial inhibit state.

Legal values:

- y this point does not monitor any device
- n this point monitors a device
- null scan point not used

inhibit

Current inhibit state.

Legal values:

- y do not report state transitions
- n report state transitions
- null scan point not used

state

Boot inhibit state. Inhibit state set by SCSD controller at initialization.

Legal values:

- y do not report state transitions
- n report state transitions
- null scan point not used

33. num_lgrps:

The number of logical groups (that is, *scsdbody* forms) associated with this SCSD option block.

Legal values: 0 – 20

34. sc_grpdata:

Information about a single logical group.

sc_num

Number of scan points in this logical group.

Legal values: 0 – 128, null (logical group not in use)

sd_num

Number of distributor points in this logical group.

Legal values: 0 – 128, null (logical group not in use)

scsdopt

This form gives options for the Scanner and Signal Distributor (SCSD) Peripheral Controller (PC) in the Input/Output Processor (IOP). This information is used by the SCSD administrator to perform physical-to-logical device translation and to keep the status of the scan points.

Form Layout

SCSD Option Block Head	scsdopt	(1/1) (Recent Change and Verify)
1.option_name: _____		

Definition of Fields

1. option_name: (OPTIONNAME) Required

SCSD option block name. This is a key field.

Legal values: alphanumeric characters

simcntl

This form contains System Integrity Monitor (SIM) control information.

Form Layout

```
                                simcntl      (1/1)
System Integrity Monitor Control Record      (Recent Change and Verify)

  1.ksimcntl: ____      2.asimprc: ____
  3.app_wait: __      4.mc_wait: __
  5.kmax: __      6.kmin: __
  7.smax: __      8.smin: __
  9.sysinit: __      10.maxspaud: __
 11.kpnseg: __      12.aud_alw: __
 13.ovldinit: ____
```

Definition of Fields

1. ksimcntl: (KSIMCNTL) Required

Simcntl record key.

Legal value: SIMCNTL

Default value: SIMCNTL

2. asimprc: (ASIMPRC)

Audit SIM process link.

Legal value: SIMPRC

Default value: SIMPRC

3. app_wait: (APPWAIT)

Full-config wait time for Application Integrity Monitor (AIM) (seconds).

Legal values: 0 – 3600

Default value: 0

4. mc_wait: (MCWAIT)

Min-config wait time for AIM (seconds).

Legal values: 0 – 3600

Default value: 0

5. kmax: (KMAX)

100 times maximum percentage of system time that is to be used by kernel.

Legal values: 1 – 10000

Default value: 500

6. kmin: (KMIN)

100 times minimum percentage of system time to be used by kernel.

Legal values: 1 – 10000

Default value: 50

7. smax: (SMAX)

100 times maximum percentage of system time to be used by supervisor.

Legal values: 1 – 10000

Default value: 500

8. smin: (SMIN)

100 times minimum percentage of system time to be used by supervisor.

Legal values: 1 – 10000

Default value: 50

9. sysinit: (SYSINIT)

Operating system initialization period (seconds).

Legal values: 0 – 255

Default value: 30

10. maxspaud: (MAXSPAUD)

Maximum simultaneous requested supervisory audits.

Legal values: 0 – 4

Default value: 2

11. kpnsseg: (KPNSEG)

Number of manual kernel audit segments.

Legal values: 0 – 15

Default value: 2

12. aud_alw: (AUDALW)

Wait time to allow audits in seconds.

Legal values: 0 – 32767, 0x0 – 0x7FFF

Default value: 900

13. ovlidinit: (OVLIDINIT)

Wait time to initialize overload control in seconds.

Legal values: 0 – 65535

Default value: 120

Form Checks

```
kmin <= kmax  
  error: "kmin must be less than or equal to kmax";  
  
smin <= smax  
  error: "smin must be less than or equal to smax";
```

sphopt

This form enters information for the International Telegraph and Telephone Consultative Committee (CCITT) protocols session layer handler.

Form Layout

```
                                sphopt                (1/1)
Session Protocol Handler Option Record      (Recent Change and Verify)

1.sphoptname:_____          2.local_tsap_sel:____
3.timer_exp:____              4.max_sc:____
5.num_ssap:____               6.reuse:_____
```

Definition of Fields

1. sphoptname: (SPHOPTNAME) Required

Session layer form identifier. This is a key field.

Legal values: SPHOPT

Default value: SPHOPT

2. local_tsap_sel: (LOCALTSAPSEL) Required

Local Transport Service Access Point (TSAP) selector.

Legal values: 1 – 4 decimal characters

3. timer_exp: (TIMEREXP) Required

Timer expiration. The length of time after which the disconnect and abort timer TIM expires.

Legal values: 1 – 600 seconds

Default value: 300

4. max_sc: (MACSC) Required

Maximum number of session connections.

Legal values: 0 – 200, decimal

Default value: 32

5. num_ssap: (NUMSSAP) Required

Number of Session Service Access Points (SSAPs).

Legal values: 0 – 32, decimal

Default value: 16

6. reuse: (REUSE) Required

Transport service reuse indicator.

Legal values: y or n

Default value: n

Form Checks

```
if max_sc > 0 then num_ssap > 0
error: "num_ssap must be greater
than 0 to allow session connections";
```

splrinfo

This form contains miscellaneous spooler information.

Form Layout

```

                                     splrinfo  (1/2)
Miscellaneous Spooler Information      (Recent Change and Verify)
1.ksplrinfo:_____                    2.shell_interface:___
3.capability_mode:___  4.time_stamp:_____  5.omdb_mode___
6.report_header:
_____
7.report_trailer:
_____
8.solicited_header:
_____
```

```

                                     splrinfo  (2/2)
9.default_language:___  10.report_logging:_____
```

Definition of Fields

1. ksplrinfo: (KSPLRINFO) Required

This is a key field. Because there is only one *splrinfo* form in a database, this field always contains "*splrinfo*."

Legal values: SPLRINFO

Default value: SPLRINFO

2. shell_interface: (SHELLINTERFACE) Required

Shell interface selection. This field can be set to Man-Machine Language (MML) or Program Documentation Standard (PDS). The craft spooler and shell reads this field to determine the proper Input/Output (I/O) syntax.

Legal values: PDS, MML

Default value: PDS

3. capability_mode: (CAPABILITYMODE) Required

Craft shell input message catalogue capability mode. Whether this field is FULL (complete capability) or PART (partial capability) determines how the shell responds to an input message when it is unable to find a matching catalogue entry.

Legal values: FULL, PART

Default value: FULL

4. time_stamp: (TIMESTAMP) Required

Time stamp flag. If this field is set to msgon or msgoff, it will override the requested time stamp of output messages for all devices. If the field is set to "dontcare," the force time stamp specification for each device will be used. time_stamp overrides the device form and user specified control string.

Legal values: dontcare, msgoff, msgon

Default value: dontcare

5. omdb_mode: (OMDBMODE) Required

Number of output message database segments that will reside in main memory.

Legal values: 1 – 15

Default value: 1

6. report_header: (REPORTHEADER) Required

Layout of the output report header. This field may include any printable characters plus control strings that the spooler will replace with designated variables, including:

Control String	Description
%b	basic time stamp (minutes after the hour)
%c	command number
%d	date stamp (in the format specified in the <i>eaiopt</i> record)
%g	global report sequence number
%i	identity (person or terminal)
%k	Output Message Database (OMDB) key
%m	report sequence number per message class
%n	dialogue number
%p	process id of sending process
%r	report destination (message class)
%s	system name
%t	time
%u	utility id of sending process

Legal values: printable characters

Default value: \n

7. report_trailer: (REPORTTRAILER) Required

Layout of the output report trailer. See description of report header in "6.report_header: (REPORTHEADER)."

Legal values: printable characters

Default value: 'd 't #'g\n

8. solicited_header: (SOLICITEDHEADER)

Layout of solicited report header. See description of report header in "6.report_header: (REPORTHEADER)."

The *UNIX* RTR operating system craft shell does not use this field.

Legal values: printable characters or null

9. default_language: (DEFAULTLANGUAGE)

Indicates whether the English or National language portion of the Output Message Database (OMDB) is kept in core.

Legal values: ENGL, NATL

Default value: ENGL

10. report_logging: (REPORTLOGGING)

Indicates whether output logging is done and what kind of report logging is done.

Legal values: NONE, ALARM, SPONT

Default value: NONE

tphopt

This form enters option information for the International Telegraph and Telephone Consultative Committee (CCITT) transport protocol handler.

Form Layout

```

                                     tphopt  (1/1)
Transport Protocol Handler Option Block Record (Recent Change and Verify)
1.option_name:_____ 2.t_type:____
3.lnsap:_____
4.max_nc:___ 5.max_tc:___
6.max_tsap:___ 7.max_mux:___
8.o_setup:___ 9.o_data:___ 10.o_misc:___
11.ts1_timer:___ 12.ts2_timer:___ 13.ts3_timer:___
14.ttr_timer:___ 15.twr_timer:___ 16.tf_timer:___
```

Definition of Fields

1. option_name: (OPTIONNAME) Required

Transport protocol handler option block name. This is a key field.

Legal values: 1 – 8 alphanumeric characters

2. t_type: (TTYPER) Required

Transport entity type.

Legal values: Open Systems Interconnection (OSI)

Default value: OSI

3. Insap: (LNSAP) Required

Local Network Service Access Point (NSAP) identifier (maximum 40 bytes).

Legal values: 1 – 40 decimal characters

4. max_nc: (MAXNC) Required

Maximum number of network connections.

Legal values: 0 – 64, decimal for *UNIX*® Real-Time Reliable (RTR) operating system Release 1

Legal values: 0 – 128, decimal for *UNIX* RTR operating system Release 6

5. max_tc: (MAXTC) Required

Maximum number of transport connections.

Legal values: 0 – 64, decimal for *UNIX* RTR operating system Release 1

Legal values: 0 – 128, decimal for *UNIX* RTR operating system Release 6

6. max_tsap: (MAXTSAP) Required

Maximum number of transport session access points.

Legal values: 1 – 64, decimal

⇒ NOTE:

The maximum number of Transport Service Access Points (TSAPs) allowed is 64, but the code implementation for TSAPs only allocates space for 16 TSAPs. Regardless of what the user enters, only 16 TSAPs will be gotten.

7. max_mux: (MAXMUX) Required

Maximum level of multiplexing.

Legal values: 1 – 8, decimal

8. o_setup: (OSETUP) Required

Maximum number of outgoing header buffers for connect request and connect confirm.

Legal values: 1 – 128, decimal

9. o_data: (ODATA) Required

Maximum number of outgoing header buffers for Data Terminal (DT) Transport Protocol Data Unit (TPDU) per TC.

Legal values: 1 – 16, decimal

10. o_misc: (OMISC) Required

Maximum outgoing header buffers for AK, RJ, and EA TPDU per TC.

Legal values: 1 – 16, decimal

11. ts1_timer: (TS1TIMER) Required

The timer is started after a connect request TPDU is sent.

Legal values: 0 – 300 seconds

Default value: 300

12. ts2_timer: (TS2TIMER) Required

The timer is used to time out a disconnect request.

Legal values: 0 – 300 seconds

Default value: 300

13. ts3_timer: (TS3TIMER) Required

Time in seconds to try reassignment/resynchronization.

Legal values: 0 – 300 seconds

Default value: 300

14. ttr_timer: (TTRTIMER) Required

Timer in seconds to try reassignment/resynchronization.

Legal values: 0 – 300 seconds

Default value: 300

15. twr_timer: (TWRTIMER) Required

Time (in seconds) to wait for reassignment/resynchronization.

Legal values: 0 – 300 seconds

Default value: 300

16. tf_timer: (TFTIMER) Required

The period of time (in seconds) to place the reference in the frozen state.

Legal values: 0 – 300 seconds

Default value: 300

tsapinfo

This form enters Transport Service Access Point (TSAP) information for the International Telephone and Telegraph Consultative Committee (CCITT) protocols.

Form Layout

```

                                     tsapinfo (1/2)
Transport Service Access Point Information Record (Recent Change and Verify)

1.l_tsap:_____                2.r_tsap:_____

3.tp_size:_____                4.r_nsap:_____

5.rmtdte                                r_dteaddr
   1)_____                    5)_____
   2)_____                    6)_____
   3)_____                    7)_____
   4)_____                    8)_____

10.l_nsap:_____

11.extended_tsapid:_____
```

```

                                     tsapinfo (2/2)

12.xl_tsapid:_____

13.xr_tsapid:_____

14.preferred_class:_____        15.alt_0_class:_____

16.alt_1_class:_____            17.alt_2_class:_____

18.alt_3_class:_____            19.alt_4_class:_____
```

Definition of Fields

1. l_tsap: (LTSAP) Required

Local TSAP. This is a key field.

Legal value: 1 – 4 decimal characters

2. r_tsap: (RTSAP) Required

Full remote TSAP address. This is a key field.

Legal values: 1 – 44 decimal characters

3. tp_size: (TPSIZE) Required

Transport Protocol Data Unit (TPDU) maximum size for transport connections.

Legal values: 128, 256, 512, 1024, 2048, 4096, 8192

Default value: 128

4. r_nsap: (RNSAP) Required

This field contains the address of the remote Network Service Access Point (NSAP).

Legal values: 1 – 40 decimal characters

5. rmtdte.r_dteaddr: [RMTDTE.RDTEADDR(ROW)] Required

This field contains the array of remote Data Terminal Equipment (DTE) addresses. The maximum number of remote DTEs is eight.

Legal values: 1 – 14 decimal characters

10. l_nsap: (LNSAP) Required

A local NSAP address which a full local TSAP address will be constructed from; that is, a full local TSAP address consists of the local TSAP selector and the local NSAP address.

Legal values: 1 – 40 decimal characters

11. extended_tsapid: (EXTENDEDTSAPID)

Indicates whether the extended TSAP addressing is used.

Legal values: y, n

Default value: n

12. xl_tsapid: (XLTSAPID)

The extended local TSAP address. This value is used in the connect request TPDU or in the connect confirm TPDU.

Legal values: 32 printable characters allowed, null

Default value: null

13. xr_tsapid: (XRTSAPID)

The extended remote TSAP address. This value is used in the connect request TPDU or in the connect confirm TPDU.

Legal values: 32 printable characters allowed, null

Default value: null

14. preferred_class: (PREFERREDCLASS)

The preferred CCITT Transport Layer Protocol (TLP) class.

Legal values: 0, 1, 2, 3

Default value: 3

15. alt_0_class: (ALT0CLASS)

Indicates whether class 0 is an allowed alternate.

Legal values: y, n

Default value: n

16. alt_1_class: (ALT1CLASS)

Indicates whether class 1 is an allowed alternate.

Legal values: y, n

Default value: n

17. alt_2_class: (ALT2CLASS)

Indicates whether class 2 is an allowed alternate.

Legal values: y, n

Default value: n

18. alt_3_class: (ALT3CLASS)

Indicates whether class 3 is an allowed alternate.

Legal values: y, n

Default value: n

19. alt_4_class: (ALT4CLASS)

Indicates whether class 4 is an allowed alternate.

Legal value: n

Default value: n

Form Checks

```
count {select r dteaddr from rmtdte}>0
  error: "a minimum of one r_dteaddr must be specified";

{select r_dteaddr from rmtdte} must_be_unique
  error: "r_dteaddr values must be unique";

if preferred_class = "0"
  then do;
    alt_1_class must_be "n"
    error: alt_1_class must be n for preferred_class of "0"
    alt_2_class must_be "n"
    error: alt_2_class must be n for preferred_class of "0"
    alt_3_class must_be "n"
    error: alt_3_class must be n for preferred_class of "0"
    alt_4_class must_be "n"
    error: alt_4_class must be n for preferred_class of "0"
  end;

if preferred_class = "1"
  then do;
    alt_2_class must_be "n"
    error: alt_2_class must be n for preferred_class of "1"
    alt_3_class must_be "n"
    error: alt_3_class must be n for preferred_class of "1"
    alt_4_class must_be "n"
    error: alt_4_class must be n for preferred_class of "1"
  end;

if preferred_class = "2"
  then do;
    alt_3_class must_be "n"
    error: alt_3_class must be n for preferred_class of "2"
    alt_4_class must_be "n"
    error: alt_4_class must be n for preferred_class of "2"
  end;

if preferred_class = "3"
  then do;
    alt_4_class must_be "n"
    error: alt_4_class must be n for preferred_class of "3"
  end;

end;
```

ttop

This form gives Teletypewriter (TTY) control information.

Form Layout

```

                                     ttop      (1/3)
Terminal Control Structure For /dev/ttyx  (Recent Change and Verify)
1.option_name: _____
delay  2.bsdelay:_  3.vtdelay:_  4.crdelay: __  5.noal:_
        6.tbdelay:_  7.nldelay: __
parity  8.evenp:_  9.oddp:_
mode10.raw:_      11.crmod:_
12.echo:_ 13.lcase:_ 14.xtabs:_      15.hupcl:_
16.tysdspeed: _____ 17.duplex: _____ 18.stop_bit:_
```

```

                                     ttop      (2/3)
19.character_set: _____ 20.transmission: _____
21.kill: __
      killr1  killr2  killr3  killr4  killr5
22.grp_kill 23. __  24. __  25. __  26. __  27. __
28.eras: __
      erasr1  erasr2  erasr3  erasr4  erasr5
29.grp_eras 30. __  31. __  32. __  33. __  34. __
      delim1  delim2  delim3  delim4  delim5  delim6
35.grp_delim 36. __  37. __  38. __  39. __  40. __  41. __
42.ict: __  43.ackt: _____
```

```

                                     ttop      (3/3)
                                     inval1  inval2
44.grp_inval      45. __  46. __
47.escape: __    48.esc_char: __
                                     gecho1   gecho2   gecho3   gecho4   gecho5   gecho6
49.grp_echo      50. ____ 51. ____ 52. ____ 53. ____ 54. ____ 55. ____
56.bkspc: ____  57.abandonmsg: ____ 58.tt_size: ____ 59.nl: ____
60.wdth: __    61.color: ____
```

Definition of Fields

1. option_name: (OPTIONNAME) Required

Option block name. An eight-character name uniquely identifying this ttop option block. This is a key field.

Legal values: alphanumeric characters

2. bsdelay: (BSDELAY)

Backspace delays.

Legal values: y, n, null

3. vtDELAY: (VTDELAY)

Form feed and vertical tab delays.

Legal values: y, n, null

4. crdelay: (CRDELAY)

Carriage return delay.

Legal values: 0 – 3, 0x0 – 0x3, null

5. noal: (NOAL)

No new line after reading.

Legal values: y, n, null

6. tbdelay: (TBDELAY)

Tab delays.

Legal values: y, n, null

7. nldelay: (NLDELAY)

New line delays.

Legal values: 0 – 3, 0x0 – 0x3, null

8. evenp: (EVENP)

Even parity allowed on input.

Legal values: y, n, null

9. oddp: (ODDP)

Odd parity allowed on input.

Legal values: y, n, null

10. raw: (RAW)

Raw mode.

Legal values: y, n, null

11. crmod: (CRMOD)

Map CR into LF, echo LF or CR as CR-LF.

Legal values: y, n, null

12. echo: (ECHO)

Echo (full duplex).

Legal values: y, n, null

13. lcase: (LCASE)

Map uppercase into lowercase on input.

Legal values: y, n, null

14. xtabs: (XTABS)

Expand on echo and print tabs as space.

Legal values: y, n, null

15. hupcl: (HUPCL)

Hang up on last close.

Legal values: y, n, null

16. tysdspeed: (TYSDSPEED)

Transmission speed in baud.

Legal values: 110, 300, 1200, 1800, 2400, 4800, 9600, 19200, 56000, 64000, null

17. duplex: (DUPLEX)

Full or half-duplex.

Legal values: full, half, null

18. stop_bit: (STOPBIT)

Indicates the number of stop bits.

Legal values: 1, 2, null

19. character_set: (CHARACTERSET)

Indicates the American Standard Code for Information Interchange (ASCII) or Extended Binary Coded Decimal Interchange Code (EBCDIC) character set.

Legal values: ASCII, EBCDIC, null

20. transmission: (TRANSMISSION)

Indicates synchronous or asynchronous transmission.

Legal values: sync, sync, null

21. kill: (KILL)

Indicates the line kill character.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

Default value: @

22. grp_kill (GRPKILL)

23.killr1: (*GRPKILL.KILLR1) — 27.killr5: (*GRPKILL.KILLR5)

Response for kill character. Any five characters in the character set. Default is no response.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

28. eras: (ERAS)

Erase character. Possible values: Any characters in the character set.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

Default value: #

29. grp_eras (GRPERAS)

30. erasr1: (*GRPERAS.ERASR1) — 34.grp_eras.erasr5: (*GRPERAS.ERASR5)

Response for erase character. Any five characters in the character set.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

35. grp_delim (GRPDELIM)

36. delim1: (*GRPDELIM.DELIM1) — 41.grp_delim.delim6: (*GRPDELIM.DELIM6)

End of line delimiters.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

42. ict: (ICT) Required

Intercharacter timer.

Legal values: 0 – 255, HEX

43. ackt: (ACKT) Required

Acknowledgment timer.

Legal values: 0 – 255, HEX

44. grp_inval: (GRPINVAL)

45. grp_inval.inval1: (*GRPINVAL.INVALID1) —

46. grp_inval.inval2: (*GRPINVAL.INVALID2)

Invalid character sequences.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

47. escape: (ESCAPE)

Pathname escape character.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

48. esc_char: (ESCCHAR)

Single escape character. (set to “\”)

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

Default value: xff

49. grp_echo (GRPECHO)

50. gecho1: (*GRPECHO.GECHO1) —

55. gecho6: (*GRPECHO.GECHO6)

Group echo.

Legal values: printable characters, null

56. bkspace: (BKSPACE)

Backspace character.

Possible values: Any character in the character set.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

57. abandonmsg: (ABANDONMSG)

Abandon message character.

Legal values: 0 – 255, ESCAPE, HEX2D, QCHAR, null

58. tt_size: (TTSIZE) Required

Input job field size.

Legal values: 0 – 255, HEX

59. nl: (NL) Required

Number of lines for Cathode Ray Tube (CRT).

Legal values: 0 – 255, HEX

60. wdh: (WDTH) Required

Width of CRT.

Legal values: 0 – 255, HEX

61. color: (COLOR) Required

Color indicator. 0 – black and white; 1 – datamedia

Legal values: 0 – 255

Default value: 0

ucb

This form represents a hardware entity in the physical system. Unit Control Blocks (UCBs) are linked together in a hierarchical manner corresponding to the physical system. UCB information is device independent except for the equipage field. Device dependent information is specified in an option block record and linked to the *ucb* form. UCBs need not have associated option blocks.

UCBs are named with a two-level naming convention. The first level, complex name, is optional. A UCB complex (or community) is a group of logically related and physically connected UCBs. Complex names are used when the unit name (the second level name) is not unique. This occurs only in Control Unit (CU) communities. For example, CU 0 and CU 1 both have units named Central Control (CC) 0. The two-level names would be CU 0, CC 0 and CU 1, CC 0.

Form Layout

```
ucb (1/4)
Information About Each Physical Device (Recent Change and Verify)
1.k_complex_name: _____ 2.k_complex_number: ____
3.k_unit_name: _____ 4.k_unit_number: ____
ucb_type 5.essential:_ 6.top:_ 7.unique_unit:_
8.pseudo_node:_ 9.restorable:_ 10.removable:_
11.device_port: _____ 12.device_type: _____
13.internal_device: ____ 14.plant:_
15.critical:_16.portswitch:_ 17.posswitch:_
```

ucb (2/4)

minor_device_status 18.boot:_ 19.rexinh:_ 20.errlog:_
21.major_status: ____ 22.equipage: _____
23.device_num: ____ 24.channel_num: ____
25.hv: _____ 26.mt: ____ 27.mv: ____
28.interrupt_num: _ 29.service_intrpt: _
30.pathname: _____
31.location
32.floor: _ 33.aisle:_ 34.plate: __ 35.pack: _

ucb (3/4)

	complex_name	complex_number	unit_name	unit_number
36.down	37) _____	38) ____	39) _____	40) ____
41.side	42) _____	43) ____	44) _____	45) ____
46.cntrl_0	47) _____	48) ____	49) _____	50) ____
51.cntrl_1	52) _____	53) ____	54) _____	55) ____
56.error_table: _____	57.option_name: _____			
58.pcid: ____	59.pcsdid:_	60.h_type: __		
61.handler_id: ____				

ucb (4/4)

62.packcode: ____ 63.issue: _ 64.version: _

65.cvaddr: ____ 66.u_model: ____

Definition of Fields

1. k_complex_name: (KCOMPLEXNAME) Optional

This is a key field. When a two-level naming convention is used, it is the name of the complex in which this UCB resides. Otherwise, this field is null.

2. k_complex_number: (KCOMPLEXNUMBER) Optional

This a key field. When a two-level naming convention is used, it is the number of the complex in which this UCB resides. Otherwise, it is null.

3. k_unit_name: (KUNITNAME) Required

This is a key field. It is the name of the unit represented by this UCB.

4. k_unit_number: (KUNITNUMBER) Required

This is a key field. It is the member number of the unit represented by this UCB.

5. essential: (ESSENTIAL)

Describes when a device is essential for system operation. An essential device can be removed from service only when its function is provided by a suitable replacement.

Legal values: AFTERBOOT – only after the boot interval
 ALWAYS – always essential
 DURINGBOOT – only during the boot interval
 N – never essential
 NEVER – never essential
 NOMANRMV – only during manual remove attempts
 Y – always essential
 null

Default value: NEVER

6. top: (TOP)

A UCB marked top is the highest level UCB in a community.

Legal values: y, n, null

Default value: n

7. unique_unit: (UNIQUEUNIT)

A UCB marked unique is one whose corresponding physical device has a unique name.

Legal values: y, n, null

Default value: n

8. pseudo_node: (PSEUDONODE)

Pseudo nodes are used to represent one or more physical entities. A UCB that is a pseudo node does not have a corresponding hardware entity.

Legal values: y, n, null

Default value: n

9. restorable: (RESTORABLE)

A restore command may be issued only for units marked restorable. If a UCB is not restorable, it may be restored only by restoring its controller. This flag is used by the Diagnostic Monitor (DIAMON).

Legal values: y, n, null

Default value: n

10. removable: (REMOVABLE)

A UCB that is removable can be removed (together with devices under it) without endangering the system. This flag is used by CONFIG to quickly determine if there are essential units below.

Legal values: y, n, null

Default value: n

11.device_port: (DEVICEPORT) Required

This specifies the port to which a driver for this device is connected. Ports are used for driver/device communication.

Legal values:	ADP	(automatic diagnostic process)
	AIM	(application integrity monitor)
	APACP	(craft interface)
	APDAP	[Display Administration Process (DAP) applications port]
	API	(attach processor)
	APRTS	(craft interface)
	BWM	(field update port)
	DAP	(display administration process)
	DCI	(DSCH computer interconnect driver)
	DFC	[Disk File Controller (DFC) driver]
	DISK	(disk driver) (<i>UNIX</i> ® Real-Time Reliable (RTR) Operating System G2 – Release 6.8)
	DSCH	(TUS dual serial channel)
	EMUDL	(emulation data link driver for AWS)
	ERRLOG	(error logger)
	FIP	(foundation interface processor)
	FLDUPD	(field update)
	GOS	(generic update output spooler)
	IOP	[Input/Output (I/O) processor]
	KLLA	[kernel Low-Level Access (LLA)]
	MEA	
	MIRA	(maintenance input request administrator)
	MSGS	
	NONE	
	OVLD	
	PCPEIH	(PCP error interrupt handler)
	PSM	(power switch monitor)
	RETRO	
	RMT	(SCAN-2 interface)
	RNG	
	RTS	(real-time status reports)
	SCHED	(scheduler process)
	SCSDA	[Scanner and Signal Distributor (SCSD) administrator]
	SNIC	(DSCH computer interconnect driver) (<i>UNIX</i> RTR Operating System Release 21.3 only)
	TOS	[Teletypewriter (TTY) output spooler]
	TUS	(test utility system)
	UDP	(user diagnostic process)
	VTTY	(Virtual TTY)
	XCIA	(critical indicator process)
	null	

12. device_type: (DEVICETYPE)

This specifies the type of hardware device represented by this form.

Legal values:	ACHI	(attach processor)
	API	(attached processor)
	CACHE	
	CC	(central control unit)
	CU	(control unit)
	DAT	(SCSI digital audio tape)
	DCI	(DSCH computer interconnect) (UNIX RTR Operating System Release 21.7 and later)
	DFC	(disk file controller)
	DMA	(direct memory access)
	DSCH	(dual serial channel)
	DSMLNK	(Physical TTY link to an adjunct processor) (UNIX RTR Operating System Release 21.13 and Later)
	DUI	(TN82)
	DUIC	(for TN82)
	EAI	(emergency action interface)
	FIP	(foundation interface processor)
	FP	(foundation processor)
	FPC	(foundation processor controller)
	HSD	(high-speed data link)
	HSDC	(high-speed data link controller)
	IOP	(input/output processor)
	MAS	(mass storage unit)
	MHD	(moving head disk)
	MMP	
	MSGS	
	MT	(magnetic tape)
	MTC	(magnetic tape controller)
	MTTY	(maintenance TTY)
	MTTYC	(maintenance TTY controller)
	NSC	(NSC hyperchannel)
	NSCI	
	PL	(programmable link)
	PLC	(programmable link controller)
	PSI	(peripheral system interface)
	ROP	(read-only printer)
	RPC	
	RUN	
	SAT	(store address translation)
	SBUS	[Small Computer System Interface (SCSI) bus]
	SCDL	(synchronous data link)

SCH	(serial channel)
SCSD	(scan and distributor)
SDL	(synchronous data link)
SDLC	(synchronous data link controller)
SMT	(SCSI 9-track magnetic tape)
SNIC	(DSCH computer interconnect (UNIX RTR Operating System Release 21.3 only)
SOFT	(software device)
TTY	(TTY)
TTYC	(TTY controller)
UC	(utility circuit)
null	(not a physical device)

13. internal_device: (INTERNALDEVICE)

This field is used to distinguish among hardware units of the same type.

Legal values: 0 – 255, HEX, null

14. plant: (PLANT)

Specifies if the history of unit (failure rates and so forth) is being kept by plant measurement.

Legal values: y, n, null

15. critical: (CRITICAL)

Critical device. Used to identify test equipment not needed for functionality of the unit (Diags).

Legal values: y, n

Default value: y

16. portswitch: (PORTSWITCH)

Port switchable.

Legal values: y, n

Default value: n

17. posswitch: (POSSWITCH)

Port switch position.

Legal values: 0, 1, null

Default value: 0

18. boot: (BOOT)

Indicates whether a device is a boot device. The access to unit is to be inhibited when system is booting (unit is considered out of service).

Legal values: y, n, null

19. rexinh: (REXINH)

This field tells the fault recovery system's *rex* process whether automatic regular maintenance processes (known as Routine EXercises or REX) have been inhibited (that is, REX will not be run on this device) or if they are allowed (that is, REX will be run on this device). REX processes include such things as diagnostics.

Only "top" UCBs are consulted for this flag. A top UCB contains "y" in field 6 (top).

Legal values: y (REX is inhibited), n or null (REX is allowed)

⇒ NOTE:

rexinh must contain 'y' if device_type (field 12) is null or contains any of the following values: DFC, DCI, SNIC, MSGS, or SOFT.

Additionally, the physical device must be Out of Service (OOS) before the value contained in the rexinh field can be changed.

For Releases 6.7, 21.1 and later: rexinh cannot be changed if t_rexinh (field 23 in the *ucbdyn* form for this device) contains 'y.'

20. errlog: (ERRLOG)

Indicates to keep a detailed error log for this unit.

Legal values: y, n, null

21. major_status: (MAJORSTATUS)

This field gives the major device status.

Legal values:	ACT	Active. Unit is currently performing on-line functions. See the following Note.
	GROW	UCB is being grown.
	INIT	The device is currently being initialized. This state is set internally and is a transient state.
	OFL	Off-line. Unit has been made unavailable by a program for normal operational use.
	OOS	Out of service. Unit does not meet criteria for any of the above states. It may be faulty or manually out of service. Additional information may be found in the minor status. See the following Note.

STBY	Standby. No known faults, the unit is currently available to perform on-line functions, but is not currently doing so. See the following Note.
UNAV	Unavailable. Unit has been manually made unavailable under any circumstances for normal operational use.
UNEQIP	Unequipped. UCB is in the Equipment Configuration Database (ECD), but the corresponding hardware is not logically available. It may or may not be physically present.

null

Default value: OOS

⇒ NOTE:

For the data communication peripheral units, SDLs (that is, TN75C), HSDs (that is, TN82), and SCCs (that is, TN983), the following state correction apply:

ACT	Information transfer state (S5): State S5, at the data link layer (level 2), is entered after link initialization has been completed and if no errors or special requests are currently affecting the protocol. In this state, both stations may exchange level 2 data freely.
OOS	Logically disconnected state (S1): State S1, at the data link layer (level 2), is entered as a result of having been instructed to logically discontinue operation by either the remote station or a higher entity.
STBY	Link setup state (S2): A station enters state S2, at the data link layer (level 2), after transmitting a request to (re)initialize the link. The restoral process has completed in this state. This state does not guarantee the presence of the data set interface signals (level 1).

22. equipage: (EQUIPAGE)

This is a bit field reflecting the equipage of circuits but can also be used to identify the type of circuit present. The layout of this field is different for each type of hardware entity and its meaning varies from device to device. It is usually used as a group of 1 bit flags or subdivided into groups of bits to form hex digits. The diagnostics use the information in this field to make test/no test decisions based on the presence or absence of circuits.

For a 3B20D computer MASC UCB, each 1 bit represents the presence of a memory array. A memory array may contain 1/2, 1, or 2 megabytes (MB) of memory. There may be up to 16 memory arrays. The first array is represented by bit 0, the second by bit 1, and so forth.

For a 3B21D computer MASC UCB, this field contains the total amount of memory (in MB) present. This value must be between 0x20 (32 MB) and 0x100 (256 MB) and must be divisible by 8.

For a CH UCB, this field indicates that the corresponding port is equipped. Bits 0 – 15 are used for a dual serial channel and bits 0 – 19 for a serial channel.

For an Input/Output Processor (IOP) or Storage Module Drive-Disk File Controller (SMD-DFC) UCB, the equipage number indicates the size of the Peripheral Interface Controller (PIC) memory (in units of 4096 bytes).

For a SCSI-DFC, this field is unused.

For a SCSI-MT (tape), this field gives the type of drive.

For an MHD UCB, the equipage number gives the type and size of drive.

Legal values: 0 – MAXINT, HEX, null

23. device_num: (DEVICENUM)

This designates the address of the device on the channel.

Legal values: 0 – 15, NULL, null

24. channel_num: (CHANNELNUM)

This designates a particular channel with which the device is associated.

Legal values: 0 – 19, NULL, null

25. hv: (HV)

Hardware version (optional and Class B LDI status). hv is a bit field reflecting the installation of non-Class A Change Notices (CNs) that have coordinated software changes or flags to diagnostics and Trouble Location Process (TLP). This field can be used as a whole or subdivided into smaller fields or flag bits as required. The layout of this UCB field is different for each type of hardware entity. This field is also used as a port switch indicator.

Legal values: 0 – MAXINT, HEX, null

26. mt: (MT)

Modification level (Class A LDI level). mt indicates the latest Class A CN with coordinated program and/or database changes installed in a unit (top UCB) and subunit. Class A CNs that require coordinated program and/or database changes are assigned a sequential mt value. The mt value is usually updated in both the unit (top UCB) mt and a subunit mt.

To be forward compatible, CNs that change mt field(s) must be installed in sequence; for example, an mt of 6 implies that Class A CNs associated with mt values 1 through 6 are installed. Installing Class A CNs with coordinated program and/or database changes out of mt sequence or failing to properly update the mt field(s) when a CN with coordinated program and/or database changes is installed may have undesirable results. This controls the execution of certain diagnostic tests that are sensitive to the LDI hardware level.

Legal values: 0 – 65535, HEX, null

27. mv: (MV)

A numeric field reflecting the version of the unit or subunit. It can be used as a whole or subdivided into smaller number fields or flag bits as required. It is the UCB field used most often by diagnostics and Trouble Location Process (TLP) to execute alternate phases or tests and identify alternate circuit board codes. The layout of this UCB field is different for each type of hardware entity.

Legal values: 0 – 255, HEX, null

28. interrupt_num: (INTERRUPTNUM)

This is the interrupt register level for a channel UCB. For a peripheral device, this is the interrupt level on a channel for units that can interrupt the 3B20D/3B21D computers. This indicates which device port of the channel the device is connected. Otherwise, it is unused.

Legal values: 0 – 31, null

29. service_intrpt: (SERVICEINTRPT)

The service request interrupt number. This is for a non-Direct Memory Access (DMA) service request. This number is the bit in the interrupt register to which this service request is wired. Consult each individual device type to determine the correct meaning.

Legal values: 0 – 31, null

30. pathname: (PATHNAME)

This is a partial pathname used by the deferrable maintenance package to locate a utility directory or file associated with the unit. This directory contains the diagnostics and restoral/removal procedures and other utilities for this unit.

Legal values: printable characters

31. location (LOCATION)

This gives the physical location of the unit in the office.

32. location.floor: (*LOCATION.FLOOR)

Unit floor location.

Legal values: 0 – 99, null

33. location.aisle: (*LOCATION.AISLE)

Unit aisle location.

Legal values: alphabetic characters, null

34. location.plate: (*LOCATION.PLATE)

Inches from the floor.

Legal values: 0 – 68, null

35. location.pack: (*LOCATION.PACK)

Inches from the left side.

Legal values: 0 – 188, –, null

36. down: (DOWN)

Indicates by name the first controlled device under a particular device. This is the link to the first lower level UCB (a unit controlled by this unit).

37) down.complex_name: (*DOWN.COMPLEXNAME)

First part of a four-part *ucb* identifier.

38) down.complex_number: (*DOWN.COMPLEXNUMBER)

Second part of a four-part *ucb* identifier.

39) down.unit_name: (*DOWN.UNITNAME)

Third part of a four-part *ucb* identifier.

40) down.unit_number: (*DOWN.UNITNUMBER)

Fourth part of a four-part *ucb* identifier.

41. side (SIDE)

Link to the next UCB at the same level (a unit with the same controller as this unit).

42) See Items 37 – 40.

46. cntrl_0 (CNTRL0)

Link to controller UCB. Indicates the controlling device by name.

47) See Items 37 – 40.

51. cntrl_1: (CNTRL1)

Link to controller 1 UCB. Indicates the second controlling device by name. This is used only by peripheral interface control devices.

52) See Items 37 – 40.

56. error_table: (ERRORTABLE)

Error threshold record name. Indicates by symbolic name the error table associated with a device.

Legal values: alphanumeric characters, null

57. option_name: (OPTIONNAME)

Option block name. Indicates by symbolic name the option block associated with a particular device.

Legal values: alphanumeric characters, null

58. pcid: (PCID)

Peripheral controller ID.

Legal values: 0 – 15, 0x0 – 0xF, null

59. pcsdid: (PCSDID)

Peripheral controller subdevice ID. This is the backplane subdevice connector position.

Legal value: 0 – 3, null

60. h_type: (HTYPE)

Handler type.

The input represents a 4-bit field in which each bit specifies a particular requirement for the I/O driver or the device handler.

When set	Means
bit 0	Application handler administers this unit.
bit 1	Unit uses DMA for normal operation.
bit 2	Automatic delayed restore required.
bit 3	Peripheral controller and subdevice unit required in DIOP.

Legal values: 0 – 15, 0x0 – 0xF, null

61.handler_id: (HANDLERID)

Device handler IDs.

Legal values:	aph	(application protocol handler)
	ci	(craft interface)
	ci83	
	dui	(direct user interface)
	eai	(emergency action interface)
	fpc	(foundation pc for 5ESS switch)
	fsd	
	iop	
	lc	
	mmp	(module message processor for 5ESS switch)
	mtce	
	mtp	(message transfer protocol handler)
	np	
	null	
	plc	(CCIS pc)
	rng	
	rpc	
	run	
	scsd	(scanner and signal distributor)
	sdl	(synchronous data link)
	sl	
	tape	(tape)
	test	
	tp	(OSI – transport protocol handler)
	tty	(terminal)

62. packcode: (PACKCODE)

Pack name.

Legal values: alphanumeric characters, null

63. issue: (ISSUE)

PC code issue.

Legal values: numeric digits, null

64. version: (VERSION)

Microcode version.

Legal values: numeric digits, null

65. cvaddr: (CVADDR)

Microcode version address.

Legal values: HEX, null

66. u_model: (UMODEL)

This field indicates the model information for a major hardware unit such as CU, DFC, or IOP.

Legal values: 0 – 255 (See the following Note.)

Default value: 0

⇒ NOTE:

Although the legal range for this field includes “0,” this value has no meaning to the drivers which access the field.

The device types with defined mapped numerical equivalents to hardware types are listed in Table 6.3. The table should be used to identify the correct “u_model” value when the device_type is either CU, DFC, or IOP. (For all other device_types, this field can remain NULL.)

Table 6-3. u_model Values Top Units ucb Forms

Hardware Type	"u_model" Value Entered by the User
CU: 3B20D Computer Model 1 3B20D Computer Model 2/3 Uses SMD DFC as its Primary DFC 3B20D Computer SCSI New Ship SCSI DFC Built in as Primary DFC 3B21D Computer	1 3 4 5
DFC: SMD-DFC Uses TN62 & TN61 Boards SMD-DFC Uses TN68 & TN61 Boards SMD-DFC Uses TN18 & TN20 Boards SMD-DFC Uses TN68 & TN54 Boards SMD-DFC Uses TN19 & TN20 Boards SMD-DFC Uses TN19 & UN55 Boards SCSI-DFC Uses TN2116 & UN294 Boards SCSI-DFC Uses TN2116 & UN373 Boards SCSI-DFC Uses UN580 Board	1 2 3 4 5 6 7 8 9
IOP: 3B20D Computer Model 1 3B20D Computer Model 2/3 Uses SMD DFC as its Primary DFC 3B20D Computer SCSI New Ship SCSI DFC Built in as Primary DFC 3B21D Computer Uses KBN10 Board Emulation Datalink Driver for AWS	1 3 4 5 6

Form Checks

```
if device_type is unknown or device_type is_in
{"DFC", "DCI", "MSGs", "SOFT"}
  then rexin must_be known and rexin must_be "y"
  error: "when device_type is null, DFC, DCI, MSGs, or SOFT rexin
  must be 'y'";

if device_type is "SOFT" and device_port != "DCI"
  then major_status must_be "OOS"
  error: "major_status must be 'OOS' when device_type is 'SOFT'";

if major_status is "GROW"
  then handler_id must_not_be_in {"tty", "mtce", "iop", "test"
  "fsd", "eai", "mmp", "rng", "ic", "sl", "mtp", "tp", "aph"}
  error: "when major_status is "GROW", this handler_id is not allowed;

  case (essential):
    when{"Y", "ALWAYS", "AFTERBOOT", "DURINGBOOT", "NOMANRMV"}: do;
      removable must_be "n"
      error: "removable must be n";
    end;
  end;

(k_complex_name != down.complex_name or
k_complex_number != down.complex_number or
k_unit_name != down.unit_name or
k_unit_number != down.unit_number)
  error: "down ucb must not be the same as parent ucb";

(k_complex_name != side.complex_name or
k_complex_number != side.complex_number or
k_unit_name != side.unit_name or
k_unit_number != side.unit_number)
  error: "side ucb must not be the same as parent ucb";

(k_complex_name != cntrl_0.complex_name or
k_complex_number != cntrl_0.complex_number or
k_unit_name != cntrl_0.unit_name or
k_unit_number != cntrl_0.unit_number)
  error: "cntrl_0 ucb must not be the same as parent ucb";

(k_complex_name != cntrl_1.complex_name or
k_complex_number != cntrl_1.complex_number or
k_unit_name != cntrl_1.unit_name or
k_unit-number != cntrl_1.unit_number)
  error: "cntrl_1 ucb must not be the same as parent ucb";
case (device_type):
  when {"CU"}: do;
    case (essential);
      when{"N", "NEVER", unknown, "DURINGBOOT", "AFTERBOOT",
"NOMANRMV"}: do;
        essential must_be "Y"
        error: "essential must be Y or ALWAYS";
      end;
    end;
  end;
```

```
end;
top must_be "y"
  error: "top must by y";
unique_unit must_be "n"
  error: "unique_unit must be n";
pseudo_node must_be "y"
  error: "pseudo_node must by y";
restorable must_be "y"
  error: "restorable must be y";
u_model must be known and u_model in {1,3,4,5}
  error: "when device type is CU, u_model must be 1, 3,
4, or 5";
end;

when {"CC", "MAS", "SAT", "SCH", "DSCH", "ACHI", "DMA"}: do;
  case (essential);
    when {"N", "NEVER", unknown, "DURINGBOOT", "AFTERBOOT",
"NOMANRMV"}: do;
      essential must_be "Y"
        error: "essential must be Y or ALWAYS";
    end;
  end;
top must_be "n"
  error: "top must be n";
unique_unit must_be "n"
  error: "unique_unit must be n";
pseudo_node must_be "n"
  error: "pseudo_node must be n";
restorable must_be "n"
  error: "restorable must be n";
end;

when {"CACHE", "UC"}: do;
top must_be "n"
  error: "top must be n";
unique_unit must_be "n"
  error: "unique_unit must be n";
pseudo_node must_be "n"
  error: "pseudo_node must be n";
restorable must_be "n"
  error: "restorable must be n";
end;

when {"DFC"}: do;
  case (essential);
    when {"Y", "ALWAYS", "DURINGBOOT", "AFTERBOOT",
"NOMANRMV"}:do;
      essential must_be "N"
        error: "essential must be N,NEVER,
or null";
    end;
  end;
end;

if u_model is_in {7 thru 9}
```

```
        then down.unit_name must_be_in {unknown,"SBUS"}
            error: "if DFC is SCSI device type,
down pointer must be either null or SBUS";
    if u_model is_in {1 thru 6}
        then down.unit_name must_be_in {unknown,"MHD"}
            error:"if DFC is SMD device type,
down pointer must be either null or MHD";
    top must_be "y"
        error: "top must be y";
    unique_unit must_be "y"
        error: "unique_unit must be y";
    pseudo_node must_be "n"
        error: "pseudo_node must be n";
    restorable must_be "y"
        error: "restorable must be y";
    device_port must_be "DFC"
        error: "device_port must be DFC";
    u_model must be known and u_model must be in {1 thru 9}
        error: "when device type is DFC, u_model must
contain the values 1 thru 9";
end;

when {"DCI"}: do;
    case (essential);
        when {"Y", "ALWAYS", "DURINGBOOT", "AFTERBOOT",
"NOMANRMV"}: do;
            essential must_be "N"
                error: "essential must be N, NEVER,
or null";
        end;
    end;
    top must_be "y"
        error: "top must be y";
    unique_unit must_be "y"
        error: "unique_unit must be y";
    pseudo_node must_be "n"
        error: "pseudo_node must be n";
    restorable must_be known and restorable must_be "y"
        error: "restorable must be y";
    device_port must_be "DCI"
        error: "device_port must be DCI";
    option_name must_be known
        error: "option_name cannot be blank when device_type is
DCI";
end;

when {"SOFT"}: do;
    case (device_port);
        when {"DCI"}: do;
            case (essential);
                when {"Y", "ALWAYS", "DURINGBOOT", "AFTERBOOT",
"NOMANRMV"}: do;
                    essential must_be "N"

```

```
                error: "essential must be N, NEVER, or
                null";
            end;
        end;
    top must_be "n"
        error: "top must be n";
    unique_unit must_be known and unique_unit must_be "y"
        error: "unique_unit must be y";
    pseudo_node must_be known and pseudo_node must_be "y"
        error: "pseudo_node must be y";
    restorable must_be known and restorable must_be "n"
        error: "restorable must be n";
    (side.complex_name must_be unknown and
    side.complex_number must_be unknown and
    side.unit_name must_be unknown and
    side.unit_number must_be unknown)
        error: "side ucb must be null";
    end;
end;
end;
when {"SBUS"}: do;
    internal_device must_be_in {unknown, "0", "1"};
    error: "when device type is SBUS,
    internal_device must be null, 0, or 1";
end;
when {"MHD"}: do;
    top must_be "n"
        error: "top must be n";
    unique_unit must_be "y"
        error: "unique_unit must be y";
    pseudo_node must_be "n"
        error: "pseudo_node must be n";
    restorable must_be "y"
        error: "restorable must be y";
    device_port must_be "DFC"
        error: "device_port must be DFC";
end;
when {"IOP"}: do;
    case (essential);
        when {"Y", "ALWAYS", "DURINGBOOT", "AFTERBOOT",
        "NOMANRMV"}: do;
            essential must_be "N"
                error: "essential must be N,
                NEVER, or null";
            end;
        end;
    top must_be "y"
        error: "top must be y";
    unique_unit must_be "y"
        error: "unique_unit must be y";
```

```
pseudo_node must_be "n"
    error: "pseudo_node must be n";
restorable must_be "y"
    error: "restorable must be y";
device_port must_be_in {"IOP", "EMUDDL", "VTTY"}
    error: "device_port must be IOP, EMUDDL or VTTY";
if mv is known and mv !=0
    then packcode must_be known
        error: "when device type is IOP and
            mv is known, packcode must be known";
u_model must be known and u_model in {1,3,4,5,6}
    error: "when device type is IOP, u_model must be
        1, 3, 4, 5, or 6";
if u_model is 6 and k_unit_name is "IOP" and k_unit_number
    is 2 then device_port must be "EMUDDL"
    error: "device_port must be EMUDDL when u_model is 6
        and unit is IOP 2";
end;

when {"TTYC", "MTTYC", "MTC", "SDLC"}: do;
    case (essential);
        when {"Y", "ALWAYS", "DURINGBOOT"}: do;
            essential must_be "N"
                error: "essential must be N, NEVER,
                    AFTERBOOT, NOMANRMV or null";
        end;
    end;

if internal_device is_in {"0",unknown}
    then pcid must_be_in {"0",unknown}
        error: "if internal_device field is unknown then
            pcid field must be unknown";

if internal_device is known and internal_device != 0
    then pcid must_be known
        error: "if internal_device is known then pcid must
            be known";

if internal_device is known
    then internal_device must_equal pcid
        error: "for controller devices pcid and
            internal_device must be identical";

end;

when {"DUIC", "FPC", "PLC", "SCSD", "HSDC"}: do;
    if internal_device is_in {"0",unknown}
        then pcid must_be_in {"0",unknown}
            error: "if internal_device field is unknown
                then pcid field must be unknown";

.SK
if internal_device is known and internal_device != 0
    then pcid must_be known
        error: "if internal_device is known then
```

```
        pcid must be known";

    if internal_device is known
        then internal_device must_equal pcid
            error: "for controller devices pcid and
                internal_device must be identical";
    end;

when {"TTY", "MT", "ROP", "EAI", "SDL"}: do;
    case (essential);
        when {"Y", "ALWAYS", "DURINGBOOT"}: do;
            essential must_be "N"
                error: "essential must be N, NEVER,
                    AFTERBOOT, NOMANRMV or null";
        end;
    end;

if internal_device is_in {"0",unknown}
    then pcsdid must_be_in {"0",unknown}
        error: "if internal_device field is unknown
            then pcsdid field must be unknown";

if internal_device is known and internal_device != 0
    then pcsdid must_be known
        error: "if internal_device is known then
            pcsdid must be known";

if internal_device is known
    then internal_device must_equal pcsdid
        error: "for devices under controller pcsdid
            and internal_device must be identical";

    end;
```

```
when {"DUI", "FP", "PL", "HSD", "SCDL", "MTTY", "DSMLNK"}: do;
  if internal_device is_in {"0",unknown}
    then pcsdid must_be_in {"0",unknown}
    error: "if internal_device field is unknown
    then pcsdid field must be unknown";

  if internal_device is known and internal_device != 0
    then pcsdid must_be known
    error: "if internal_device is known then
    pcsdid must be known";
  if internal_device is known
    then internal_device must_equal pcsdid
    error: "for devices under controller pcsdid
    and internal_device must be identical";

  device_port must_be "NONE"
  error: "device_port must be NONE";
end;

when {"NSCI"}: do;
  device_port must_be "PCPEIH"
  error: "device_port must be PCPEIH";
end ;
end;

if device_type is "DMA"
  then channel_num must_be known and channel_num must_be_in
  {"10","15"} error: "channel_num must be 10 or 15 when
  device_type is 'DMA'";

if device_type is_in {"DAT","DCI","DFC","DSCH","DSMLNK","DUI",
"DUIC","EAI","FP","FPC","HSD","HSDC","IOP","MHD","MT",
"MTC","MTTY","MTTYC","OPT","PL","PLC","ROP","SCDL",
"SCSD","SDL","SDLC","SMT","TTY","TTYC"}

and major_status != "UNEQIP"
  then channel_num must_be known and channel_num must_be_in
  {"11","12","13","14","16","17","18","19"}
  error: "channel_num must be 11-14,16-19 for this unit's
  device_type";
```

ucbdyn

This form shows review-only information associated with a ucb.

Form Layout

Ucb Dynamic Data	ucbdyn	(1/2)	(Review Only)
1.complex_name: _____	2.complex-number: ____		
3.unit_name: _____	4.unit_number: ____		
5.u_rmvd: _____	6.usable:_		
7.updated:_	8.inhibited:_		
9.bypass:_	10.man_request:_		
11.util: ____	12.admin: ____	13.force_act:_	

	ucbdyn	(2/2)	
14.rstcnt			
15.rec_restorable: ____	16.max_restorable: ____		
17.beg_time: _____	18.interval: _____		
19.terminal_num: ____	20.srch_ptr: _____		
21.ext_opt_ptr: _____	22.min_dev_hd: _____		
23.t_rexinh:_			

Definition of Fields

1. complex_name: (COMPLEXNAME) Optional

This is a key field. When a two-level naming convention is used, it is the name of the complex in which this UCB resides. Otherwise, this field is null.

2. complex_number: (COMPLEXNUMBER) Optional

This is a key field. When a two-level naming convention is used, it is the number of the complex in which this UCB resides. Otherwise, it is null.

3. unit_name: (UNITNAME) Required

This is a key field. It is the name of the unit represented by this form.

4. unit_number: (UNITNUMBER) Required

This is a key field. It is the member number of the unit represented by this ucb.

5. u_rmvd:

This tells whether a unit has been removed and why.

Legal values: COMRMVD (community removed)
MANRMVD (manually removed)
NOTRMVD (unit not removed)
XFLTRMVD (faulty unit removed)
ERRRMVD (device removed with errors)
REXRMVD (removed by REX)
RSTRMVD (removed for restore)

6. usable:

The ucb marked usable is accessible and has no known faults. This is used by CONFIG to determine which units can be restored to service in an emergency situation.

Legal values: y, n, null

7. updated:

For main store controllers, this tells that the memory is up to date between the active and standby CUs.

Legal values: y, n, null

8. inhibited:

The unit is in an inhibited state. Errors are ignored or cleaned up. The unit will not be removed from service due to excessive errors.

Legal values: y, n, null

9. bypass:

This is used for a cache. It means the unit is not being used.

Legal values: y, n, null

10. man_request:

There is a pending manual request on the unit.

Legal values: y, n, null

11. util:

This gives the diagnostic status of the device. It is used for the coordination of utility control functions.

Legal values: ABT
ATP (all tests pass)
CATP
NTR (no test run)
STF (some tests fail)

12. admin:

This field is used for internal administration to prevent utility operations from interfering with each other.

Legal values: RSV (unit is reserved)
UNRSV (unit is unreserved)

13. force_act:

This indicates if unit is forced to remain active.

Legal values: y, n, null

14. rstcnt

15. rec_restorable:

Count of recent restorals.

Legal values: 0 – 65535, HEX, null

16. max_restorable:

The maximum restorals allowed.

Legal values: 0 – 65535, HEX, null

17. beg_time:

The time when restoral count began.

Legal values: 0 – MAXINT, HEX, null

18. interval:

The interval when less than max restorals allowed.

Legal values: 0 – MAXINT, HEX, null

19. terminal_num:

Id of the terminal whose manual request is pending.

Legal values: printable characters or null

20. scrch_ptr:

For IOP units, this is a main memory pointer to specific device associated read-write scratch areas within an IOP. This field is initialized by the IOP driver during bootstrap.

Legal values: 0 – MAXINT, HEX, null

21. ext_opt_ptr:

External device dependent scratch area.

Legal values: 0 – MAXINT, HEX, null

22. min_dev_hd:

The link to minor device chain head cell.

Legal values: printable characters, null

23. t_rexinh:

Routine exercise (such as diagnostics) are temporarily inhibited. Only "top" UCBs are consulted for this flag.

Legal Values: y, n, null

Default value: n

ularp

This form contains information about a process or run command to be executed by User-Level Automatic Restart Process (ULARP).

Form Layout

```
ularp          (1/1)
ULARP ECD Record (Recent Change and Verify)

1.rec_name: _____ 2.exec_no: ____
3.child_priority: __4.boot_prm: __ 5.sleep: _____
6.utilid: _____ 7.runtype: ____ 8.run_config: ____
9.craft:_10.cruc_craft:_ 11.uniq_child:_ 12.wait_for_run:_
13.proc_path: _____
14.proc_args: _____
15.shell_cmd: _____
```

Definition of Fields

1. rec_name : (RECNAME) Required

ularp record name. This is the key field of the form.

Legal values: printable characters

2. exec_no: (EXECNO) Required

Specifies the order in which run commands and child processes are executed during a bootstrap or craft initialization.

Legal values: 0 – 32767

3. child_priority: (CHILDPRIORITY) Required

Child process priority if different from normal priority.

Legal values: 0 – 192

Default value: 50

4. boot_prm: (BOOTPRM) Required

Boot progress Processor Recovery Message (PRM).

Legal values: 0 – 255 or HEX

Default value: 0

5. sleep: (SLEEP) Required

Time to delay before executing next process or command (in seconds).

Legal values: 0 – 2147483647

Default value: 0

6. utilid: (UTILID) Required

Utility id of the process or the run command to be used as a success PRM.

Legal values: 0 – 2047 or 0x0 – 0x7ff

Default value: 0

7. runtype: (RUNTYPE) Required

Is this a run command or process?

Legal values: PROCESS, COMMAND

8. run_config: (RUNCONFIG)

The configuration in which the process or command is to run.

Legal values: BOTH, FULL, MIN

Default value: BOTH

9. craft: (CRAFT)

Is the process to be killed by a craft initialization?

Legal values: y, n

Default value: n

10. `cruc_craft`: (CRUCCRAFT)

Is the process crucial for the printing of messages instead of PRMs?

Legal values: y, n

Default value: n

11. `uniq_child`: (UNIQCHILD)

Should there be only one system copy of this process?

Legal values: y, n

Default value: n

12. `wait_for_run`: (WAITFORRUN)

Should the run command complete before starting the next process or command?

Legal values: y, n

Default value: n

13. `proc_path`: (PROCPATH)

Process pathname.

Legal values: printable characters, null

14. `proc_args`: (PROCARGS)

Process arguments (up to 9).

Legal values: printable characters, null

15. `shell_cmd`: (SHELLCMD)

This field should contain the full pathname of the run command and the arguments that the command should use.

This field should be null for *ularp* entries specified as processes in field 7 of this form.

Legal values: printable characters, null

Form Checks

```
if runtype is "COMMAND"
  then do:
    proc_path must_be unknown
    error: "proc_path must be null when runtype is COMMAND";
    proc_args must_be unknown
    error: "pro_args must be null when runtype is COMMAND";
  end;
if runtype is "PROCESS"
  then
    shell_cmd must_be unknown
    error: "shell_cmd must be null when runtype is PROCESS";
```

workhrs

This form is used to specify the working hours for an exchange: two blocks of time for each day of the week. During the working hour blocks specified, dynamic terminal authority is allowed.

Form Layout

Work Hours Record	workhrs	(1/1)
		(Recent Change and Verify)
1.workhrs_name:_____		
2.work_hours:	block 1	block 2
	start end	start end
1) Sunday:	_____	_____
2) Monday:	_____	_____
3) Tuesday:	_____	_____
4) Wednesday:	_____	_____
5) Thursday:	_____	_____
6) Friday:	_____	_____
7) Saturday:	_____	_____

Definition of Fields

1.workhrs_name: (WORKHRSNAME) Required

Work hours form identifier. This is a key field.

Legal value: WORKHRS

2.work_hours: (WORKHOURS)

Exchange working hours.

work_hours.start1: (WORKHOURS.START1)

Block 1 start time.

Legal values: HHMM where HH = 00 through 23 and MM = 00 through 59, or null

Default value: none

work_hours.end1: (WORKHOURS.END1)

Block 1 end time.

Legal values: HHMM where HH = 00 through 23 and MM = 00 through 59, or null

Default value: none

work_hours.start2: (WORKHOURS.START2)

Block 2 start time.

Legal values: HHMM where HH = 00 through 23 and MM = 00 through 59, or null

Default value: none

work_hours.end2: (WORKHOURS.END2)

Block 2 end time.

Legal values: HHMM where HH = 00 through 23 and MM = 00 through 59, or null

Default value: none

Form Checks

```
on wh_blocks element entry:
do:
  if start1 is known then end1 must_be known
    error: "start and end times must be input together";
  if end1 is known then start1 must_be known
    error: "start and end times must be input together";
  if start1 is known then end1 must_be_greater_than start1
    error: "end time must be later than the start time";
  if start2 is known then end1 must_be known
    error: "block 1 start and end times must be input before block 2";
  if start2 is known then start2 must_be_greater_than end1
    error: "start time of block 2 must be later than the end time of block 1";
  if start2 is known then end2 must_be known
    error: "start and end times must be input together";
  if end2 is known then start2 must_be known
    error: "start and end times must be input together";
  if start2 is known then end2 must_be_greater_than start2
    error: "end times must be later than the start time";
end;
```


SG Forms

fp

This form describes information about a particular file partition. It should be filled out after the corresponding *pack* form has been entered.

The following information applies to *UNIX*® Real-Time Reliable (RTR) Operating System Release 6.8 and later. For a boot disk only, field 5 (*root_disk*) is set to “y” on the *pack* form corresponding to this form, one *fp* form must exist for each of the following partition usages (field 7.*par_usage*): either **lboot** or **lboot21** (or both) and root. Additionally, one *fp* form whose partition usage is **lboot** and whose *par_primary* field (field 10) contains “primary” must exist in the System Generation (SG) database. If any of these are missing, **trend** will fail and you will not be able to exit **rcvsg** until either the missing *fp* form(s) are inserted or you abort the entire transaction.

Form Layout

```
fp (1/1)
File Partition Header Information (Recent Change And Verify)

1.pack_id
   pack_name: _
   pack_number: _____ 4.par_number: ____

5.par_name: _____ 6.par_flag: _____

7.par_usage: _____ 8.par_ldin: ____

9.par_bkldin: ____ 10.par_primary: _____

11.par_nblk: _____ 12.par_e: _____
```

Definition of Fields

1.pack_id: (PACKID)

A unique name for the disk image. This key field consists of a two-letter name and an integer.

pack_id.pack_name: (*PACKID.PACKNAME) Required

Name given to the disk image to which this record belongs.

Legal values: printable characters

pack_id.pack_number: (*PACKID.PACKNUMBER) Required

Disk pack number.

Legal values: 0 – 32767, 0x0 – 0x7FFF, null

4.par_number: (PARNUMBER) Required

This field specifies the relative position of this entry within the Volume Table of Contents (VTOC) for a nonboot disks but not its physical location in either the VTOC nor on the disk. Thus, a partition with a par_number of 10 will always appear in the VTOC before one with a par_number of 11; however, it will not necessarily reside in position 10 in the VTOC.

On a boot disk, the previous explanation holds true for all partitions except little boot (**lboot** or **lboot21**), big boot primary (**bboot-primary**), and big boot back-up (**bboot-backup**). These are placed in the VTOC by **3bmkdsk** as follows, regardless of the par_number value specified in the *fp* form.

Partition Name	VTOC Type		
	3B20D	3B21D	Common
lboot	0	doesn't exist	0
lboot21	doesn't exist	0	8
bboot-primary	2	2	2
bboot-backup	3	3	3

⇒ NOTE 1:

For *UNIX* RTR Operating System Generic 2 through Release 6.8:
 if par_e field (field 12) is set to a non-null value, the par_number field of this *fp* form must be 0.

⇒ NOTE 2:

For *UNIX* RTR Operating System Release 21.1 and later:
 if par_e field (field 12) is set to a non-null value, the par_number field of this *fp* form must be either 0 or 8.

Legal values: 0 – 63, 0x0 – 0x3F

5.par_name: (PARNAME) Required

Any useful label for a partition.

Legal values: printable characters

6.par_flag: (PARFLAG)

Description of the partition function.

Legal values: reserve and noreserve

Default value: noreserve

7.par_usage: (PARUSAGE) Required

Description of the partition use.

Legal values:

applstart	ecd	lboot	root
bboot	etc	ordinary	swap
free	pdump	unasgn	
bwm	invalid	lboot21	

8.par_ldin: (PARLDIN) Required

Logical partition number. This field is used by software to access the partition. This value corresponds to the partition number in the *logdev* record, if one exists for this file partition.

Legal values: 1 – 62, 0x1 – 0x3E

9.par_bkldin: (PARBKLDIN)

Backup partition logical device number.

Legal values: 0 – 62, 0x0 – 0x3E, null

10.par_primary: (PARPRIMARY) Required

Primary partition. This field should be primary if it is the primary of a primary-backup pair. The field should be backup if it is the backup of a primary-backup pair. Otherwise, it should be simplex.

Legal values: backup, primary, simplex

11.par_nblk: (PARNBLK) Required

The number of blocks to be allocated for the partition.

The maximum legal value possible is calculated from $((2^{32}-1)/NPARTS)$, where *NPARTS* is the maximum number of partitions on a disk. This value is chosen such that when the number of blocks for all the partitions (*NPARTS*) are added together, the processor's 32-bit arithmetic register will not overflow.

Legal values: 1 – 67108863, 0x1 – 0x3FFFFFF

12.par_e: (PARE) *UNIX* RTR Operating System Generic 2 through Release 6.8

Entry point. The core address of little boot entry point. This can be entered only on the *fp* form for the little boot partition (partition 0).

Legal values: 43008, 0xa800, null

12.par_e: (PARE) *UNIX* RTR Operating System Release 21.1 and later

Entry point. The core address of little boot entry point. This can be entered only on the *fp* form for the little boot partition (partition 0) or for lboot21 (partition 8).

Legal values: 43008, 0xa800, null

Form Checks

For *UNIX* RTR Operating System Generic 2 through Release 6.8:

```
if par_e is known
  then par_number must_be 0
      error: "par_number must be 0";
```

For *UNIX* RTR Operating System Release 21.1 and later:

```
if par_e is known
  then par_number must_be_in {0,8}
      error: "par_number must be 0 or 8";
```

fpb

This form allows a file partition to be populated with some information when a disk image is built. The pathnames of files containing the initial data are provided. The *fp* form for the corresponding file partition should already exist.

Form Layout

```

                                     fpb          (1/1)
File Partition Body Information          (Recent Change and Verify)
1.pack_id
   pack_name:___  pack_number:_____  4.par_number:_____
5.part_info          6.srcname          7.swab          8.strip
1) _____          -          -
2) _____          -          -
3) _____          -          -
4) _____          -          -
5) _____          -          -
6) _____          -          -
7) _____          -          -
8) _____          -          -
9) _____          -          -
10) _____          -          -
```

Definition of Fields

1.pack_id: (PACKID)

Packid specifies a unique name for the disk image. It consists of a two-letter name and an integer. This is a key field.

pack_id.pack_name: (*PACKID.PACKNAME) Required

Name given to the disk image to which this record belongs.

Legal values: printable characters

pack_id.pack_number: (*PACKID.PACKNUMBER) Required

Disk pack number.

Legal values: 0 – 32767, 0x0 – 0x7FFF, null

4.par_number: (PARNUMBER) Required

This field is used along with the pack_id field to tie this *fpb* form to the corresponding *fp* form.

Legal values: 0 – 63, 0x0 – 0x3F

5.part_info: (PARTINFO)

Pathname of files with which to initialize the partition.

6.part_info.srcname: (*PARTINFO.SRCNAME[ROW])

Pathname on support processor of file.

Legal values: printable characters, null

7.part_info.swab: (*PARTINFO.SWAB[ROW])

File swab.	c	convert binary object files
	s	swab binary files that are not object files
	n	ASCII files
	null	

Legal values: c, s, n, null

8.part_info.strip: (*PARTINFO.STRIP[ROW])

File strip.

Legal values: a, f, l, null, r, s, x

fs

This form describes a file system to be included in a disk partition. The *pack* form for the corresponding disk image should already exist.

Form Layout

```
fs (1/2)
(Recent Change And Verify)
File System Header Information
1.pack_id
   pack_name: _   pack_number: ____
4.par_number: ____   5.par_name: _____
6.par_flag: _____   7.par_usage: _____
8.par_ldin: ____   9.par_bkldin: ____
10.par_primary: _____
space_allocation 11.par_nblk: ____   12.par_i: ____   13.par_nreg: _____
14.set_uid: _   15.set_gid: _
```

```
fs (2/2)
access_method16.usr: ____   17.grp: ____   18.other: ____
owner_identification19.usr_id: _____   20.grp_id: _____
```

Definition of Fields

1.pack_id: (PACKID)

This is a key field. It specifies a unique name for the disk image. It consists of a two-letter name and an integer.

pack_id.pack_name: (*PACKID.PACKNAME) Required

Name given to the disk image to which this record belongs.

Legal values: printable characters

pack_id.pack_number: (*PACKID.PACKNUMBER) Required

Disk pack number.

Legal values: 0 – 32767, 0x0 – 0x7FFF, null

4.par_number: (PARNUMBER) Required

This field specifies the relative position of this entry within the Volume Table of Contents (VTOC) but not its physical location in either the VTOC nor on the disk. Thus, a file system with a par_number of 10 will always appear in the VTOC before one with a par_number of 11; however, it will not necessarily reside in position 10 in the VTOC.

Legal values: 0 – 63, 0x0 – 0x3F

5.par_name: (PARNAME) Required

Partition name. Any useful name for the partition.

Legal values: printable characters

6.par_flag: (PARFLAG)

Description of the partition function.

Legal values: noreserve, reserve

Default value: noreserve

7.par_usage: (PARUSAGE) Required

Description of the partition usage.

Legal values:

applstart	ecd	lboot	root
bboot	etc	ordinary	swap
free	pdump	unasgn	
bwm	invalid	lboot21	

8.par_idin: (PARLDIN) Required

Logical partition number. This is used by software to access the partition. This value corresponds to the partition number in the *logdev* record, if one exists for this file partition.

Legal values: 1 – 62, 0x1 – 0x3E

9.par_bkldin: (PARBKLDIN)

Logical partition number of the backup partition for this partition (if part of primary-backup pair).

Legal values: 0 – 62, 0x0 – 0x3E, null

10.par_primary: (PARPRIMARY) Required

Primary partition. This is the primary partition of a primary-backup pair.

Legal values: backup, primary, simplex

11.par_nblk: (PARNBLK) Required

The number of blocks to be allocated for the partition.

The maximum legal value possible is calculated from $((2^{32}-1)/NPARTS)$, where NPARTS is the maximum number of partitions on a disk. This value is chosen such that when the number of blocks for all the partitions (NPARTS) are added together, the processor's 32-bit arithmetic register will not overflow.

The value entered must be the size of the disk image being built, less any and all the sizes of the other partitions being placed in the same disk image.

Legal values: 1 – 67108863, 0x1 – 0x3FFFFFF

12.par_i: (PARI) Required

Number of inode blocks in the partition.

Legal values: 1 – 585939, 0x1 – 0x8F0D3

13.par_nreg: (PARNREG) Required

Number of regular file blocks in the partition. A file system is partitioned into contiguous and noncontiguous (or regular) files. This field specifies the number of blocks to devote to the latter.

The maximum legal value possible is calculated from $((2^{32}-1)/NPARTS)$, where NPARTS is the maximum number of partitions on a disk. This value is chosen such that when the number of blocks for all the partitions (NPARTS) are added together, the processor's 32-bit arithmetic register will not overflow.

The value entered must be equal to or less than the value in field 11 (par_nblk), minus the value in field 12 (par_i).

Legal values: 1 – 585939, 0x1 – 0x8F0D3

14.set_uid: (SETUID)

Set user id (run with permissions of the user owning the process).

Legal values: u and -

Default value: -

15.set_gid: (SETGID)

Set group id (run with permissions of the group owning the process).

Legal values: g and -

Default value: -

16.usr: (USR)

User read, write, and execute permissions for the root node of the file system.

Legal values: --- -- x - w - - w x
 r-- r - x r w - r w x

Default value: rwx

17.grp: (GRP)

Group read, write, and execute permissions for the root node for the file system.

Legal values: --- -- x - w - - w x
 r-- r - x r w - r w x

Default value: r-x

18.other: (OTHER)

Other read, write, and execute permissions for the root node of the file system.

Legal values: --- --x -w- -wx
 r-- r-x rw- rwx

Default value: r-x

19.usr_id: (USRID)

The user id of the owner of the file system.

Legal values: 0 – 99999, 0x0 – 0x1869F

Default value: 0

20.grp_id: (GRPID)

The group id of the owner of the file system.

Legal values: 0 – 99999, 0x0 – 0x1869F

Default value: 0

Form Checks

```
par_flag must_not_be unknown  
error: "par_flag must not be null";
```

fsb

This form allows population of a file system by the **3bmkdsk** program. The primary information on an *fsb* form is the location and type of a file on the new file system and the name of a file with which it should be initialized. The *fs* form for the corresponding file system should exist.

Form Layout

		fsb	(1/2)
File System Body Information		(Recent Change and Verify)	
1.pack_id	pack_name:___	pack_number:_____	
4.par_number:_____		5.directory:_____	
6.fname:_____		7.ftype:_	
8.set_uid:_		9.set_gid:_	
10.usr:_____	11.grp:___	12.other:_____	
13.usr_id:_____		14.grp_id:_____	
15.crc_code:_		16.strip:_____	

		fsb	(2/2)
17.extra_blks:_____		18.fldupd_type:_____	
19.srcname:			

Definition of Fields

1.pack_id: (PACKID)

This is a key field. It specifies a unique name for the disk image. It consists of a two-letter name and an integer.

pack_id.pack_name: (*PACKID.PACKNAME) Required

Name given to the disk image to which this record belongs.

Legal values: printable characters

pack_id.pack_number: (*PACKID.PACKNUMBER) Required

Disk pack number.

Legal values: 0 – 32767, 0x0 – 0x7FFF, null

4.par_number: (PARNUMBER) Required

This field is used along with the pack_id field to tie this *fsb* form to the corresponding *fs* form.

Legal values: 0 – 63, 0x0 – 0x3F

5.directory: (DIRECTORY) Optional

Directory pathname. The directory in which the new file will reside. For *UNIX*® Real-Time Reliable (RTR) operating system, special files exist in directories */dev* and */dgn*. This is a key field.

Legal values: printable characters, null

6.fname: (FNAME) Required

The name of the file in the file system built by the **3bmkdsk** process. This is a key field.

Legal values: printable characters

7.ftype: (FTYPE) Required

File type. This field asks what type is the new file.

Legal values:

- b (block special file)
- c (character special file)
- C (contiguous regular file; single extent file)
- d (file corresponds to a directory)
- i (IOP special file)
- o (regular file)
- p (protected pipe)
- r (record special file)
- x (extent file; contains one or more large contiguous areas)

For special files, a specification of c, i, r, or b is required.

8.set_uid: (SETUID)

For an executable file, a "u" indicates that the "set userid" bit is turned on.

Legal values: u, -

Default value: -

9.set_gid: (SETGID)

For an executable file, a "g" indicates that the "set groupid" bit is turned on.

Legal values: g, -

Default value: -

10.usr: (USR)

User read, write, and execution permissions.

Legal values: --- --x -w- - wx
 r-- r-x r w - r w x

Default value: rwx

11.grp: (GRP)

Group read, write, and execution permissions.

Legal values: --- --x -w- - wx
 r-- r-x r w - r w x

Default value: r-x

12.other: (OTHER)

Other read, write, and execution permissions.

Legal values: --- --x -w- - wx
 r-- r-x r w - r w x

Default value: r-x

13.usr_id: (USRID)

User id of the owner of the special file. Special files are owned by the root file system; 0 should be specified.

Legal values: 0 – 99999, 0x0 – 0x1869F

Default value: 0

14.grp_id: (GRPID)

Group id of the owner of the special file. The group owner of special files is the root file system.

Legal values: 0 – 99999, 0x0 – 0x1869F

Default value: 0

15.crc_code: (CRCCODE)

Specifies where the cyclic redundancy check (CRC) information for this file is kept.

Legal values: i (internal), g (global), n (none), null

Default value: n

16.strip: (STRIP)

File strip. The **3bmkdsk** process uses this value to strip a file. The legal values are the same as the options of the *UNIX* system command **3bstrip**, with the exception of the values **a** and **null**. The value **a** indicates no options and **null** indicates do not strip.

Legal values: a, f, l, null, r, s, x

17.extra_blks: (EXTRABLOCKS)

Extra blocks. This is for contiguous files on disk. The number of extra blocks needed for field updates.

Legal values: 0 – 2147483647

Default value: 0

18.fldupd_type: (FLDUPDTYPE) Required

Field update type - integer value.

Legal values:	0 (noupd)	6 (drpcp)	12 (mira)	18 (qkill15)
	1 (boot)	7 (filereplace)	13 (nkfiler)	19 (imcatalog)
	2 (bootfiles)	8 (fport)	14 (nonkill)	20 (omdb)
	3 (btfiler)	9 (gspac)	15 (nub)	21 (ackdb)
	4 (dap)	10 (inhtimer)	16 (pfucode)	22 (kpupdate)
	5 (diag)	11 (kill15)	17 (qkill)	23 (newdcidrv)

19.srcname: (SRCNAME2)

Pathname on the support processor of the file with which to initialize the new file. This is only for ordinary, contiguous, or extent files.

Legal values: printable characters, null

Form Checks

```
case (ftype):  
  
  when {"d"}: do  
    srcname2 must_be unknown  
    error: "srcname must be null";  
  end;  
  
  when {"c", "r", "b", "i"}: do;  
    srcname2 must_be unknown  
    error: "srcname must be null";  
  end;  
  
  when {"p"}: do;  
    srcname2 must_be unknown  
    error: "srcname must be null";  
  end;  
  
  when {"C", "x", "o"}: do;  
    srcname2 must_be known  
    error: "srcname must be known (not null)";  
  end;  
end;
```

klist

This form selectively dumps the keys of key-based forms.

Form Layout

```
Key List                                klist      (1/1)
1.temp_name: _____
2.fp_form:_   3.fpb_form:_   4.fs_form:_   5.fsb_form:_
6.pack_form:_ 7.process_form:_ 8.sgen2_form_: 9.share_form:_
```

Definition of Fields

1.temp_name: (TEMPNAME) Required

The name of the file where keys are to be written. This is a key field.

Legal values: printable characters

2.fp_form: (FPFORM) – 9.share_form: (SHAREFORM)

These fields are used to tell whether to dump keys for the forms indicated by the field name. A “y” indicates that keys should be printed for this form type.

Legal values: y, n, null

pack

This form identifies characteristics of a disk image to be built. It assigns an identifier (two characters and a number) to the image, specifies the disk image size, and declares some default values for the **3bmkdsk** process.

Form Layout

Global Information About Disk	pack	(1/1) (Recent Change and Verify)
1.pack_id	pack_name: _ pack_number: _____	
4.disk_size: _____	5.root_disk: _	
6.verbose: _	7.tape: _	
8.destname:	_____	

Definition of Fields

1.pack_id: (PACKID) Required

Specifies a unique name for the particular disk image. This key field consists of the following two fields.

pack_id.pack_name: (*PACKID.PACKNAME) Required

Name to be given the disk image being built. A two-character identifier.

Legal values: printable characters

pack_id.pack_number: (*PACKID.PACKNUMBER) Required

Disk pack number.

Legal values: 0 – 32767, 0x0 – 0x7FFF, null

4.disk_size: (DISKSIZE)

This gives the size (in megabytes) of the disk image to be built.

Legal values: 160, 300, 322, 340, 600, 1GB, 2GB

Default value: 300

5.root_disk: (ROOTDISK)

This field is "y" when the disk is to be used as a boot device. A boot disk must always contain a little boot partition, big boot partition, and a root file system.

Legal values: y, n, null

6.verbose: (VERBOSE)

Describes the mode in which 3bmkdsk is to run. When "y," a detailed history is recorded.

Legal values: y, n, null

Default value: n

7.tape: (TAPE)

When "y", the disk image is built on a support processor (and put on tape for transfer to the 3B20D/3B21D computers). When "y," files are swabbed to give the correct byte ordering.

⇒ NOTE:

This field is not supported by 3bmkdsk, and the value of the field will be ignored.

Legal values: y, n, null

Default value: n

8.destname: (DESTNAME) Required

The pathname of the file on which the image is to be placed. This is usually a tape device because of the large size of the disk images.

⇒ NOTE:

This field is not supported by 3bmkdsk, and the value of the field will be ignored.

Legal values: printable characters

process

This form describes processes in addition to those on the *sgen2* form that need to be entered in the boot image.

Form Layout

process (1/1)
Processes Other Than Basic UNIX RTR Processes (Recent Change and Verify)

1.boot_prc: _____

2.process_num: _____

3.utility_id: _____

Definition of Fields

1.boot_prc: (BOOTPRC) Required

Pathname of the additional boot process. The most common boot processes are given in the *sgen2* form. This is a key field.

Legal values: printable characters

2.process_num: (PROCESSNUM) Required

The process number to be assigned.

Legal values: 0 – 255

3.utility_id: (UTILITYID) Required

Utility id number.

Legal values: 0 – MAXINT, HEX, null

rcvparams

This form contains RC/V parameter information. It cannot be manually invoked but is invoked automatically, when appropriate, by RC/V.

Form Layout

RCV SG Parameter Info	rcvparams	(1/1) (Recent Change and Verify)
1.database_name:	_____	
2.reviewonly:_	3.journaling:_	
4.print_file:	_____	
5.echo_file:	_____	
6.sysgen:_		

Definition of Fields

1.database_name: Required

Database name. This is a key field.

Legal values: printable characters

2.reviewonly:

Review only.

Legal values: y, n

Default value: y

3.journaling:

Journaling.

Legal values: y, n

Default value: y

4.print_file:

Print file.

Legal values: printable characters, null

5.echo_file:

Echo filename.

Legal values: printable characters, null

6.sysgen:

Legal values: y, n

Default value: n

sgen2

This form specifies the pathnames of the most common processes needed at boot time. Additional processes may be entered in the process form. By convention, all files on the target machine needed for a boot image reside in the */bootfiles* directory.

This form represents the bootfile pathnames on the target machine. On the support machine, the bootfiles need not be in the */bootfiles* directory.

Form Layout

Specifies Required Processes	sgen2	(1/2) (Recent Change and Verify)
1.ksgen2:	_____	
2.kernel:	_____	
3.fmgr:	_____	
4.dfcdrv:	_____	
5.pmgr:	_____	
6.nub:	_____	
7.eih:	_____	

sgen2 (2/2)

8.simprc: _____

9.pcpaud: _____

10.boot_process.pcreate

1) _____

2) _____

3) _____

4) _____

5) _____

6) _____

7) _____

8) _____

9) _____

10) _____

Definition of Fields

All of the fields on sgen2 take printable characters as legal values.

1. ksgen2: (KSGEN2) Required

This is a key field. Because there is only one *sgen2* form in a database, this field always contains "sgen2."

Default value: SGEN2

2. kernel: (KERNEL)

Pathname of combined kernel and kboot.

3. fmgr: (FMGR)

Pathname of file manager.

4. dfcdrv: (DFCDRV)

Pathname of the Disk File Controller (DFC) driver.

5. pmgr: (PMGR)

Pathname of the process manager.

6. nub: (NUB)

Pathname of the nub process.

7. eih: (EIH)

Pathname of the EIH.

8. simprc: (SIMPRC)

Pathname of the System Integrity Monitor (SIM).

9. pcpaud: (PCPAUD)

Pathname of the PCPAUD.

10. boot_process.pcreate (*BOOTPROCESS.PCREATE[ROW])

Pathname of the processes to be pcreated at boot.

share

Two boot processes may share a segment if they are in a parent-child relationship. This form specifies pairs of such processes. These processes must also be listed in either the *sgen2* or process forms.

Form Layout

	share	(1/1)
Share Segment Through Parent-Child Relationship	(Recent Change and Verify)	
1.parent:	_____	
2.child:	_____	

Definition of Fields

1.parent: (PARENT) Required

This specifies the pathname of the parent. This is a key field.

Legal values: printable characters

2.child: (CHILD) Required

This specifies the pathname of the child. This is a key field.

Legal values: printable characters

ECD/SG Forms

dbdump

This form dumps the database into a special format that is used by the *dbload* form to recreate the database. The dump format is the format used in database evolution. A dump and load [refer to “Reclaiming Fragmented Database Space (**dbdump** and **dbload**)” in Chapter 5] can be used to collect free space in the Equipment Configuration Database (ECD) or the System Generation (SG) database. This form cannot be used on an *incore* database.

Form Layout

Dump Database Information	dbdump	(1/1)	(Execute Only)
1.request_file: _____			
2.load_file: _____			

Definition of Fields

1.request_file: (REQUESTFILE) Required

This is a key field. This is a temporary file used to build a load file. A full pathname should be specified.

Legal values: printable characters

2.load_file: (LOADFILE) Required

This is a key field. File containing database for subsequent load. This file is used in the *dbload* form. A full pathname should be specified.

Legal values: printable characters

dbload

This form loads a database from the special format file created by the *dbdump* form in the name given in field 2 of the *dbdump* form.

⇒ NOTE:

A *dbload* form should only be performed to a newly created skeleton database. A skeleton database can be generated using the **CREATEECD** or **CREATESG** command. (Refer to the “**CREATEECD** and **CREATESG** Commands” in Chapter 4.)

Form Layout

Load Database Information	dbload	(1/1)	(Execute Only)
1.load_file:	_____		
2.error_file:	_____		

Definition of Fields

1.load_file: (LOADFILE) Required

This is a key field. This field should specify the full path to the load file built previously from *dbdump*.

Legal values: printable characters

2.error_file: (ERRORFILE) Required

This is a key field. This field should contain the full path to an error file to which forms that fail checks will be printed.

Legal values: printable characters

prtlist

This form prints form instances of the key-based forms. The keys must be in a file that is the same as that produced by *klist*.

Form Layout

Print Form Instances	prtlist	(1/1)	(Execute Only)
1.keylist_file:	_____		
2.print_list_file:	_____		

Definition of Fields

1.keylist_file: (KEYLISTFILE) Required

This is a key field. This field should contain the complete path to a file which was previously produced by *klist*.

Legal values: printable characters

2.print_list_file: (PRINTLISTFILE) Required

This is a key field. This field should contain the full path to an output file to which the form instances for the keys in the *keylist_file* will be printed.

Legal values: printable characters

trabort

This form aborts the current transaction (a logically consistent group of database changes). Any database modifications made since the last trbegin will be removed.

Form Layout

Transaction Abort	trabort	(1/1) (Execute Only)
1.tr_name:_____		

Definition of Fields

1.tr_name: (TRNAME) Required

Transaction name. This is a key field.

Legal values: printable characters

Default value: TRABORT

trbegin

This form signals the beginning of a transaction (a logically consistent group of database changes).

Form Layout

Transaction Begin	trbegin	(1/1) (Execute Only)
1.tr_name: _____		

Definition of Fields

1.tr_name: (TRNAME) Required

Transaction name. This is a key field.

Legal values: printable characters

Default value: TRBEGIN

trend

This form applies a transaction (a logically consistent group of changes) to the database.

Form Layout

Transaction End	trend	(1/1) (Execute Only)
1.tr_name: _____		
2.dis_cf_checks: _		
3.apply_tr: _		
4.o_missing_links: _		

Definition of Fields

1.tr_name: (TRNAME) Required

Transaction name. This is a key field.

Legal values: printable characters

Default value: TREND

2.dis_cf_checks: (DISCFCHECKS)

Disable cross-form checks. Cross-form checks ensure data integrity between forms of different types.



CAUTION:

This option should always be "n" in normal processing. Otherwise, database integrity may be lost.

Legal values: y, n

Default value: n

3.apply_tr: (APPLYTR)

Apply transaction. Move transaction table from buffer to test state. If this flag is "y," the transaction is still active.



CAUTION:

This option should always be "y" in normal processing. Otherwise, database integrity may be lost.

Legal values: y, n

Default value: y

4.o_missing_links: (OMISSINGLINKS)

Override missing links.

Legal values: y, n

Default value: n



NOTE:

This field is nonfunctional on the 3B20D/3B21D computers. Links will be resolved regardless of the field setting when running on the computers.

Contents

Introduction	7-1
High-Level RC/V Tools	7-1
■ RCVECD Command	7-1
■ VFYDFLT Command	7-3
Using High-Level RC/V Tools	7-4
■ Growing and Degrowing IOP, DFC, and DCI Devices	7-4
■ Listing All Devices Under an IOP	7-7
■ Verifying Low-Level ECD Form Defaults	7-7
Default Files	7-7
High-Level RC/V Database Forms Guide	7-8
■ <i>dciadd</i>	7-9
■ <i>dcidel</i>	7-12
■ <i>formform</i>	7-14
■ <i>iopadd</i>	7-17
■ <i>iopdel</i>	7-19
■ <i>ioppc</i>	7-21
■ <i>iopslots</i>	7-23
■ <i>mtadd</i>	7-25
■ <i>mtcadd</i>	7-28
■ <i>mtcdel</i>	7-31
■ <i>mtdel</i>	7-33
■ <i>rcvparams</i>	7-35
■ <i>sccadd</i>	7-37
■ <i>sccdel</i>	7-42
■ <i>scsiadd</i>	7-45

Contents

■ <i>sdladd</i>	7-49
■ <i>sdlcadd</i>	7-59
■ <i>sdlcdel</i>	7-61
■ <i>sldel</i>	7-63
■ <i>ttyadd</i>	7-67
■ <i>ttycadd</i>	7-72
■ <i>ttycdel</i>	7-74
■ <i>ttydel</i>	7-76
■ <i>vfydflt</i>	7-78

Introduction

This chapter contains information concerning the optional high-level RC/V feature. This feature consists of a high-level Recent Change and Verify (RC/V) form interface which facilitates the growth and degrowth of Input/Output Processor (IOP), Small Computer System Interface (SCSI) Disk File Controller (DFC), and Dual Serial Channel (DSCH) Computer Interconnect (DCI) devices in the Equipment Configuration Database (ECD). The feature has no impact on *rcvsg* or the System Generation (SG) database.

High-Level RC/V Tools

RCVECD Command

Additional RCVECD Options: Described in Chapter 4, the **RCVECD** command is the only command affected by the high-level Recent/Change/Verify (RC/V) feature. Two additional command line options have been added as follows:

HLMODE | NHLMODE

Default = NHLMODE

HLMODE enables the high-level RC/V feature. **NHLMODE** disables the high-level RC/V feature. This option determines the initial state of the RC/V. Once RC/V is running, the high-level mode can be turned OFF or ON via the “toggle” capability.

DEFAULTS "defaultdir"

Default = *//la/defaults*

Full or relative pathname of the directory containing the low-level ECD form default files.

Control Display Terminal: The **RCVECD** command is still invoked via the *199* menu option on the control display terminal. The *rcvparams* will include the two new options previously listed. The PF function keys (refer to "Control Display Terminal" in Chapter 4) are particularly useful in the context of the high-level RC/V feature. When running on the *incore* database, the user may be prompted to perform certain shell messages or Display Administration Process (DAP) menu commands. The PF function keys allow the user to access both the message and menu command areas of the screen without exiting RC/V. The PF function keys are illustrated in Figure 4-1.

RC/V Session Flow: In addition to the form selection prompt page elements (refer to "RC/V Session Flow" in Chapter 4), the mode of operation (high level or low level) is displayed in the upper right corner of the screen. The following help message appears at the bottom of the screen:

```
may also enter '?' or 'help' to list forms or 'toggle' to  
switch mode
```

The "toggle" provides a means of switching back and forth between high- and low-level modes of operation in the course of a single RC/V session. Entering **?** or **help** at this point will provide a list of all forms available to the user in the current mode.

The only other difference in session flow concerns the use of **trbegin-trend** pair in a high-level, NREVIEW session. Whereas in the low-level mode, the user must be inside a **32trbegin-trend** pair in order to perform insert, update, or delete database operations; in high-level mode, explicit execution of **trbegin** and **trend** by the user is not allowed. Transaction blocking for high-level forms is done implicitly in the course of the underlying processing for that form. If the session is NREVIEW, all legal operations for each high-level form are available as long as the high-level mode is in effect.

Serial Text Messages RC/V: The high-level RC/V feature is **not** supported for text RC/V.

VFYDFLT Command

The **VFYDFLT** command is used to verify that the low-level ECD form default files used during high-level fork processing contain legal field values [that is, pass On-line Data Integrity (ODIN) domain checks]. The command may be executed via the high-level *vfydfit* form (see “*vfydfit*” in this chapter) or through the use of **MENU** and RC/V craft shell interface as follows.

In PDS,

```
RCV:MENU:VFYDFLT, EFILE "errorfile", DFILE \  
"defaultfile" [DDIR "defaultf"] [,MDIR "maskdir"]
```

In MML,

```
RCV:MENU:DATA, VFYDFLT, EFILE="errorfile",\  
DFILE="defaultfile"[,DDIR="defaultdir"] [,MDIR="maskdir"]
```

The command options are:

EFILE "errorfile"

errorfile is the full or relative pathname of the file to which any errors found in the specified default file will be written.

DFILE "defaultfile"

defaultfile is the name of the default file to be verified. The default files have the same name as the low-level mask associated with them.

DDIR "defaultdir"

Default = *//a/defaults*

defaultdir is the full or relative directory pathname of the directory containing the default files.

MDIR "maskdir"

Default = *//a/ecdmasks*

maskdir is the full or relative pathname of the directory containing the ODIN masks.

Using High-Level RC/V Tools

Growing and Degrowing IOP, DFC, and DCI Devices

The high-level forms described in the next section are used primarily to grow and degrow the IOP devices (iop, mt, mtc, tty, ttyp, sdl, and sdlc), the following SCSI DFC devices (mhd, dat, and mt) and DCI devices. The high-level forms are also used to grow and degrow DFC devices. The corresponding forms require the user to input a minimum of information, then the underlying high-level form process generates the associated transactions and applies them to the ECD. When accessing on the *incore* database, the user is prompted to perform craft commands, such as removing, restoring, or diagnosing a unit at various points during processing. The high-level RC/V feature thus allows the user to grow and degrow devices without having to know the low-level ECD forms required, their field values, their order of application, or the number of transactions required. Because most of the steps involved in these activities will be done automatically, the opportunity for human error is greatly reduced, and the time and effort involved are minimized.

The following provides an example of a high-level form operation. In this instance, the user wants to add a Magnetic Tape Unit, Magnetic Tape (MT) 3, under Magnetic Tape Controller (MTC) 3. The operation is being performed on the *incore* database.

In response to the form selection prompt, the user enters **mtadd**. Because the *mtadd* form is insert-only, the form operation prompt is not displayed. The *mtadd* form is displayed, and the user proceeds to fill it out as follows.

```
Addition of Magnetic Tape          mtadd          (1/2)
                                     (Insert Only)
1.unit_name: MT                    2.unit_number: 3
3.pcsdid: 3                        4.mdct_name: mt3
controller
5.cntl_unit_name: MTC              6.cntl_unit_number: 3
previous logical unit
7.plu_unit_name:_____            8.plu_unit_number:_____
```

```
9.logdevs      logical_name      partition
1) /dev/mt3c   12
2) /dev/mt38   8
3) _____  ___
4) _____  ___
```

When the form is completed, the user is prompted as follows:

Enter Insert, Change, Substitute, Validate, Screen#, or Print:

The user enters **i** to insert the form.

At this point, the underlying high-level form process begins the work of growing the MT. Because the *incore* database is being accessed, three transactions are required. At several points during the processing, acknowledgment messages are displayed at the bottom of the screen to apprise the user of progress. The actual steps performed during processing are as follows:

1. Verify that MT 3 does not already exist in the database. (This step was performed immediately after the key fields were entered.)
2. Verify that the previous logical unit (whether specified by the user or not) and the controller are of the correct device types.

3. Verify that the *pcsdid* form is not already in use.

4. Transaction 1:

The following acknowledgment messages document the underlying activities:

- Executing *trbegin*...
- Inserting MT 3 in "UNEQIP" state...
- Inserting *mdct mt3*...
- Inserting *logdev /dev/mt3c*...
- Inserting *logdev /dev/mt38*...
- Executing *trend*...

5. The user will be prompted to perform the following activities:

- Remove MTC 3 from service and power it down. Press the <CR> key to continue.
- Physically connect MT 3. Press the <CR> key to continue.
- Power up and restore MTC 3. When All Tests Pass (ATP), press the <CR> key to continue.
- Remove MTC 3 from service. Press the <CR> key to continue.

6. Transaction 2:

- Executing *trbegin*...
- Updating MTC 3 to point to MT 3...
- Updating MT 3 to "GROW" state...
- Executing *trend*...

7. The user will be prompted to perform the following activities:

- Restore and diagnose MTC 3. When complete, press the <CR> key to continue.
- Run diagnostic demand Phase 5 on MTC 3. When complete, press the <CR> key to continue.

8. Transaction 3:

- Executing *trbegin*...
- Updating MT 3 to "OOS" state...
- Executing *trend*...

The form has now successfully completed execution. The `FORM INSERTED!!` message will flash in the upper right corner of the screen, and the cursor will return to the first field. The user will only restore MT 3 to bring it into the active state.

Had this operation been performed on a *disk* database, only one transaction would have occurred and the user would not have been prompted to perform any craft tasks. The text of the acknowledgment message may have differed slightly, but in general, the sequence of events would have remained the same.

Listing All Devices Under an IOP

To list all devices under a particular IOP, use the *ioppc* form or the *iopslots* form. Both are review-only forms; the *ioppc* form lists the devices in link order, the *iopslots* form lists them by slot number.

The keys for both forms are the IOP unit name and unit number.

Verifying Low-Level ECD Form Defaults

Low-level ECD form default files supply most of the values needed to populate the low-level forms generated during processing of a high-level form. The low-level form field values listed in these files can be verified (that is, subjected to ODIN domain checks) while within a RC/V session by means of the *vfydfit* (execute-only) form. (This tool is also available through the **MENU** command described in the previous section.)

The key for **vfydfit** is the name of the file to which errors in the specified default file are to be printed. The remaining input consists of the default filename and the default file and mask directories (if other than the official directories). When the execution is successful, an acknowledgment message will be printed at the bottom of the form indicating if errors were found. When the execution is not successful, the user will be directed to check the error file, if necessary. Exit RC/V to view the error file.

Default Files

The low-level ECD form default files used by the high-level RC/V feature are located in */lla/defaults*. They are ASCII files and may be modified with any available text editor. These files may be viewed via the **DUMP:FILE:ALL** command. (Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.)

High-Level RC/V Database Forms Guide

This section illustrates the forms associated with the high-level RC/V feature and serves as a guide in filling out these forms. The forms documented here are specific to the high-level RC/V feature with the exception of the *rcvparams* form. The version of the *rcvparams* form in this chapter reflects the additional command line parameters required by the feature. *All forms listed in this chapter are accessible by the user only when the high-level mode is turned on.* Conversely, the low-level forms listed in Chapter 6 are accessible only when the high-level mode is turned off. Table 7-1 list these forms, their description, and their operation modes.

Table 7-1. High-Level RC/V Forms

Form	Operation Mode	Description
<i>dciadd</i>	i	Addition of a Dual Serial Channel (DSCH) Computer Interconnect
<i>dcidel</i>	d	Deletion of a Dual Serial Channel Computer Interconnect
<i>formform</i>	e	High-level form / process correlation
<i>iopadd</i>	i	Addition of an IOP
<i>iopdel</i>	d	Deletion of an IOP
<i>ioppc</i>	r	Review of IOP PC linkages
<i>iopslots</i>	r	Review of IOP slot assignments
<i>mtadd</i>	i	Addition of a magnetic tape device
<i>mtcadd</i>	i	Addition of a magnetic tape controller
<i>mtcdel</i>	d	Deletion of a magnetic tape controller
<i>mtdel</i>	d	Deletion of a magnetic tape device
<i>rcvparams</i>	e	RCVECD command parameters
<i>sccadd</i>	i	Addition of a Switching Control Center (SCC) link
<i>sccdcl</i>	d	Deletion of an SCC link
<i>scsiadd</i>	i	Addition of an SCSI peripherals
<i>sdladd</i>	i	Addition of a synchronous data link device
<i>sdlcadd</i>	i	Addition of a synchronous data link controller
<i>sdlcdcl</i>	d	Deletion of a synchronous data link controller
<i>sdlldcl</i>	d	Deletion of a synchronous data link device
<i>ttyadd</i>	i	Addition of a terminal
<i>ttycadd</i>	i	Addition of a terminal controller
<i>ttycdcl</i>	d	Deletion of a terminal controller
<i>ttydel</i>	d	Deletion of a terminal
<i>vfydffl</i>	e	Default file verification

⇒ NOTE:

Because text RC/V is **not** supported for this feature, the text RC/V version of field names will not be included in the form listings.

dciadd

This form adds a DCI.

Form Layout

```

                                     dciadd      (1/1)
Addition of a Dual-Serial Channel Computer Interconnect      (Insert Only)

  1.unit_name:_____                2.unit_number:____
  3.device_num:___                    4.adj_prcsr_id:_____
  5.logical_channels: ___
controller
  6.cntl_unit_name:_____            7.cntl_unit_number:____
previous logical unit
  8.plu_unit_name:_____             9.plu_unit_number:____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: DCI

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. device_num: Required

The address of the device on the channel.

Legal values: 0 – 15

4. `adj_pcrs_id`:

The identifier of the adjunct processor to which the 3B20D/3B21D computer is attached via the DCI cable.

Legal values: 0 – 3

⇒ NOTE:

A DCI device is a logical unit defined in a *ucb* form in the ECD; it is not the adjunct processor.

5. `logical_channels`:

The number of DCI application logical channels to be defined for the DCI.

The *logdev* records reserved for the DCI Application Interface will be automatically inserted by *dciadd* for logical channels 240 through 255. These logical channels should not be used by application processes.

If the `logical_channels` field is set to a non-zero value, then additional *logdev* records will be inserted starting with 239 and decrementing from there. These logical channels are available for application use.

For example, if the `logical_channels` field is set to 20, then logical channels 239 through 220 will be configured for application use as well as logical channels 255 through 240 which are always configured and reserved for DCI Application Interface use.

Legal values: 0 – 240

Default value: 20

6. `cntl_unit_name`: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: CH

7. `cntl_unit_number`: Required

The UCB unit number of the controller.

Legal values: 0 – 255

8. plu_unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

Legal values: alphanumeric, null

Default value: set to the last ucb in the chain

9. plu_unit_number:

The UCB unit number of the previous logical unit in the ucb linkage chain.

Legal values: 0 – 255, null

Default value: set to the last ucb in the chain

dcidel

This form deletes a DCI.

Form Layout

dcidel (1/1)	
Deletion of a Dual-Serial Channel Computer Interconnect (Delete Only)	
1. unit_name: _____	2. unit_number: _____
3. device_num: _____	4. adj_prcsr_id: _____
controller	
5. cntl_unit_name: _____	6. cntl_unit_number: _____
previous logical unit	
7. plu_unit_name: _____	8. plu_unit_number: _____

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: DCI

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. device_num:

The address of the device on the channel.

4. `adj_pcrsr_id`:

The identifier of the adjunct processor to which the 3B20D/3B21D computer is attached via the DCI cable.

⇒ NOTE:

A DCI device is a logical unit defined in a *ucb* form in the Equipment Configuration Database (ECD); it is not the adjunct processor.

5. `cntl_unit_name`:

The UCB unit name of the controller.

6. `cntl_unit_number`:

The UCB unit number of the controller.

7. `plu_unit_name`:

The UCB unit name of the previous logical unit in the *ucb* linkage chain.

8. `plu_unit_number`:

The UCB unit number of the previous logical unit in the *ucb* linkage chain.

formform

This form correlates high-level forms to high-level processes.

Form Layout

formform (1/3)

High-Level Form / Process Correlation (Execute Only)

1. ff_name: _____

2. hl_prsss_dir: _____

	hl_form	hl_prsss
3. fp1	4) _____	5) _____
6. fp2	7) _____	8) _____
9. fp3	10) _____	11) _____
12. fp4	13) _____	14) _____
15. fp5	16) _____	17) _____
18. fp6	19) _____	20) _____
21. fp7	22) _____	23) _____
24. fp8	25) _____	26) _____

formform (2/3)

	hl_form	hl_prsss
27. fp9	28) _____	29) _____
30. fp10	31) _____	32) _____
33. fp11	34) _____	35) _____
36. fp12	37) _____	38) _____
39. fp13	40) _____	41) _____
42. fp14	43) _____	44) _____
45. fp15	46) _____	47) _____
48. fp16	49) _____	50) _____
51. fp17	52) _____	53) _____
54. fp18	55) _____	56) _____
57. fp19	58) _____	59) _____
60. fp20	61) _____	62) _____
63. fp21	64) _____	65) _____

formform (3/3)

66.fp22 hl_form hl_prCSS
67)_____ 68)_____

Definition of Fields:

1. ff_name:

This is the key field. It should be FORMFORM.

Default value: FORMFORM

2. hl_prCSS_dir:

The pathname to the directory containing the high-level processes (hl_prCSS). If no path is given, the \$PATH environment variable is used to search for the process.

Default value: null

3. fp1:

Structure containing a form versus high-level process pair.

4. fp1.hl_form:

The high-level form (mask) name.

Legal value: printable

5. fp1.hl_prCSS:

The corresponding high-level process name.

Legal value: printable

6. – 66. fp2 – fp22:

The remaining structures for the form versus high-level processes. The following table shows the current defaults.

	hl_form	hl_prCSS
fp1:	4. iopadd	5. iopadd.p
fp2:	7. iopdel	8. iopdel.p
fp3:	10. mtadd	11. mtadd.p
fp4:	13. mtdel	14. mtdel.p
fp5:	16. mtcadd	17. mtcadd.p
fp6:	19. mtcdel	20. mtcdel.p
fp7:	22. sdladd	23. sdladd.p
fp8:	25. sldel	26. sldel.p
fp9:	28. sdlcadd	29. sdlcadd.p
fp10:	31. sdlcdel	32. sdlcdel.p

fp11:	34. ttyadd	35. ttyadd.p
fp12:	37. ttydel	38. ttydel.p
fp13:	40. tycadd	41. tycadd.p
fp14:	43. tycdel	44. tycdel.p
fp15:	46. ioppc	47. links.p
fp16:	49. iopslots	50. slots.p
fp17:	52. vfydflt	53. vfydflt.p
fp18:	55. sccadd	56. sccadd.p
fp19:	58. sccdel	59. sccdel.p
fp20:	61. scsiadd	62. scsiadd.p
fp21:	64. dciadd	65. dciadd.p
fp22:	67. dcidel	68. dcidel.p

iopadd

This form adds an Input/Output Processor (IOP).

Form Layout

```

                                     iopadd      (1/1)
Addition of Input/Output Processor                                     (Insert Only)
  1.unit_name:_____          2.unit_number:____
  3.device_num:___            4.u_model:___
controller
  5.cntl_unit_name:_____    6.cntl_unit_number____
previous logical unit
  7.plu_unit_name:_____     8.plu_unit_number:____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: IOP

2. unit_number: Required

The UCB unit number. This a key field.

Legal values: 0 – 255, HEX

3. device_num: Required

The address of the device on the channel.

Legal values: 0 – 15

4. u_model: Required

The model information for a major hardware unit.

Legal values: 0 – 255 (See *ucb* form for explanation of values.)

5. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: CH

6. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255

7. plu_unit_name:

The UCB unit name of the previous logical unit in the *ucb* linkage chain.

Legal values: alphanumeric, null

Default value: set to the last *ucb* in the chain

8. plu_unit_number:

The UCB unit number of the previous logical unit in the *ucb* linkage chain.

Legal values: 0 – 255, null

Default value: set to the last *ucb* in the chain

iopdel

This form deletes an Input/Output Processor (IOP).

Form Layout

Deletion of Input/Output Processor		iopdel	(1/1)	(Delete Only)
1.unit_name:_____		2.unit_number:_____		
3.device_num:___				
controller				
4.cntl_unit_name:_____		5.cntl_unit_number_____		
previous logical unit				
6.plu_unit_name:_____		7.plu_unit_number:_____		

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: IOP

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read only and are supplied by the high-level process.

3. device_num:

The address of the device on the channel.

4. cntl_unit_name:

The UCB unit name of the controller.

5. cntl_unit_number:

The UCB unit number of the controller.

6. plu_unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

7. plu_unit_number:

The UCB unit number of the previous logical unit in the ucb linkage chain.

ioppc

This form reviews the logical Peripheral Controller (PC) linkages under an Input/Output Processor (IOP).

Form Layout

Review of IOP PC Linkages		ioppc	(1/1)	(Review Only)
1.unit_name:_____		2.unit_number:_____		
3.pc(in link order)	unitname	unitnumber	unitname	unitnumber
	1)_____	_____	9)_____	_____
	2)_____	_____	10)_____	_____
	3)_____	_____	11)_____	_____
	4)_____	_____	12)_____	_____
	5)_____	_____	13)_____	_____
	6)_____	_____	14)_____	_____
	7)_____	_____	15)_____	_____
	8)_____	_____	16)_____	_____

Definition of Fields:

1. unit_name: Required

The UCB unit name of the IOP controller being reviewed. This is a key field.

Legal values: alphanumeric

Default value: IOP

2. unit_number: Required

The UCB unit number of the IOP controller being reviewed. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pc:

The PC unit in the specified IOP.

pc.unitname:

The unit name of the PC.

pc.unitnumber:

The unit number of the PC.

iopslots

This form reviews the physical Peripheral Controller (PC) slots under an Input/Output Processor (IOP).

Form Layout

Review of IOP Slot Assignment		iopslots	(1/1)	(Review Only)
1.unit_name:_____		2.unit_number:_____		
3.slot				
unitname	unitnumber	unitname	unitnumber	
0)_____	_____	8)_____	_____	
1)_____	_____	9)_____	_____	
2)_____	_____	10)_____	_____	
3)_____	_____	11)_____	_____	
4)_____	_____	12)_____	_____	
5)_____	_____	13)_____	_____	
6)_____	_____	14)_____	_____	
7)_____	_____	15)_____	_____	

Definition of Fields:

1. unit_name: Required

The UCB unit name of the IOP controller being reviewed. This is a key field.

Legal values: alphanumeric

Default value: IOP

2. unit_number: Required

The UCB unit number of the IOP controller being reviewed. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. slot:

The PC position in the specified IOP.

slot.unitname:

The unit name of the PC.

slot.unitnumber:

The unit number of the PC.

mtadd

This form adds a Magnetic Tape (MT).

Form Layout

Addition of Magnetic Tape		mtadd	(1/2)	(Insert Only)
1. unit_name: _____		2. unit_number: _____		
3. pcsdid: _____		4. mdct_name: _____		
controller				
5. cntl_unit_name: _____		6. cntl_unit_number _____		
previous logical unit				
7. plu_unit_name: _____		8. plu_unit_number: _____		

		mtadd	(2/2)
9. logdevs	logical_name	partition	
	1) _____	_____	
	2) _____	_____	
	3) _____	_____	
	4) _____	_____	

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: MT

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcsdid: Required

Peripheral controller subdevice id. This is the backplane connector position.

Legal values: 0 – 3

4. mdct_name: Required

The name of the associated *mdct* to be created.

Legal values: printable

5. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: MTC

6. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

7. plu-unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

Legal values: alphanumeric, null

Default value: set to the last ucb in the chain

8. plu_unit_number:

The UCB unit number of the previous logical unit in the ucb linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last ucb in the chain

9. logdevs:

The logdevs (special device files) associated with the drive. These must not exist in the database.

logdevs.logical_name:

The logical device name must begin with */dev/*.

Legal values: alphanumeric, null

logdevs.partition:

Partition number

Legal values: 0 – 63, null

mtcadd

This form adds a Magnetic Tape Controller (MTC).

Form Layout

```

                                     mtcadd      (1/1)
Addition of Magnetic Tape Controller      (Insert Only)
  1.unit_name:_____                2.unit_number:____
  3.pcid:_                4.packname:_____  5.drive:____
controller
  6.cntl_unit_name:_____                7.cntl_unit_number____
previous logical unit
  8.plu_unit_name:_____                9.plu_unit_number:____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: MTC

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcid: Required

Peripheral controller id. This is the Input/Output Processor (IOP) slot position.

Legal values: 0 – 15

4. packname: Required

The circuit pack name code.

Legal values: un32, un134, un52, un145

5. drive: Required

The magnetic tape drive model name.

Legal values: ken, key2, key3

6. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: IOP

7. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

8. plu_unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

Legal values: alphanumeric, null

Default value: set to the last ucb in the chain

9. plu_unit_number:

The UCB unit number of the previous logical unit in the ucb linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last ucb in the chain

Form Checks:

```
packname must_be_in {"un32", "un52", "un134", "un145"}
    error:"packname must be un32, un52, un134 or un145";

case (packname);
    when {"un32", "un134"}: do;
        drive must_be_in {"KEN"}
        error: "for packname un32 or un134,
            drive must be KEN";
    end;

    when {"un52", "un145"}: do;
        drive must_be_in {"KEY2", "KEY3"}
        error: "for packname un52 or un145,
            drive must be KEY2 or KEY3";
    end;

end;
```

mtcdel

This form deletes a Magnetic Tape Controller (MTC).

Form Layout

Deletion of Magnetic Tape Controller		mtcdel	(1/1)	(Delete Only)
1. unit_name: _____		2. unit_number: _____		
3. pcid: _____				
controller				
4. cntl_unit_name: _____		5. cntl_unit_number _____		
previous logical unit				
6. plu_unit_name: _____		7. plu_unit_number: _____		

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: MTC

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcid:

Peripheral controller id. This is the Input/Output Processor (IOP) slot position.

4. cntl_unit_name:

The UCB unit name of the controller.

5. cntl_unit_number:

The UCB unit number of the controller.

6. plu_unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

7. plu-unit_number:

The UCB unit number of the previous logical unit in the ucb linkage chain.

mtdel

This form deletes a Magnetic Tape (MT).

Form Layout

Deletion of Magnetic Tape		mtdel	(1/1)	(Delete Only)
1.unit_name:_____		2.unit_number:_____		
3.pcsdid:_____		4.mdct_name:_____		
controller				
5.cntl_unit_name:_____		6.cntl_unit_number_____		
7.logdevs	logical_name	partition		
1)_____		—		
2)_____		—		
3)_____		—		
4)_____		—		

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: MT

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcsdid:

Peripheral controller subdevice id. This is the backplane connector position.

4. mdct_name:

The name of the associated *mdct* to be created.

5. cntl_unit_name:

The UCB unit name of the controller.

6. cntl_unit_number:

The UCB unit number of the controller.

7. logdevs:

The logdevs (special device files) associated with the drive.

logdevs.logical_name:

The logical device name must begin with */dev/*.

logdevs.partition:

Partition number

rcvparams

This form contains Recent Change and Verify (RC/V) parameters. It cannot be manually invoked but is invoked automatically, when appropriate, by RC/V.

Form Layout

```
rcvparams      (1/1)
RCV ECD Parameter Info      (Execute Only)
1.database_name:_____
2.reviewonly:_____      3.journaling:_____
4.print_file:_____
5.echo_file:_____
6.sysgen:_____      7.flat_database:_____      8.hlmode:_____
9.default_dir:_____
```

Definition of Fields:

1. database_name: Required

Database name. This is a key field.

Legal values: printable characters

2. reviewonly:

Review option.

Legal values: y, n

Default value: y

3. journaling:

Journaling option.

Legal values: y, n

Default value: y

4. print_file:

Print filename.

Legal values: printable characters, null

5. echo_file:

Echo filename.

Legal values: printable characters, null

6. sysgen:

System generation option. Used for initial load of the database.

Legal values: y, n

Default value: n

7. flat_database:

Contiguous file type database.

Legal values: y, n

Default value: n

8. hlmode:

High-level RC/V option.

Legal values: y, n

Default value: n

9. default_dir:

Pathname to directory containing default files used by high-level RC/V.

Legal values: printable characters, null

Default value: null



NOTE:

For a null value, the official directory *//la/defaults* is used.

sccadd

This form adds a Switching Control Center (SCC) link.

Form Layout

```

                                     sccadd      (1/4)
Addition of SCC Link                                     (Insert Only)
  1.unit_name:_____  2.unit_number:____  3.pcsdid:___
      4.mdct_name:_____  5.option_name:_____
      6.duplex:___  7.status:_____
controller
  8.cntl_unit_name:_____  9.cntl_unit_number:____
previous logical unit
 10.plu_unit_name:_____  11.plu_unit_number:____
```

```

                                     sccadd      (2/4)
12.logdevs      logical_name      partition      channel
  1) _____      _____      _____
  2) _____      _____      _____
  3) _____      _____      _____
  4) _____      _____      _____
  5) _____      _____      _____
  6) _____      _____      _____
  7) _____      _____      _____
  8) _____      _____      _____
  9) _____      _____      _____
 10) _____      _____      _____
```

sccadd (3/4)

12.logdevs(cont)	logical_name	partition	channel
11)	_____	—	—
12)	_____	—	—
13)	_____	—	—
14)	_____	—	—
15)	_____	—	—
16)	_____	—	—
17)	_____	—	—
18)	_____	—	—
19)	_____	—	—
20)	_____	—	—

sccadd (4/4)

12.logdevs(cont)	logical_name	partition	channel
21)	_____	—	—
22)	_____	—	—
23)	_____	—	—
24)	_____	—	—
25)	_____	—	—
26)	_____	—	—
27)	_____	—	—
28)	_____	—	—
29)	_____	—	—
30)	_____	—	—
31)	_____	—	—

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: SDL

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcsdid: Required

Peripheral controller subdevice id. This is the backplane connector position.

Legal values: 0 – 3

4. mdct_name:

The name of the associated *mdct*. This *mdct* may already exist in the database if this request is to duplex the Synchronous Data Link (SDL).

Legal values: printable, null

5. option_name: Required

The name of the associated option record to be added (cpblx3 type for handler = sdl, nphopt for handler = np).

Legal values: alphanumeric

6. duplex: Required

Is this a duplexed data link?

Legal values: y, n

Default value: n

7. status:

Is this the active or standby unit of a duplexed pair?

Legal values: active, standby, null

8. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: SDLC

9. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

10. plu_unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

Legal values: alphanumeric, null

Default value: set to the last ucb in the chain

11. plu_unit_number:

The UCB unit number of the previous logical unit in the ucb linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last ucb in the chain

12. logdevs:

The logdevs (special device files) associated with the drive. These must not already exist in the database.

logdevs.logical_name:

The logical device name must begin with */dev/*.

Legal values: alphanumeric, null

logdevs.partition:

Partition number

Legal values: 1 – 31, null

logdevs.channel:

Channel number associated with this logdev.

Legal values: 1 – 4095, null

List Element Checks:

```
on logdevs element entry:
  if logical_name is known then partition must_be_between 1 and 31;
  error: "partition must be between 1 and 31 when
  logical_name is known";
```

Form Checks:

```
mdct_name must_not_be unknown
  error: "mdct_name must be known when handler is 'sdl'";

if duplex is "y"
  then status must_be known
  error: "if duplex is y status must not be NULL";
end;
```

sccdel

This form deletes an SCC link.

Form Layout

		sccdel	(1/4)	
Deletion of SCC Link				(Delete Only)
1.unit_name:_____	2.unit_number:____	3.pcsdid:_		
4.mdct_name:_____		5.option_name:_____		
6.duplex:_		7.status:_____		
controller				
8.cntl_unit_name:_____		9.cntl_unit_number:____		

		sccdel	(2/4)	
10.logdevs	logical_name	partition	channel	
1)	_____	—	_____	
2)	_____	—	_____	
3)	_____	—	_____	
4)	_____	—	_____	
5)	_____	—	_____	
6)	_____	—	_____	
7)	_____	—	_____	
8)	_____	—	_____	
9)	_____	—	_____	
10)	_____	—	_____	

```
                                sccdel      (3/4)
10.logdevs(cont)  logical_name  partition  channel
11) _____  —           —
12) _____  —           —
13) _____  —           —
14) _____  —           —
15) _____  —           —
16) _____  —           —
17) _____  —           —
18) _____  —           —
19) _____  —           —
20) _____  —           —
```

```
                                sccdel      (4/4)
10.logdevs(cont)  logical_name  partition  channel
21) _____  —           —
22) _____  —           —
23) _____  —           —
24) _____  —           —
25) _____  —           —
26) _____  —           —
27) _____  —           —
28) _____  —           —
29) _____  —           —
30) _____  —           —
31) _____  —           —
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: SDL

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX



NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcsdid:

Peripheral controller subdevice id. This is the backplane connector position.

4. mdct_name:

The name of the associated *mdct*. This *mdct* may already exist in the database if this request is to duplex the Synchronous Data Link (SDL).

5. option_name:

The name of the associated option record to be added (cpblx3 type for handler = sdl, nphopt for handler = np).

6. duplex:

Is this a duplexed data link?

7. status:

Is this the active or standby unit of a duplexed pair?

8. cntl_unit_name:

The UCB unit name of the controller.

9. cntl_unit_number:

The UCB unit number of the controller.

10. logdevs:

The logdevs (special device files) associated with the drive.

logdevs.logical_name:

The logical device name must begin with */dev/*.

logdevs.partition:

Partition number

logdevs.channel:

Channel number associated with this logdev.

scsiadd

This form adds a Small Computer System Interface (SCSI) Moving Head Disk (MHD), Digital Audio Tape (DAT), or 9-track tape.

Form Layout

	scsiadd	(1/2)
Addition of SCSI Peripherals		(Insert Only)
1.unit_name:___		2.unit_number:___
3.device_id:___		4.sbus_number:___
5.spu_location:___		6.type_drive:___
7.duplex:___		8.duplex_mhd_num:___
9.disk_option:___		10.scsd_cntrl_num:___
11.sc_list.sc_entry:		13.sd_list.sd_entry:
1)___		rqip 1)___
2)___		oos 2)___

	scsiadd	(2/2)
15.vtoc_size:___		

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: MHD, MT

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 31

3. device_id: Required

The id number of the growth unit being added to the SCSI Bus (SBUS).

Legal values: 0 – 6

4. sbus_number: Required

The unit number of the SBUS. This number identifies a particular SCSI Bus under a DFC. SBUS 0 and 2 are controlled by DFC 0. SBUS 1 and 3 are controlled by DFC 1.

Legal values: 0 – 3

5. spu_location:

The physical location of the SCSI Peripheral Unit (SPU). Each SPU location value equates to a unique equipment location in the 3B21D computer cabinet. The equipment location is defined using the cabinet aisle, inches from the floor, and inches from the left side.

Legal values: 0 – 27, 54, 56 – 59, null

6. type_drive: Required

The type of the growth SPU. MHD types are differentiated by number of megabytes.

Legal values: 322, 600, 1000, 2000, DAT, 9TRK

7. duplex:

Indicator that the growth MHD is the second member of a duplex pair.

Legal values: y, n, or null

8. duplex_mhd_num:

The UCB unit number of the mate MHD. This field is the number of the MHD that the growth MHD will be duplexed with.

Legal values: 0 – 31, or null

9. disk_option:

The optional application disk layout type. This field will be used by *UNIX*® Real-Time Reliable (RTR) operating system applications to determine the appropriate MHD partition layout for a telephone exchange.

Legal values: 1 – 10, or null

10. scsd_cntrl_num:

The UCB unit number of the Scanner and Signal Distributor (SCSD) controller.

Legal values: 0 – 255, or null

11. sc_list.sc_entry:

The growth MHD or DAT will be monitored by these scan points.

Legal values: 0 – 47, HEX, or null

13. sd_list.sd_entry:

The growth MHD or DAT will be monitored by these distributor points.

Legal values: 0 – 31, HEX, or null

15. vtoc_size:

The maximum VTOC size to be used for the MHD.

Legal values: 322, 600, 1GB, 2GB, or null

Form Checks

```
if unit_name is "MT"
  then duplex must_be_in {"n",unknown}
  error: "When unit_name is 'MT' then duplex may not be 'y'";

if duplex is "y" and unit_name is "MHD"
  then duplex_mhd_num must_be known
  error: "When duplex is 'y' then duplex_mhd_num must be entered";

if type_drive is_in {"322","600","1000","2000","DAT"}
then do;

  scsd_cntrl_num must_be known
  error: "The scsd_cntrl_num must be entered for MHD or DAT
        devices";

  count { select sc_entry from sc_list } must_equal 2
  error: "both entries of sc_list.sc_entry must be entered for
        MHD or DAT devices";

  count { select sd_entry from sd_list } must_equal 2
  error: "both entries of sd_list.sd_entry must be entered for
        MHD or DAT devices";

end;

if type_drive is_in {"1000","2000","DAT","9TRK"}
  then spu_loc must_be known
  error: "The spu_loc must be entered for this type_drive";

if type_drive is "9TRK"
  then spu_loc must_be_in {56,57,58,59}
  error: "The spu_loc must be 56, 57, 58 or 59 for type_drive
        9TRK";

if sbus_number is_in {0,1} and device_id is 6
  then unit_name must_be "MT"
  error: "Only a tape unit must be defined as device 6 on SBUS 0
        or 1";

if type_drive is "2000"
  then vtoc_size must_be known
  error: "vtoc_size must be entered for this type_drive";

if type_drive is_in {"322","600","1000","DAT","9TRK"}
  then vtoc_size must_be unknown
  error: "vtoc_size must be null for this type_drive";
```

sdladd

This form adds a Synchronous Data Link (SDL).

Form Layout

		sdladd	(1/5)
Addition of Synchronous Data Link		(Insert Only)	
1.unit_name:_____	2.unit_number:____	3.pcsdid:_____	
4.mdct_name:_____	5.handler:_____	6.option_name:_____	
7.dl_type:_____	8.duplex:_____	9.status:_____	
controller			
10.cntl_unit_name:_____		11.cntl_unit_number:_____	
previous logical unit			
12.plu_unit_name:_____		13.plu_unit_number:_____	

		sdladd	(2/5)
14.logdevs	logical_name	partition	channel
1)	_____	—	—
2)	_____	—	—
3)	_____	—	—
4)	_____	—	—
5)	_____	—	—
6)	_____	—	—
7)	_____	—	—
8)	_____	—	—
9)	_____	—	—
10)	_____	—	—

```
                                sdladd      (3/5)
14.logdevs(cont)  logical_name      partition  channel
11) _____          —           —
12) _____          —           —
13) _____          —           —
14) _____          —           —
15) _____          —           —
16) _____          —           —
17) _____          —           —
18) _____          —           —
19) _____          —           —
20) _____          —           —
```

```
                                sdladd      (4/5)
14.logdevs(cont)  logical_name      partition  channel
21) _____          —           —
22) _____          —           —
23) _____          —           —
24) _____          —           —
25) _____          —           —
26) _____          —           —
27) _____          —           —
28) _____          —           —
29) _____          —           —
30) _____          —           —
31) _____          —           —
```

```

                                sdladd      (5/5)
Network Protocol Options
27.lana:_____ 28.runa:_____ 29.loc_name:_____

30.loc_addr  loc1   loc2   loc3   loc4   loc5   loc6   loc7   loc8
             —    —    —    —    —    —    —    —
             loc9  loc10  loc11  loc12  loc13  loc14  loc15  loc16
             —    —    —    —    —    —    —    —
47.rmt_name:_____

48.rmt_addr  rmt1   rmt2   rmt3   rmt4   rmt5   rmt6   rmt7   rmt8
             —    —    —    —    —    —    —    —
             rmt9  rmt10  rmt11  rmt12  rmt13  rmt14  rmt15  rmt16
             —    —    —    —    —    —    —    —
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: SDL

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcsdid: Required

Peripheral controller subdevice id. This is the backplane connector position.

Legal values: 0 – 3

4. mdct_name:

The name of the associated *mdct*. This *mdct* may already exist in the database if this request is to duplex the SDL.

Legal values: printable, null

5. handler: Required

This is the name of the Input/Output (I/O) handler for this data link.

Legal values: sdl, np

6. option_name: Required

The name of the associated option record to be added (cpblx3 type for handler = sdl, nphopt for handler = np).

Legal values: alphanumeric

7. dl_type:

The data link type. This is used to set special options via the default files.

Legal values: AMARC, AMAT4800, AMAT9600, null

8. duplex: Required

Is this a duplexed data link?

Legal values: y, n

Default value: n

9. status:

Is this the active or standby unit of a duplexed pair?

Legal values: active, standby, null

10. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: SDLC

11. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

12. plu_unit_name:

The UCB unit name of the previous logical unit in the ucb linkage chain.

Legal values: alphanumeric, null

Default value: set to the last UCB in the chain

13. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last UCB in the chain

14. logdevs:

The logdevs (special device files) associated with the driver. These must not already exist in the database.

logdevs.logical_name:

The logical device name must begin with */dev/* or */dgn/*.

Legal values: alphanumeric, null

logdevs.partition:

Partition number

Legal values: 1 – 31, null

logdevs.channel:

Channel number associated with this logdev.

Legal values: 1 – 4095, null

27. lana:

The local area network address, the data link's phone number.

Legal values: decimal digits, null

28. runa:

The name of the associated *runa* record, the remote user network address. This may exist in the database.

Legal values: decimal digits, null

29. loc_name:

The name of the associated *bxslloc* record. This cannot exist in the database.

Legal values: printable characters, null

30. loc_addr.loc1 — loc_addr.loc16:

Session address. This is a structure of 16 elements that specifies the session address. An entry in the first field is required. Blanks in the middle of the address translate to 0.

Legal values: hexadecimal digits, null

47. rmt_name:

The name of the associated *bxslrmt* record. This cannot exist in the database.

Legal values: printable characters, null

48. rmt_addr.rmt1 — rmt_addr.rmt16:

Session address. This is a structure of 16 elements that specifies the session address. An entry in the first field is required. Blanks in the middle of the address translate to 0.

Legal values: hexadecimal digits, null

List Element Checks:

```
on logdevs element entry:
  if logical_name is known then partition must_be_between 1 and 31;
  error: "partition must be between 1 and 31 when
  logical_name is known";
```

Form Checks:

```
if handler is "sdl"

then do;
  mdct_name must_not_be unknown
  error: "mdct_name must be known when handler is 'sdl'";

  if duplex is "y"
  then status must_be known
  error: "if duplex is y status must not be NULL";

  lana must_be unknown
  error: "lana must be NULL for handler = sdl";

  loc_name must_be unknown
  error: "loc_name must be NULL for handler = sdl";

  loc_addr.loc1 must_be unknown
  error: "loc_addr.loc1 must be NULL for handler = sdl";

  loc_addr.loc2 must_be unknown
  error: "loc_addr.loc2 must be NULL for handler = sdl";

  loc_addr.loc3 must_be unknown
  error: "loc_addr.loc3 must be NULL for handler = sdl";

  loc_addr.loc4 must_be unknown
  error: "loc_addr.loc4 must be NULL for handler = sdl";

  loc_addr.loc5 must_be unknown
  error: "loc_addr.loc5 must be NULL for handler = sdl";

  loc_addr.loc6 must_be unknown
  error: "loc_addr.loc6 must be NULL for handler = sdl";

  loc_addr.loc7 must_be unknown
  error: "loc_addr.loc7 must be NULL for handler = sdl";

  loc_addr.loc8 must_be unknown
  error: "loc_addr.loc8 must be NULL for handler = sdl";

  loc_addr.loc9 must_be unknown
  error: "loc_addr.loc9 must be NULL for handler = sdl";
```

```
loc_addr.loc10 must_be unknown
    error: "loc_addr.loc10 must be NULL for handler = sdl";

loc_addr.loc11 must_be unknown
    error: "loc_addr.loc11 must be NULL for handler = sdl";

loc_addr.loc12 must_be unknown
    error: "loc_addr.loc12 must be NULL for handler = sdl";

loc_addr.loc13 must_be unknown
    error: "loc_addr.loc13 must be NULL for handler = sdl";

loc_addr.loc14 must_be unknown
    error: "loc_addr.loc14 must be NULL for handler = sdl";

loc_addr.loc15 must_be unknown
    error: "loc_addr.loc15 must be NULL for handler = sdl";

loc_addr.loc16 must_be unknown
    error: "loc_addr.loc16 must be NULL for handler = sdl";

runa must_be unknown
    error: "for handler = sdl, runa must be NULL";

rmt_name must_be unknown
    error: "for handler = sdl, rmt_name must be NULL";

rmt_addr.rmt1 must_be unknown
    error: "rmt_addr.rmt1 must be NULL for handler = sdl";

rmt_addr.rmt2 must_be unknown
    error: "rmt_addr.rmt2 must be NULL for handler = sdl";

rmt_addr.rmt3 must_be unknown
    error: "rmt_addr.rmt3 must be NULL for handler = sdl";

rmt_addr.rmt4 must_be unknown
    error: "rmt_addr.rmt4 must be NULL for handler = sdl";

rmt_addr.rmt5 must_be unknown
    error: "rmt_addr.rmt5 must be NULL for handler = sdl";

rmt_addr.rmt6 must_be unknown
    error: "rmt_addr.rmt6 must be NULL for handler = sdl";
```

```
rmt_addr.rmt7 must_be unknown
  error: "rmt_addr.rmt7 must be NULL for handler = sdl";

rmt_addr.rmt8 must_be unknown
  error: "rmt_addr.rmt8 must be NULL for handler = sdl";

rmt_addr.rmt9 must_be unknown
  error: "rmt_addr.rmt9 must be NULL for handler = sdl";

rmt_addr.rmt10 must_be unknown
  error: "rmt_addr.rmt10 must be NULL for handler = sdl";

rmt_addr.rmt11 must_be unknown
  error: "rmt_addr.rmt11 must be NULL for handler = sdl";

rmt_addr.rmt12 must_be unknown
  error: "rmt_addr.rmt12 must be NULL for handler = sdl";

rmt_addr.rmt13 must_be unknown
  error: "rmt_addr.rmt13 must be NULL for handler = sdl";

rmt_addr.rmt14 must_be unknown
  error: "rmt_addr.rmt14 must be NULL for handler = sdl";

rmt_addr.rmt15 must_be unknown
  error: "rmt_addr.rmt15 must be NULL for handler = sdl";

rmt_addr.rmt16 must_be unknown
  error: "rmt_addr.rmt16 must be NULL for handler = sdl";

end;

if handler is "np"
then do;
  if loc_name is known
  then loc_addr.loc1 must_be known
    error: "for handler = np and loc_name known,
      loc_addr.loc1 must be known";

  if rmt_name is known
  then rmt_addr.rmt1 must_be known
    error: "for handler = np and rmt_name known,
      rmt_addr.rmt1 must be known";

  lana must_be known
  error: "if handler is np lana must not be NULL";
```

```
loc_name must_be known
error: "if handler is np loc_name must not be NULL";

mdct_name must_be unknown
error: "if handler is np mdct_name must be NULL";

dl_type must_be unknown
error: "if handler is np dl_type must be NULL";

/* logdev's must be unknown */

{select logical_name from logdevs}
  must_be null
  error: "when handler = np, logical_name must be NULL";
{select partition from logdevs}
  must_be null
  error: "when handler = np, partition must be NULL";
{select channel from logdevs}
  must_be null
  error: "when handler = np, channel must be NULL";
end;
```

sdlcadd

This form adds a Synchronous Data Link Controller (SDLC).

Form Layout

```

                                     sdlcadd      (1/1)
Addition of Synchronous Data Link Controller      (Insert Only)

  1.unit_name:_____                2.unit_number:____
  3.pcid:_                            4.packname:_____
controller
  5.cntl_unit_name:_____            6.cntl_unit_number____
previous logical unit
  7.plu_unit_name:_____              8.plu_unit_number:____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: SDLC

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcid: Required

Peripheral controller id. This is the Input/Output Processor (IOP) slot position.

Legal values: 0 – 15

4. packname: Required

The circuit packname code.

Legal values: tn75, tn82, un582, or tn1839

5. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: IOP

6. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

7. plu_unit_name:

The UCB unit name of the previous logical unit in the UCB linkage chain.

Legal values: alphanumeric, null

Default value: set to the last UCB in the chain

8. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last UCB in the chain

Form Checks:

```
packname must_be_in {"tn75", "tn82", "un582", or "tn1839"}  
error:"packname must be tn75, tn82, un582, or tn1839";
```

sdlc del

This form deletes a Synchronous Data Link Controller (SDLC).

Form Layout

sdlc del		(1/1)
Deletion of Synchronous Data Link Controller		(Delete Only)
1. unit_name: _____	2. unit_number: _____	
3. pcid: _____		
controller		
4. cntl_unit_name: _____	5. cntl_unit_number _____	
previous logical unit		
6. plu_unit_name: _____	7. plu_unit_number: _____	

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: SDLC

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcid:

Peripheral controller id. This is the Input/Output Processor (IOP) slot position.

4. cntl_unit_name:

The UCB unit name of the controller.

5. cntl_unit_number:

The UCB unit number of the controller.

6. plu_unit_name:

The UCB unit name of the previous logical unit in the UCB linkage chain.

7. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

sdlldel

This form deletes a Synchronous Data Link (SDL).

Form Layout

```

                                     sdlldel      (1/4)
Deletion of Synchronous Data Link                                     (Delete Only)
  1.unit_name:_____  2.unit_number:_____  3.pcsdid:_____
  4.mdct_name:_____  5.handler:_____  6.option_name:_____
  7.duplex:_____  8.status:_____
controller
  9.cntl_unit_name:_____  10.cntl_unit_number:_____
previous logical unit
  11.plu_unit_name:_____  12.plu_unit_number:_____
```

```

                                     sdlldel      (2/4)
13.logdevs      logical_name      partition      channel
  1) _____      _____      _____
  2) _____      _____      _____
  3) _____      _____      _____
  4) _____      _____      _____
  5) _____      _____      _____
  6) _____      _____      _____
  7) _____      _____      _____
  8) _____      _____      _____
  9) _____      _____      _____
 10) _____      _____      _____
```

```
                                sdl1del      (3/4)
13.logdevs(cont)  logical_name      partition  channel
11) _____      —             —
12) _____      —             —
13) _____      —             —
14) _____      —             —
15) _____      —             —
16) _____      —             —
17) _____      —             —
18) _____      —             —
19) _____      —             —
20) _____      —             —
```

```
                                sdl1del      (4/4)
13.logdevs(cont)  logical_name      partition  channel
21) _____      —             —
22) _____      —             —
23) _____      —             —
24) _____      —             —
25) _____      —             —
26) _____      —             —
27) _____      —             —
28) _____      —             —
29) _____      —             —
30) _____      —             —
31) _____      —             —
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: SDL

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX



NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcsdid:

Peripheral controller subdevice id. This is the backplane connector position.

4. mdct_name:

The name of the associated *mdct*. This *mdct* may already exist in the database if this request is to duplex the SDL.

5. handler:

This is the name of the Input/Output (I/O) handler for this data link.

6. option_name:

The name of the associated option record to be added (cpblx3 type for handler = sdl, nphopt for handler = np).

7. duplex:

Is this a duplexed data link?

8. status:

Is this the active or standby unit of a duplexed pair?

9. cntl_unit_name:

The UCB unit name of the controller.

10. cntl_unit_number:

The UCB unit number of the controller.

11. plu_unit_name:

The UCB unit name of the previous logical unit in the UCB linkage chain.

12. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

13. logdevs:

The logdevs (special device files) associated with the drive.

logdevs.logical_name:

The logical device name must begin with */dev/*.

logdevs.partition:

Partition number

logdevs.channel:

Channel number associated with this logdev.

ttyadd

This form adds a Teletypewriter (TTY).

Form Layout

```

                                     ttyadd      (1/1)
Addition of Teletypewriter           (Insert Only)

  1.unit_name:_____  2.unit_number:____  3.pcsdid:_
  4.mdct_name:_____  5.access_____  6.terminal_type:_____
  7.color:_  8.dap:_  9.speed_____  10.ciopt_name:_____
  11.ttopt_name:_____  12.cdopt_name:_____  13.eaiopt_name:_____

controller
  14.cntl_unit_name:_____  15.cntl_unit_number_____

previous logical unit
  16.plu_unit_name:_____  17.plu_unit_number:_____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: TTY

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcsdid: Required

Peripheral controller subdevice id. This is the backplane connector position.

Legal values: 0 – 3

4. mdct_name: Required

The name of the associated *mdct* to be created.

Legal values: printable

⇒ NOTE:

The last character of the *mdct_name* is used as the value of field 4 (*mdct_id_char*) on the *mdct* form. If an On-line Data Integrity (ODIN) form navigation character is used as the last character, the *mdct_id_char* will be translated as shown in the following table. All other characters used will be left as is, with the exception of the single quote (') and the backslash (/), which are untranslatable and thus will continue to be invalid. Refer to "Database Forms Navigation" in Chapter 4 for a list of ODIN form navigation characters.

TRANSLATION OF ODIN FORM NAVIGATION CHARACTERS			
Navigation Character	Translated Value	Navigation Character	Translated Value
^	z	;	p
<	x	equal(=)	o
>	y	%	n
!	w	\$	m
~	v	#	l
[u	}	k
]	t	@	j
?	s	{	i
:	r	*	c
'	q	period(.)	b

5. access:

Hardware access type to the terminal.

Legal values: acu, dial, direct, null

6. terminal_type: Required

Software use of the terminal.

Legal values: craft, unix, dsmlnk

7. color: Required

Is this a color terminal?

Legal values: y, n

Default value: n

8. dap: Required

Is this a control display terminal?

Legal values: y, n

Default value: n

9. speed: Required

Baud rate.

Legal values: 110, 300, 1200, 1800, 2400, 4800, 9600, 19200, 56000

10. ciopt_name: Required

The name of the associated *ciopt* record to be created or shared.

Legal values: alphanumeric

11. ttopt_name:

The name of the associated *ttopt* record to be created or shared.

Legal values: alphanumeric, null

12. cdopt_name:

The name of the associated *cdopt* record to be created or shared.

Legal values: alphanumeric, null

13. eaiopt_name:

The name of the associated *eaiopt* record to be created or shared.

Legal values: alphanumeric, null

⇒ NOTE:

The *eaiopt* forms are not required for terminal_types of dsmlnk; therefore, the eaiopt_name field can be null in this case.

14. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: TTYC

15. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

16. plu_unit_name:

The UCB unit name of the previous logical unit in the UCB linkage chain.

Legal values: alphanumeric, null

Default value: set to the last UCB in the chain

17. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last UCB in the chain

Form Checks:

```
case (terminal_type);

  when {"unix"}: do;
    dap must_be "n"
    error: "for terminal_type of unix, dap must be n.";
  end;

  when {"dsmlnk"}: do;
    dap must_be "n"
    error: "for terminal_type of dsmlnk, dap must be n.";
    access must_be_in {unknown, "direct"}
    error: "for terminal_type of dsmlnk, access must be direct
    or null.";
  end;
end;

if dap is "y"
  then terminal_type must_be "craft"
  error: "for dap = y terminal_type must be craft.";
```

ttycadd

This form adds a Teletypewriter Controller (TTYC).

Form Layout

```
ttycadd      (1/1)
Addition of Teletypewriter Controller      (Insert Only)
  1.unit_name:_____      2.unit_number:____
  3.pcid:____      4.packname:_____
controller
  5.cntl_unit_name:_____      6.cntl_unit_number____
previous logical unit
  7.plu_unit_name:_____      8.plu_unit_number:_____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: TTYC

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

3. pcid: Required

Peripheral controller id. This is the Input/Output Processor (IOP) slot position.

Legal values: 0 – 15

4. packname: Required

The circuit packname code.

Legal values: tn74, un582

Default value: tn74

5. cntl_unit_name: Required

The UCB unit name of the controller.

Legal values: alphanumeric

Default value: IOP

6. cntl_unit_number: Required

The UCB unit number of the controller.

Legal values: 0 – 255, HEX

7. plu_unit_name:

The UCB unit name of the previous logical unit in the UCB linkage chain.

Legal values: alphanumeric, null

Default value: set to the last UCB in the chain

8. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

Legal values: 0 – 255, HEX, null

Default value: set to the last UCB in the chain

Form Checks:

```
packname must_be_in {"tn74" or "un582"}  
error:"packname must be tn74 or un582";
```

ttycdel

This form deletes a Teletypewriter Controller (TTYC).

Form Layout

```

                                     ttycdel      (1/1)
Deletion of Teletypewriter Controller                                     (Delete Only)
  1.unit_name:_____          2.unit_number:_____
  3.pcid:_____
controller
  4.cntl_unit_name:_____      5.cntl_unit_number_____
previous logical unit
  6.plu_unit_name:_____        7.plu_unit_number:_____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: TTYC

2. unit_number: Required

The ucb unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcid:

Peripheral controller id. This is the Input/Output Processor (IOP) slot position.

4. cntl_unit_name:

The UCB unit name of the controller.

5. cntl_unit_number:

The UCB unit number of the controller.

6. plu_unit_name:

The UCB unit name of the previous logical unit in the UCB linkage chain.

7. plu_unit_number:

The UCB unit number of the previous logical unit in the UCB linkage chain.

ttydel

This form deletes a Teletypewriter (TTY).

Form Layout

```

                                     ttydel      (1/1)
Deletion of Teletypewriter                (Delete Only)

1.unit_name:_____  2.unit_number:____  3.pcsdid:_
4.mdct_name:_____  5.access_____  6.speed____  7.ciopt_name:_____
8.ttopt_name:_____  9.cdopt_name:_____  10.eaiopt_name:_____
controller  11.cntl_unit_name:_____  12.cntl_unit_number____
13. logdevs  logical_name  partition
    1) _____  _____
    2) _____  _____
    3) _____  _____
    4) _____  _____
```

Definition of Fields:

1. unit_name: Required

The UCB unit name. This is a key field.

Legal values: alphanumeric

Default value: TTY

2. unit_number: Required

The UCB unit number. This is a key field.

Legal values: 0 – 255, HEX

⇒ NOTE:

The following fields on the form are read-only and are supplied by the high-level process.

3. pcsdid: Required

Peripheral controller subdevice id. This is the backplane connector position.

4. mdct_name: Required

The name of the associated *mdct* to be created.

5. access:

Hardware access type to the terminal.

6. speed: Required

Baud rate

7. ciopt_name: Required

The name of the associated *ciopt* record.

8. ttopt_name:

The name of the associated *ttopt* record.

9. cdopt_name:

The name of the associated *cdopt* record.

10. eaiopt_name:

The name of the associated *eaiopt* record.

11. cntl_unit_name: Required

The UCB unit name of the controller.

12. cntl_unit_number: Required

The UCB unit number of the controller.

13. logdevs:

The logdevs (special device files) associated with the terminal.

logdevs.logical_name:

The logical device name must begin with */dev/*.

logdevs.partition:

Partition number

vfydflt

This form verifies the data specified in a default file.

Form Layout

Default File Verification	vfydflt	(1/1)	(Execute Only)
1.error_file:	_____		
2.default_file:	_____		
3.default_dir:	_____		
4.mask_dir:	_____		

Definition of Fields:

1. error_file: Required

This is the name of the file that is created for error messages. This is a key field.

Legal values: printable characters

2. default_file: Required

Name of the file containing the default values.

Legal values: printable characters

3. default_dir:

Pathname of the directory containing the default_file.

Legal values: printable, null

Default value: null

⇒ NOTE:

If null, the directory *//la/defaults* is used.

4. mask_dir:

Pathname of the directory containing the masks.

Legal values: printable, null

Default value: null

⇒ NOTE:

If null, official directory */lla/ecdmasks* is used.

RC/V Error Code Explanations

8

Contents

Low-Level Access Error Codes

[8-1](#)

RC/V Error Code Explanations

8

Low-Level Access Error Codes

The following is a list of Recent Change and Verify (RC/V)/Low-Level Access (LLA) fatal error codes:

A_APHO00 — cannot get aphopt record — (ecd/rcv/integ)

A_APHO01 — cannot open pphuser set — (ecd/rcv/integ)

A_APHO02 — error while sequencing thru pphuser records — (ecd/rcv/integ)

A_APHO03 — APHOPT and PPHUSER must not have same loc_ssap_sel
APHOPT.aphoptname is %s and PPHUSER.attach_id is %d. — (ecd/rcv/integ)

A_APHU00 — cannot get aphuser record — (ecd/rcv/integ)

A_APHU01 — cannot open tsapinfo set — (ecd/rcv/integ)

A_APHU02 — error sequencing thru tsapinfo records — (ecd/rcv/integ)

A_AUD00 — cannot get audrec record — (ecd/rcv/integ)

AB01 — unable to use target — (lla/am)

AB02 — index overflow — (lla/am)

AB03 — unable to unuse target — (lla/am)

A_BPARM00 — cannot get record info — (ecd/rcv/integ)

A_BPARM01 — cannot read btparm record — (ecd/rcv/integ)

A_BPARM02 — cannot read ecdorg record — (ecd/rcv/integ)

AC00 — cannot close incore database — (ecd/rcv/ftamaux)

AC01 — cannot attach boot ecd — (ecd/rcv/ftamaux)

AC02 — cannot close boot ecd — (ecd/rcv/ftamaux)

AC03 — cannot attach incore ecd — (ecd/rcv/ftamaux)

AC04 — cannot close boot ecd — (ecd/rcv/ftamaux)

AC05 — cannot attach incore ecd — (ecd/rcv/ftamaux)

A_CIOPT00 — cannot get ciopt record — (ecd/rcv/integ)

A_CIOPT01 — unable to open the database — (ecd/rcv/integ)

A_CIOPT02 — cannot get the ucb controller record — (ecd/rcv/integ)

A_CIOPT03 — error sequencing through ucb records — (ecd/rcv/integ)

A_CIOPT04 — unable to retrieve ucb record for ciopt — (ecd/rcv/integ)

A_CIOPT05 — only one TTY can be controlled by (u583) MTTYC controller —
(ecd/rcv/integ)

A_CIOPT06 — unable to get the rid for TTY down ptr — (ecd/rcv/integ)

A_CIOPT07 — unable to retrieve ucb for side search — (ecd/rcv/integ)

A_CIOPT08 — invalid PCSDID on ucb record — (ecd/rcv/integ)

A_CLS000 — can not get classdef record — (ecd/rcv/integ)

ACT00 — activate can only be run when attached to incore
database — (ecd/rcv/ftamecd)

ACTECD00 — cannot access ecdorg record — (ecd/rcv/integ)

ACTECD01 — cannot update ecdorg rids — (ecd/rcv/integ)

ACTMDCT00 — cannot access mdct record — (ecd/rcv/integ)

ACTMDCT01 — cannot update mdct rids — (ecd/rcv/integ)

ACTUCB00 — cannot access ucb record — (ecd/rcv/integ)

ACTUCB01 — cannot update ucb rid — (ecd/rcv/integ)

AD00 — cannot attach ecd database — (ecd/rcv/ftamaux)

AD00 — cannot attach sg database — (ecd/rcv/ftamaux)

AD01 — ecd tr_setup error — (ecd/rcv/ftamaux)

AD01 — sg tr_setup error — (ecd/rcv/ftamaux)

ADB01 — cannot malloc datadir structure — (lla/audit)

ADB01 — cannot malloc datadir structure — (lla/dml)

ADB02 — sdp_house error — (lla/audit)

ADB02 — sdp_house error — (lla/dml)

ADB03 — cannot connect database — (lla/audit)

ADB03 — cannot connect database — (lla/dml)

ADB04 — cannot load dmlhead — (lla/audit)

ADB04 — cannot load dmlhead — (lla/dml)

ADB05 — currphypage must be multiple of creation size — (lla/audit)

ADB05 — currphypage must be multiple of creation size — (lla/dml)

ADB06 — bad dmlhead structure — (lla/audit)

ADB06 — bad dmlhead structure — (lla/dml)

ADB07 — your database not compatible with this version — (lla/audit)

ADB07 — your database not compatible with this version — (lla/dml)

A_EAIOPT00 — ucb record for timezone %s does not exist — (ecd/rcv/integ)

A_EAIOPT01 — dst record is not specified in ucb record %s 0 — (ecd/rcv/integ)

A_EAIOPT02 — record specified in option field in ucb record is not dst record —
(ecd/rcv/integ)

AF00 — bad PLACE sanity id — (lla/audit)

A_FTMO00 — cannot get ftamopt record — (ecd/rcv/integ)

A_FTMO01 — cannot open gfsappl set — (ecd/rcv/integ)
A_FTMO02 — error sequencing thru gfsappl records — (ecd/rcv/integ)
A_FTMO03 — cannot get gfscom record — (ecd/rcv/integ)
A_FTMO00 — cannot get ftamuser record — (ecd/rcv/integ)
A_FTMO01 — cannot open tsapinfo set — (ecd/rcv/integ)
A_FTMO02 — error sequencing thru tsapinfo records — (ecd/rcv/integ)
AFTO01 — cannot create AFTOPT record — (ecd/rcv/ftamecd)
AFTO02 — cannot delete AFTOPT record — (ecd/rcv/ftamecd)
AFTO03 — cannot read AFTOPT record — (ecd/rcv/ftamecd)
AFTO04 — cannot update AFTOPT record — (ecd/rcv/ftamecd)
A_GFSA00 — cannot get gfsappl record — (ecd/rcv/integ)
A_GFSA01 — cannot open cursor on gfsappl set — (ecd/rcv/integ)
A_GFSA02 — cannot get gfsappl record — (ecd/rcv/integ)
A_GFSA03 — cannot get gfsinit record — (ecd/rcv/integ)
A_GFSA05 — cannot get gfsresp record — (ecd/rcv/integ)
A_GFSA06 — cannot get gfscom record — (ecd/rcv/integ)
A_GFSA07 — cannot get gfssec record — (ecd/rcv/integ)
A_GFSC00 — cannot get gfscom record — (ecd/rcv/integ)
A_GFSC01 — cannot open cursor on gfsappl set — (ecd/rcv/integ)
A_GFSC02 — cannot get gfsappl record — (ecd/rcv/integ)
A_GFSC03 — cannot get gfsresp record — (ecd/rcv/integ)
A_GFSC04 — cannot get gfssec record — (ecd/rcv/integ)
A_GFSR00 — cannot get gfsresp record — (ecd/rcv/integ)
A_GFSR01 — cannot open cursor on gfsappl set — (ecd/rcv/integ)
A_GFSR02 — cannot get gfsappl record — (ecd/rcv/integ)

A_GFSR03 — cannot get gfscom record — (ecd/rcv/integ)
A_GFSR04 — cannot get gfssec record — (ecd/rcv/integ)
A_GFSS00 — cannot get gfssec record — (ecd/rcv/integ)
A_GFSS01 — cannot open cursor on gfsappl set — (ecd/rcv/integ)
A_GFSS02 — cannot get gfsappl record — (ecd/rcv/integ)
A_GFSS03 — cannot get gfscom record — (ecd/rcv/integ)
A_GFSS04 — cannot get gfsresp record — (ecd/rcv/integ)
AI00 — cannot attach incore database — (ecd/rcv/ftamaux)
AI00 — unable to malloc PLACE structure — (lla/audit)
AI01 — cannot close incore database — (ecd/rcv/ftamaux)
AI01 — unable to malloc control table — (lla/audit)
AI02 — bad audname specified in cntrl_tab — (lla/audit)
AI02 — sysgen illegal for incore db — (ecd/rcv/ftamaux)
AI03 — cannot attach auditecd database — (ecd/rcv/ftamaux)
AI04 — cannot close auditecd database — (ecd/rcv/ftamaux)
AINST00 — cannot retrieve audinst record — (ecd/rcv/ftamecd)
AINST01 — cannot update audinst record — (ecd/rcv/ftamecd)
AINST02 — cannot create audinst record — (ecd/rcv/ftamecd)
AINST03 — could not read audrec record — (ecd/rcv/ftamecd)
AINST04 — could not update audrec record — (ecd/rcv/ftamecd)
AINST05 — could not delete audinst record — (ecd/rcv/ftamecd)
AL01 — space not connected %o — (lla/sdp)
AL02 — no permissions to allot — (lla/sdp)
AL03 — fail to pop stack %o — (lla/sdp)
AL04 — address space overflow %o — (lla/sdp)

AL05 — fail to redeem fragment from multipage item %o — (lla/sdp)

AL06 — fail to redeem fragment from old current page %o — (lla/sdp)

ALAU — audit failed in sdp_allot — (lla/sdp)

A_MDCT00 — cannot get mdct record — (ecd/rcv/integ)

A_MDCT01 — cannot get ucb record — (ecd/rcv/integ)

A_MDCT02 — cannot open set of ucb's — (ecd/rcv/integ)

A_MDCT03 — error sequencing thru set of ucb's — (ecd/rcv/integ)

A_PACK00 — cannot get record info — (ecd/rcv/integ)

A_PACK01 — record not a pack record — (ecd/rcv/integ)

A_PACK02 — cannot retrieve pack record — (ecd/rcv/integ)

A_PACK03 — error getting rid from pack set — (ecd/rcv/integ)

A_PACK04 — rid not found in pack set — (ecd/rcv/integ)

A_PACK05 — rid in pack set doesn't match given rid — (ecd/rcv/integ)

A_PACK06 — cannot open partition set — (ecd/rcv/integ)

A_PACK07 — error while sequencing thru partition records — (ecd/rcv/integ)

A_PART00 — cannot get record info — (ecd/rcv/integ)

A_PART01 — record not a partition record — (ecd/rcv/integ)

A_PART02 — cannot retrieve partition record — (ecd/rcv/integ)

A_PART03 — partition record not in exactly 1 set — (ecd/rcv/integ)

A_PART04 — cannot get setid for partition record — (ecd/rcv/integ)

A_PART05 — cannot open set of all pack records — (ecd/rcv/integ)

A_PART06 — error while sequencing thru pack records — (ecd/rcv/integ)

A_PART07 — disconnected partition record — (ecd/rcv/integ)

A_PART08 — cannot open partition set — (ecd/rcv/integ)

A_PART09 — error while sequencing thru partition records — (ecd/rcv/integ)

- APHO01 — cannot create APHOPT record — (ecd/rcv/ftamecd)
- APHO02 — cannot delete APHOPT record — (ecd/rcv/ftamecd)
- APHO03 — cannot read APHOPT record — (ecd/rcv/ftamecd)
- APHO04 — cannot update APHOPT record — (ecd/rcv/ftamecd)
- APHO05 — cannot open set of APHOPT records — (ecd/rcv/ftamecd)
- APHO06 — cannot get record from APHOPT set — (ecd/rcv/ftamecd)
- APHO07 — cannot open set of APHUSER records — (ecd/rcv/ftamecd)
- APHO08 — cannot get record from APHUSER set — (ecd/rcv/ftamecd)
- APHO09 — sphopt record must be inserted before inserting aphopt — (ecd/rcv/ftamecd)
- APHU00 — cannot open set of APHUSER records — (ecd/rcv/ftamecd)
- APHU01 — cannot create APHUSER record — (ecd/rcv/ftamecd)
- APHU02 — cannot delete APHUSER record — (ecd/rcv/ftamecd)
- APHU03 — cannot read APHUSER record — (ecd/rcv/ftamecd)
- APHU04 — cannot update APHUSER record — (ecd/rcv/ftamecd)
- APHU05 — aphopt specified on APHUSER form does not exist — (ecd/rcv/ftamecd)
- APHU06 — cannot read APHOPT record — (ecd/rcv/ftamecd)
- APHU07 — cannot get record from APHUSER set — (ecd/rcv/ftamecd)
- APHU08 — cannot delink the old APHUSER record — (ecd/rcv/ftamecd)
- A_PPHU00 — cannot get pphuser record — (ecd/rcv/integ)
- A_PPHU01 — cannot open aphopt set — (ecd/rcv/integ)
- A_PPHU02 — error while sequencing thru aphopt records — (ecd/rcv/integ)
- A_PPHU03 — PPHUSER and APHOPT must not have same loc_ssap_sel
PPHUSER.attach_id is %d and APHOPT.aphoptname is %s. — (ecd/rcv/integ)
- AR00 — bad audname specified — (lla/audit)
- AR01 — bad PLACE sanity id — (lla/audit)

- AR01 — cannot load record — (lla/dml)
- AR02 — audnum must be > 0 for an enabled audit phase — (lla/audit)
- AR02 — cannot attach new set — (lla/dml)
- AR03 — cannot unload record — (lla/dml)
- AR04 — cannot load large record — (lla/dml)
- AR05 — cannot load set — (lla/dml)
- AR06 — insert controller error — (lla/dml)
- AR07 — cannot unload record — (lla/dml)
- AR08 — cannot unload set — (lla/dml)
- AR09 — d2p error — (lla/dml)
- AR10 — cannot load variable length record — (lla/dml)
- AS01 — cannot change record status — (lla/dml)
- AS02 — cannot change record status — (lla/dml)
- AS03 — cannot allocate for set member block — (lla/dml)
- AS04 — cannot load set member block — (lla/dml)
- AS05 — cannot change record status — (lla/dml)
- AS06 — cannot unload set member block — (lla/dml)
- AS07 — cannot load set member block — (lla/dml)
- AS08 — cannot write set member block — (lla/dml)
- AS09 — set member block inconsistent — (lla/dml)
- AS10 — cannot allocate set member block — (lla/dml)
- AS11 — cannot load new set member block — (lla/dml)
- AS12 — cannot write new set member block — (lla/dml)
- AS13 — cannot unload set member block — (lla/dml)
- AS14 — cannot redeem old set member block — (lla/dml)

AS15 — inconsistent record header — (lla/dml)
AS16 — cannot change record status — (lla/dml)
AT00 — cannot delete delta files — (ecd/rcv/ftamaux)
AT01 — cannot release trigger resources — (ecd/rcv/ftamaux)
ATP01 — cannot allocate rid block — (lla/dml)
ATP02 — cannot load new rid block — (lla/dml)
ATP03 — cannot load last rid block — (lla/dml)
ATP04 — cannot unload last rid block — (lla/dml)
ATP05 — cannot update dml header — (lla/dml)
ATP06 — cannot load current rid block — (lla/dml)
ATP07 — cannot write current rid block — (lla/dml)
ATP08 — cannot write current full rid block — (lla/dml)
ATP09 — cannot change dmlhead — (lla/dml)
ATP10 — no free entry in ridblk with free slots — (lla/dml)
A_TSAP00 — cannot get tsapinfo record — (ecd/rcv/integ)
A_TSAP01 — error sequencing thru tsapinfo records — (ecd/rcv/integ)
A_TSAP02 — error sequencing thru btihdev records — (ecd/rcv/integ)
A_TSAP03 — cannot get sphopt record — (ecd/rcv/integ)
AU00 — whatnext failure — (lla/audit)
AU01 — audit failure in global hash — (lla/sdp)
AU02 — audit failure in space hash — (lla/sdp)
AU03 — audit failure in environment hash — (lla/sdp)
AU04 — audit failure in manager hash — (lla/sdp)
AUAU — audit failed in sdp_audit — (lla/sdp)
A_UCB00 — cannot get ucb record — (ecd/rcv/integ)

A_UCB01 — cannot get ucb up0 record — (ecd/rcv/integ)

A_UCB02 — cannot get ucb up1 record — (ecd/rcv/integ)

A_UCB03 — cannot get ucb side record — (ecd/rcv/integ)

A_UCB04 — cannot get ucb down record — (ecd/rcv/integ)

A_UCB05 — cannot get record info for option record — (ecd/rcv/integ)

AUD00 — could not block ECD audit(s) — (ecd/rcv/ftamaux)

AUD01 — ECD audit running - try later — (ecd/rcv/ftamaux)

AUDERR00 — cannot retrieve audrec record — (ecd/rcv/ftamecd)

AUDERR01 — cannot update audrec record — (ecd/rcv/ftamecd)

AUDERR02 — cannot update associated audrec record — (ecd/rcv/ftamecd)

AUDHIST00 — bad rid stored in symbol table — (ecd/rcv/ftamecd)

AUDINH00 — Check simcntl record for validity — (ecd/rcv/ftamaux)

AUDIT00 — cannot open UNIVERSAL set — (lla/tb)

AUDIT01 — error sequencing through UNIVERSAL set — (lla/tb)

AUDIT02 — error getting record info — (lla/tb)

AUDREC00 — cannot create audrec record — (ecd/rcv/ftamecd)

AUDREC01 — cannot retrieve audrec record — (ecd/rcv/ftamecd)

AUDREC02 — cannot update audrec record — (ecd/rcv/ftamecd)

AUDREC03 — cannot delete audrec record — (ecd/rcv/ftamecd)

AUDREC04 — cannot break old links — (ecd/rcv/ftamecd)

AUDREC05 — cannot update audinst record — (ecd/rcv/ftamecd)

AUDREC06 — cannot delete audrec until all associated audinsts are deleted —
(ecd/rcv/ftamecd)

AUDSTAT00 — bad rid stored in symbol table — (ecd/rcv/ftamecd)

AUDSTAT01 — Audit instance name cannot be retrieved — (ecd/rcv/ftamecd)

AUDSTAT02 — Illegal inhibit state 0x%x — (ecd/rcv/ftamecd)
AUTHR01 — cannot create authdef record — (ecd/rcv/ftamecd)
AUTHR02 — cannot delete authdef record — (ecd/rcv/ftamecd)
AUTHR03 — cannot read authdef record — (ecd/rcv/ftamecd)
AUTHR04 — cannot update authdef record — (ecd/rcv/ftamecd)
AUTHR05 — cannot get authdef record — (ecd/rcv/ftamecd)
A_VCHNUM00 — cannot get ucb record — (ecd/rcv/integ)
A_VCHNUM01 — cannot get ucb up0 record — (ecd/rcv/integ)
A_VDID00 — cannot get ucb record — (ecd/rcv/integ)
A_VDID01 — cannot get ucb up0 record — (ecd/rcv/integ)
A_VDID02 — cannot get ucb down record — (ecd/rcv/integ)
A_VDID03 — cannot get ucb side record — (ecd/rcv/integ)
BAS00 — unable to change write status on page — (lla/porter)
BAS01 — cannot load set member block — (lla/porter)
BAS02 — set member block inconsistent — (lla/porter)
BAS03 — unable to write set member block — (lla/porter)
BAS04 — bad record header set status field — (lla/porter)
BBR00 — pointer to displacement conversion error — (lla/porter)
BBR01 — cannot allot space for record — (lla/porter)
BBR02 — cannot add record ridblk entry — (lla/porter)
BBR03 — cannot allot space for set member block — (lla/porter)
BBR04 — cannot load allocated set member block — (lla/porter)
BBR05 — cannot unload set member block — (lla/porter)
BBR06 — cannot write new record out — (lla/porter)
BCK00 — unable to load current rid block — (lla/audit)

BCK01 — unable to load specified record — (lla/audit)
BCK02 — fatal error doing back link check — (lla/audit)
BCK03 — unable to unload specified record — (lla/audit)
BCK04 — unable to unload current rid block — (lla/audit)
BCK05 — bad sanity id in current rid block — (lla/audit)
BCKAUD00 — record sanity id at 0x%lx found to have incorrect value 0x%lx — (lla/audit)
BCKAUD01 — record at 0x%lx back link being reset from 0x%lx to 0x%lx — (lla/audit)
BCKAUD02 — rid block slot at 0x%lx being reset from 0x%lx to 0x%lx — (lla/audit)
BCKAUD03 — rid block slot 0x%lx bad, being reset from 0x%lx to 0x%lx — (lla/audit)
BDB00 — cannot read database header information — (lla/porter)
BECD01 — unable to create ecd database — (ecd/rcv/createdb)
BECD02 — unable to attach to newly created ecd database — (ecd/rcv/createdb)
BECD03 — unable to create set — (ecd/rcv/createdb)
BECD04 — could not create ecdorg (root) record — (ecd/rcv/createdb)
BECD05 — could not release ecd database — (ecd/rcv/createdb)
BF00 — cannot initialize 'formform' — (ecd/rcv/fiop)
BF01 — cannot calloc ff_table — (ecd/rcv/fiop)
BF01 — error encountered on read of key or form — (ecd/rcv/prelinks)
BF01 — unable to use target — (lla/am)
BF02 — error encountered on read of key — (ecd/rcv/prelinks)
BF02 — %s: 'formform' field %d illegal — (ecd/rcv/fiop)
BF02 — unable to unuse target — (lla/am)
BF03 — batch input error - cross form check — (ecd/rcv/prelinks)
BGI00 — old itemid doesnot match and record or set instances — (lla/porter)
BGI01 — unable to locate old to new mapping record — (lla/porter)

BL00 — unable to load rid block slot page — (Ila/audit)
BL01 — unable to unload rid block slot page — (Ila/audit)
BMN00 — cannot open specified input file — (Ila/porter)
BMN01 — cannot create specified user database — (Ila/porter)
BMN02 — cannot create specified itemid mapping database — (Ila/porter)
BMN03 — cannot attach user database and directory — (Ila/porter)
BMN04 — cannot attach mapping database and directory — (Ila/porter)
BMN05 — cannot create mapping set — (Ila/porter)
BMN06 — failure converting user database — (Ila/porter)
BMN07 — bad user rootid — (Ila/porter)
BMN08 — cannot close user or mapping databases and directories — (Ila/porter)
BMN09 — cannot remove mapdb — (Ila/porter)
BRC00 — cannot read record header — (Ila/porter)
BRC01 — cannot read and encode user record info. — (Ila/porter)
BRC02 — cannot build record — (Ila/porter)
BRC03 — cannot create old to new rid mapping record — (Ila/porter)
BRC04 — record section termination error — (Ila/porter)
BRC05 — bad record entry identifier — (Ila/porter)
BRC06 — unknown record name — (Ila/porter)
BS01 — unable to use target — (Ila/am)
BS02 — unable to allot new id — (Ila/am)
BS03 — unable to use directory — (Ila/am)
BS04 — unable to unuse directory — (Ila/am)
BS05 — unable to use new id — (Ila/am)
BS06 — unable to unuse new id — (Ila/am)

BS07 — unable to unuse target — (lla/am)

BSC00 — cannot read set header — (lla/porter)

BSC01 — cannot retrieve set mapping info — (lla/porter)

BSC02 — mapping entry not for a set — (lla/porter)

BSC03 — cannot get new rid for standard key — (lla/porter)

BSC04 — cannot attach record to set — (lla/porter)

BSC05 — set section termination error — (lla/porter)

BSC06 — bad set entry identifier — (lla/porter)

BSC07 — cannot convert non-record based key — (lla/porter)

BSC08 — cannot get new rid for non-record based key — (lla/porter)

BSC09 — cannot attach record to set — (lla/porter)

BSC10 — error parsing non-record based key info — (lla/porter)

BSG01 — unable to create sg database — (ecd/rcv/createdb)

BSI00 — cannot read set instance section — (lla/porter)

BSI01 — cannot read set instance header — (lla/porter)

BSI02 — cannot read set instance info. — (lla/porter)

BSI03 — set ordinal doesnot correspond to any set names — (lla/porter)

BSI04 — cannot create set instance — (lla/porter)

BSI05 — cannot create set mapping record — (lla/porter)

BTIHD01 — cannot create BTIHDEV record — (ecd/rcv/ftamecd)

BTIHD02 — cannot delete BTIHDEV record — (ecd/rcv/ftamecd)

BTIHD03 — cannot read BTIHDEV record — (ecd/rcv/ftamecd)

BTIHD04 — cannot update BTIHDEV record — (ecd/rcv/ftamecd)

BTP00 — cannot insert BTP record — (ecd/rcv/ftamecd)

BTP01 — cannot read BTP record — (ecd/rcv/ftamecd)

BTP02 — cannot update BTP record — (ecd/rcv/ftamecd)

BUR00 — cannot open cursor through UNIVERSAL set — (lla/porter)

BUR01 — nsqrid sequencing error — (lla/porter)

BUR02 — cannot load entire record — (lla/porter)

BUR03 — displacement to pointer error — (lla/porter)

BUR04 — can update user link and owner fields — (lla/porter)

BUR05 — pointer to displacement error — (lla/porter)

BUR06 — cannot write revised record — (lla/porter)

BV00 — formname not found — (ecd/rcv/links)

BV01 — form instance data file not found — (ecd/rcv/links)

BV02 — form instance data file is locked — (ecd/rcv/links)

BV03 — fail returned from mdatent — (ecd/rcv/links)

BV04 — data base access error — (ecd/rcv/links)

BXINFO01 — cannot create BXINFO record — (ecd/rcv/ftamecd)

BXINFO02 — cannot delete BXINFO record — (ecd/rcv/ftamecd)

BXINFO03 — cannot read BXINFO record — (ecd/rcv/ftamecd)

BXINFO04 — cannot update BXINFO record — (ecd/rcv/ftamecd)

BXLOC01 — cannot create BXLOC record — (ecd/rcv/ftamecd)

BXLOC02 — cannot delete BXLOC record — (ecd/rcv/ftamecd)

BXLOC03 — cannot read BXLOC record — (ecd/rcv/ftamecd)

BXLOC04 — cannot update BXLOC record — (ecd/rcv/ftamecd)

BXRMT01 — cannot create BXRMT record — (ecd/rcv/ftamecd)

BXRMT02 — cannot delete BXRMT record — (ecd/rcv/ftamecd)

BXRMT03 — cannot read BXRMT record — (ecd/rcv/ftamecd)

BXRMT04 — cannot update BXRMT record — (ecd/rcv/ftamecd)

CC01 — error return from create hash — (lla/am)

CC02 — error return from create index — (lla/am)

CC03 — error return from create btree — (lla/am)

CC04 — attempt to create unknown access method — (lla/am)

CD00 — bad argument count — (ecd/rcv/createdb)

CD01 — bad argument specified — (ecd/rcv/createdb)

CD02 — no database specified — (ecd/rcv/createdb)

CD03 — database name too long — (ecd/rcv/createdb)

CDOPT00 — cannot create CDOPT record — (ecd/rcv/ftamecd)

CDOPT01 — cannot delete cdopt from ecd — (ecd/rcv/ftamecd)

CDOPT02 — cannot read cdopt in ecd — (ecd/rcv/ftamecd)

CDOPT03 — cannot update cdopt in ecd — (ecd/rcv/ftamecd)

CFTI01 — cannot create CFTINFO record — (ecd/rcv/ftamecd)

CFTI02 — cannot read CFTINFO record — (ecd/rcv/ftamecd)

CFTI03 — cannot update CFTINFO record — (ecd/rcv/ftamecd)

CH01 — no permission to change %o — (lla/sdp)

CH02 — frame not found %o — (lla/sdp)

CHAU — audit failed in sdp_change — (lla/sdp)

CHC000 — internal error; transaction is not allowed for btihdev — (ecd/rcv/integ)

CHC001 — internal error; transaction is not allowed for tphopt record — (ecd/rcv/integ)

CHC002 — cannot retrieve tphopt record — (ecd/rcv/integ)

CHC003 — invalid TE type in the tphopt record — (ecd/rcv/integ)

CHC004 — internal error; transaction not allowed for aphopt — (ecd/rcv/integ)

CHC005 — internal error; transaction not allowed for sphopt — (ecd/rcv/integ)

CHC006 — internal error; transaction not allowed for tsapinfo — (ecd/rcv/integ)

CHC007 — internal error; transaction not allowed for pphopt — (ecd/rcv/integ)
CHC008 — internal error; transaction not allowed for pphuser — (ecd/rcv/integ)
CHOPT00 — cannot create CHOPT record — (ecd/rcv/ftamecd)
CHOPT01 — cannot delete chopt from ecd — (ecd/rcv/ftamecd)
CHOPT02 — cannot read chopt in ecd — (ecd/rcv/ftamecd)
CHOPT03 — cannot update chopt in ecd — (ecd/rcv/ftamecd)
CI01 — unable to allot index array — (lla/am)
CI02 — unable to use index slot — (lla/am)
CI03 — unable to unuse index slot — (lla/am)
CIOPT00 — cannot create ciopt record — (ecd/rcv/ftamecd)
CIOPT01 — cannot delete ciopt from ecd — (ecd/rcv/ftamecd)
CIOPT02 — cannot read ciopt in ecd — (ecd/rcv/ftamecd)
CIOPT03 — cannot update ciopt in ecd — (ecd/rcv/ftamecd)
CIOPT04 — cannot break old links — (ecd/rcv/ftamecd)
CLS00 — cannot close ecd database — (ecd/rcv/ftamaux)
CLS00 — cannot close sg database — (ecd/rcv/ftamaux)
CLS01 — cannot update ecdroot last reference time — (ecd/rcv/ftamaux)
CLS01 — cannot update sgroot last reference time — (ecd/rcv/ftamaux)
CLSR00 — cannot insert CLSR set — (ecd/rcv/ftamecd)
CLSR01 — cannot create CLSR record — (ecd/rcv/ftamecd)
CLSR02 — cannot delete CLSR record — (ecd/rcv/ftamecd)
CLSR03 — cannot read CLSR record — (ecd/rcv/ftamecd)
CLSR04 — cannot update CLSR record — (ecd/rcv/ftamecd)
CLSR05 — cannot get CLSR record — (ecd/rcv/ftamecd)
CO01 — unknown permissions %d — (lla/sdp)

- CO02 — error return from rdSPACE %s — (lla/sdp)
- CO03 — your address space (%ld) not compatible with this version (%ld) %s — (lla/sdp)
- CO04 — environment not housed %ot%s — (lla/sdp)
- CO05 — framesize not compatible with file size %st%ldt%d — (lla/sdp)
- CO06 — page/frame conflict %st%dt%d — (lla/sdp)
- COAU — audit failed in sdp_connect — (lla/sdp)
- CP00 — cannot load indirect key — (lla/porter)
- CP01 — cannot unload indirect key — (lla/porter)
- CPBLX3.0 — feidl must be >= than most significant non_zero byte of far end system identifier — (ecd/rcv/ftamecd)
- CPBLX3.1 — neidl must be >= than most significant non_zero byte of near end system identifier — (ecd/rcv/ftamecd)
- CPBLX3.2 — pswdl must be >= than most significant non_zero byte of password — (ecd/rcv/ftamecd)
- CPOPT00 — cannot create cpopt record — (ecd/rcv/ftamecd)
- CPOPT01 — cannot delete cpopt from ecd — (ecd/rcv/ftamecd)
- CPOPT02 — cannot read cpopt in ecd — (ecd/rcv/ftamecd)
- CPOPT03 — cannot update cpopt in ecd — (ecd/rcv/ftamecd)
- CPPHNONE.0 — feidl must be >= than most significant non_zero byte of far end system identifier — (ecd/rcv/ftamecd)
- CPPHNONE.1 — neidl must be >= than most significant non_zero byte of near end system identifier — (ecd/rcv/ftamecd)
- CPPHNONE.2 — pswdl must be >= than most significant non_zero byte of password — (ecd/rcv/ftamecd)
- CR01 — record initialization error — (lla/dml)
- CR02 — unable to create record — (lla/dml)
- CR03 — unable to allocate record slot — (lla/dml)
- CR04 — cannot load set — (lla/dml)

- CR05 — cannot install access — (lla/dml)
- CR06 — cannot unload set — (lla/dml)
- CR07 — cannot load record — (lla/dml)
- CR08 — cannot write record — (lla/dml)
- CR09 — dml built with record size (FULLBUFFLEN) < user record — (lla/dml)
- CR10 — p2d error — (lla/dml)
- CR11 — keyschema parameter must be queuesch — (lla/dml)
- CR12 — keyschema parameter must be stacksch — (lla/dml)
- C RTP01 — Could not create from_child pipes — (ecd/rcv/fiop)
- C RTP02 — Could not create to_child pipes — (ecd/rcv/fiop)
- C RTP03 — Could not exec child errno=%d — (ecd/rcv/fiop)
- C RTP04 — Could not fork child — (ecd/rcv/fiop)
- C RTP05 — Could not get ack_init — (ecd/rcv/fiop)
- C RTP06 — Cannot malloc argv — (ecd/rcv/fiop)
- Cq01 — unable to allot queue/stack structure — (lla/am)
- Cq02 — unable to use queue/stack structure — (lla/am)
- Cq03 — unable to unuse queue/stack structure — (lla/am)
- Cr01 — cannot allocate record — (lla/dml)
- Cr02 — cannot load set header — (lla/dml)
- Cr03 — cannot unload set header — (lla/dml)
- Cr04 — cannot allocate clustered record — (lla/dml)
- Cr05 — cannot allocate record — (lla/dml)
- Cr06 — disk copy failed — (lla/dml)
- D01 — cannot load record — (lla/dml)
- D02 — cannot get record owner — (lla/dml)

- D03 — cannot delete setid — (Ila/dml)
- D04 — cannot unload record — (Ila/dml)
- D05 — cannot copy old record into memory — (Ila/dml)
- D06 — cannot load set header — (Ila/dml)
- D07 — cannot erase entry from access structure — (Ila/dml)
- D08 — cannot write set header — (Ila/dml)
- D09 — cannot move clustered record into new area — (Ila/dml)
- D10 — cannot update affected access structures — (Ila/dml)
- D11 — cannot erase record — (Ila/dml)
- D12 — cannot delete rid from rid block — (Ila/dml)
- D13 — cannot d2p variable length record to be deleted — (Ila/dml)
- DB01 — unable to use root — (Ila/am)
- DB02 — unable to unuse root — (Ila/am)
- DB03 — unable to redeem leaf — (Ila/am)
- DB04 — unable to use branch — (Ila/am)
- DB05 — failure from recall — (Ila/am)
- DB06 — unable to unuse branch — (Ila/am)
- DB07 — unable to redeem branch — (Ila/am)
- DB08 — unknown node type — (Ila/am)
- DBC01 — cannot unload DMLHEAD! — (Ila/dml)
- DBC02 — cannot unload DMLHEAD!! — (Ila/dml)
- DBC03 — sdp_disconnect failure — (Ila/dml)
- DBC04 — sdp_vacate failure — (Ila/dml)
- DBC05 — warning: Cursors still attached to this database!!!! — (Ila/dml)
- DBIM00 — unable to open key (output) file for appending — (ecd/rcv/ftamaux)

- DBIM01 — unable to close key (output) file — (ecd/rcv/ftamaux)
- DBIM02 — cannot get starting record for ucb list — (ecd/rcv/ftamaux)
- DBIM03 — specified ucb record does not exist — (ecd/rcv/ftamaux)
- DBIM04 — cannot get starting IOP record for iop list — (ecd/rcv/ftamaux)
- DBIM05 — specified ucb record does not exist — (ecd/rcv/ftamaux)
- DBIM06 — device_type for specified IOP record is not IOP — (ecd/rcv/ftamaux)
- DBIM07 — cannot get first record in side chain for iop list — (ecd/rcv/ftamaux)
- DBIM08 — cannot get next record in side chain for iop list — (ecd/rcv/ftamaux)
- DBIM09 — cannot get down record for ucb list — (ecd/rcv/ftamaux)
- DBIM10 — cannot get side record for ucb list — (ecd/rcv/ftamaux)
- DBIP00 — unable to open key (output) file for appending — (ecd/rcv/ftamaux)
- DBIP01 — unable to close key (output) file — (ecd/rcv/ftamaux)
- DBIP02 — specified record does not exist — (ecd/rcv/ftamaux)
- DBIP03 — specified form type is not correct for given key — (ecd/rcv/ftamaux)
- DBIP04 — cannot open symbol record sequence (g_ciopt) for pointer list —
(ecd/rcv/ftamaux)
- DBIP05 — cannot get next symbol record (g_ciopt) for pointer list — (ecd/rcv/ftamaux)
- DBIP06 — cannot get record information (g_ciopt) for pointer list — (ecd/rcv/ftamaux)
- DBIP07 — cannot get ciopt record (g_ciopt) for pointer list — (ecd/rcv/ftamaux)
- DBIP08 — cannot open ucb record sequence (g_ucbopt) for pointer list —
(ecd/rcv/ftamaux)
- DBIP09 — cannot get ucb record (g_ucbopt) for pointer list — (ecd/rcv/ftamaux)
- DBIP10 — cannot open ucb record sequence (g_ucbet) for pointer list — (ecd/rcv/ftamaux)
- DBIP11 — cannot get ucb record (g_ucbet) for pointer list — (ecd/rcv/ftamaux)
- DBIP12 — cannot open symbol record sequence (g_mdct) for pointer list —
(ecd/rcv/ftamaux)

- DBIP13 — cannot get symbol record (g_mdct) for pointer list — (ecd/rcv/ftamaux)
- DBIP14 — cannot get record information (g_mdct) for pointer list — (ecd/rcv/ftamaux)
- DBIP15 — cannot get mdct record (g_mdct) for pointer list — (ecd/rcv/ftamaux)
- DBIP16 — cannot open symbol record sequence (g_scscbody) for pointer list — (ecd/rcv/ftamaux)
- DBIP17 — cannot get symbol record (g_scscbody) for pointer list — (ecd/rcv/ftamaux)
- DBIP18 — cannot get record information (g_scscbody) for pointer list — (ecd/rcv/ftamaux)
- DBIP19 — cannot get scscbody record (g_scscbody) for pointer list — (ecd/rcv/ftamaux)
- DBIP20 — cannot open ucb record sequence (g_ucbucb) for pointer list — (ecd/rcv/ftamaux)
- DBIP21 — cannot get ucb record (g_ucbucb) for pointer list — (ecd/rcv/ftamaux)
- DBIP22 — cannot open symbol record sequence (g_auderr) for pointer list — (ecd/rcv/ftamaux)
- DBIP23 — cannot get next symbol record (g_auderr) for pointer list — (ecd/rcv/ftamaux)
- DBIP24 — cannot get record information (g_auderr) for pointer list — (ecd/rcv/ftamaux)
- DBIP25 — cannot get auderr record (g_auderr) for pointer list — (ecd/rcv/ftamaux)
- DBIP26 — cannot get symbol record to get name for ptr_list — (ecd/rcv/ftamaux)
- DBIP27 — cannot open symbol record sequence (g_logdev) for pointer list — (ecd/rcv/ftamaux)
- DBIP28 — cannot get next symbol record (g_logdev) for pointer list — (ecd/rcv/ftamaux)
- DBIP29 — cannot get record information (g_logdev) for pointer list — (ecd/rcv/ftamaux)
- DBIP30 — cannot get logdev record (g_logdev) for pointer list — (ecd/rcv/ftamaux)
- DBIP31 — cannot get mdct record (g_logdev) for pointer list — (ecd/rcv/ftamaux)
- DBIP32 — cannot open runa record sequence (g_runa) for pointer list — (ecd/rcv/ftamaux)
- DBIP35 — cannot get runa record (g_runa) for pointer list — (ecd/rcv/ftamaux)
- DBIP36 — cannot open clasdef record sequence (g_clasdef) for pointer list — (ecd/rcv/ftamaux)

- DBIP37 — cannot get classdef record (g_classdef) for pointer list — (ecd/rcv/ftamaux)
- DBIP38 — invalid ecd root id (g_ecdorg) — (ecd/rcv/ftamaux)
- DBIP39 — cannot get ecdorg record (g_ecdorg) for pointer list — (ecd/rcv/ftamaux)
- DBIP40 — cannot open aphuser record sequence (g_aphusr) for pointer list — (ecd/rcv/ftamaux)
- DBIP41 — cannot get aphuser record (g_aphusr) for pointer list — (ecd/rcv/ftamaux)
- DBIP42 — cannot get logdev record (g_aphuser) — (ecd/rcv/ftamaux)
- DBIP43 — error getting aphuser record (g_aphuser) — (ecd/rcv/ftamaux)
- DBIP44 — error opening aphuser set (g_aphuser) — (ecd/rcv/ftamaux)
- DBIP45 — error sequencing through aphuser set (g_aphuser) — (ecd/rcv/ftamaux)
- DBIP46 — unsupported formtype to g_aphuser function — (ecd/rcv/ftamaux)
- DBIP47 — cannot open pphuser record sequence (g_pphusr) for pointer list — (ecd/rcv/ftamaux)
- DBIP48 — cannot get pphuser record (g_pphusr) for pointer list — (ecd/rcv/ftamaux)
- DBIP49 — cannot open gfsappl record sequence (g_gfsappl) for pointer list — (ecd/rcv/ftamaux)
- DBIP50 — cannot get symbol record (g_gfsappl) for pointer list — (ecd/rcv/ftamaux)
- DBIP51 — cannot open symbol record sequence (g_gfsresp) for pointer list — (ecd/rcv/ftamaux)
- DBIP52 — cannot get symbol record (g_gfsresp) for pointer list — (ecd/rcv/ftamaux)
- DBIP53 — cannot get record information (g_gfsresp) for pointer list — (ecd/rcv/ftamaux)
- DBIP54 — cannot get gfsresp record (g_gfsresp) for pointer list — (ecd/rcv/ftamaux)
- DBIP55 — cannot open ftamopt record sequence (g_ftopt) for pointer list — (ecd/rcv/ftamaux)
- DBIP56 — cannot get ftamopt record (g_ftopt) for pointer list — (ecd/rcv/ftamaux)
- DBIP57 — cannot open symbol record sequence (g_ftinop) for pointer list — (ecd/rcv/ftamaux)
- DBIP58 — cannot get symbol record (g_ftinop) for pointer list — (ecd/rcv/ftamaux)

DBIP59 — cannot get record information (g_ftinop) for pointer list — (ecd/rcv/ftamaux)

DBIP60 — cannot get ftaminopt record (g_ftinop) for pointer list — (ecd/rcv/ftamaux)

DBIP61 — cannot open symbol record sequence (g_ftusr) for pointer list —
(ecd/rcv/ftamaux)

DBIP62 — cannot get symbol record (g_ftusr) for pointer list — (ecd/rcv/ftamaux)

DBIP63 — cannot get record information (g_ftusr) for pointer list — (ecd/rcv/ftamaux)

DBIP64 — cannot get ftamuser record (g_ftusr) for pointer list — (ecd/rcv/ftamaux)

DBIP65 — cannot open symbol record sequence (g_gfscom) for pointer list —
(ecd/rcv/ftamaux)

DBIP66 — cannot get symbol record (g_gfscom) for pointer list — (ecd/rcv/ftamaux)

DBIP67 — cannot get record information (g_gfscom) for pointer list — (ecd/rcv/ftamaux)

DBIP68 — cannot get gfscom record (g_gfscom) for pointer list — (ecd/rcv/ftamaux)

DBIP69 — cannot open symbol record sequence (g_gfssec) for pointer list —
(ecd/rcv/ftamaux)

DBIP70 — cannot get symbol record (g_gfssec) for pointer list — (ecd/rcv/ftamaux)

DBIP71 — cannot get record information (g_gfssec) for pointer list — (ecd/rcv/ftamaux)

DBIP72 — cannot get gfssec record (g_gfssec) for pointer list — (ecd/rcv/ftamaux)

DBIRID00 — Cannot get record from set — (ecd/rcv/ftamaux)

DBIRID01 — Cannot get ucb rid — (ecd/rcv/ftamaux)

DBIRID02 — Cannot get audit rid — (ecd/rcv/ftamaux)

DBIRID03 — Cannot get audinst rid — (ecd/rcv/ftamaux)

DBIRID05 — Cannot get classdef rid — (ecd/rcv/ftamaux)

DBIRID06 — Cannot get pphuser rid — (ecd/rcv/ftamaux)

DBIRID06 — Cannot get symbol record — (ecd/rcv/ftamaux)

DBIRID07 — Cannot get symbol information — (ecd/rcv/ftamaux)

DC01 — cannot load page for record — (lla/dml)

- DC01 — unable to redeem hash directory. — (lla/am)
- DC02 — cannot unload page of record — (lla/dml)
- DC02 — unable to redeem index structure. — (lla/am)
- DC03 — cannot load page for record — (lla/dml)
- DC03 — unable to destroy btree — (lla/am)
- DC04 — bad usage parameter — (lla/dml)
- DC04 — unknown access method — (lla/am)
- DC05 — unable to use linked structure — (lla/am)
- DC06 — unknown access method — (lla/am)
- DC07 — unable to unuse linked structure — (lla/am)
- DC08 — unable to redeem linked structure — (lla/am)
- DCI001 — internal error; transaction not allowed for dciopt — (ecd/rcv/integ)
- DCIOPT00 — cannot create dciopt record — (ecd/rcv/ftamecd)
- DCIOPT01 — cannot delete dciopt from ecd — (ecd/rcv/ftamecd)
- DCIOPT02 — cannot read dciopt in ecd — (ecd/rcv/ftamecd)
- DCIOPT03 — cannot update dciopt in ecd — (ecd/rcv/ftamecd)
- DD01 — unable to allot new directory — (lla/am)
- DD02 — unable to use old directory — (lla/am)
- DD03 — unable to use new directory — (lla/am)
- DD04 — unable to unuse old directory — (lla/am)
- DD05 — unable to unuse new directory — (lla/am)
- DD06 — unable to redeem old directory — (lla/am)
- DDR00 — lla_nsqrid error — (lla/porter)
- DDR01 — unable to get required record information — (lla/porter)
- DDR02 — unable to get record set owner itemid — (lla/porter)

DDR03 — cannot load entire record — (lla/porter)

DDR04 — displacement to pointer conversion error — (lla/porter)

DDR06 — fprintf error writing final record information — (lla/porter)

DDR07 — unable to get local rid — (lla/porter)

DE01 — error return from rdSPACE %s — (lla/sdp)

DE02 — fail to unlink header file %s — (lla/sdp)

DE03 — fail to build name %s%d — (lla/sdp)

DE04 — fail to unlink file %s — (lla/sdp)

DE05 — unknown descriptor value %d — (lla/sdp)

DEAU — audit failed in sdp_destroy — (lla/sdp)

DEVR00 — cannot insert DEVR set — (ecd/rcv/ftamecd)

DEVR01 — cannot create DEVR record — (ecd/rcv/ftamecd)

DEVR02 — cannot delete DEVR record — (ecd/rcv/ftamecd)

DEVR03 — cannot read DEVR record — (ecd/rcv/ftamecd)

DEVR04 — cannot update DEVR record — (ecd/rcv/ftamecd)

DEVR05 — Cannot open set of classdef records — (ecd/rcv/ftamecd)

DEVR05 — cannot get DEVR record — (ecd/rcv/ftamecd)

DEVR06 — Cannot get record from classdef set — (ecd/rcv/ftamecd)

DFCOPT00 — cannot create DFCOPT record — (ecd/rcv/ftamecd)

DFCOPT01 — cannot delete dfcopt from ecd — (ecd/rcv/ftamecd)

DFCOPT02 — cannot read dfcopt in ecd — (ecd/rcv/ftamecd)

DFCOPT03 — cannot update dfcopt in ecd — (ecd/rcv/ftamecd)

DFP01 — cannot load ridblk info — (lla/dml)

DFP02 — invalid ridblk id — (lla/dml)

DFP03 — cannot update dml header — (lla/dml)

DFP04 — cannot write ridblk — (lla/dml)

DI01 — space to be disconnected has item in use %o — (lla/sdp)

DI02 — permission conflict %dt%dt%d — (lla/sdp)

DI03 — error return from writeframe %o — (lla/sdp)

DI04 — fail to close open file %o — (lla/sdp)

DI05 — error return from wrtSPACE %o — (lla/sdp)

DI06 — space not connected %o — (lla/sdp)

DIAU — audit failed in sdp_disconnect — (lla/sdp)

DLAPU00 — cannot load aphuser record — (ecd/rcv/ftamaux)

DLFTMI00 — cannot load ftaminopt record — (ecd/rcv/ftamaux)

DLFTMO00 — cannot load ftamopt record — (ecd/rcv/ftamaux)

DLFTMU00 — cannot load ftamuser record — (ecd/rcv/ftamaux)

DLGFSA00 — cannot load gfsappl record — (ecd/rcv/ftamaux)

DLGFSC00 — cannot load gfscom record — (ecd/rcv/ftamaux)

DLGFSR00 — cannot load gfsresp record — (ecd/rcv/ftamaux)

DLGFSS00 — cannot load gfssec record — (ecd/rcv/ftamaux)

DLK01 — cannot decrement reference count — (ecd/rcv/prelinks)

DLPPU00 — cannot load pphuser record — (ecd/rcv/ftamaux)

DML00 — cannot load DML header structure — (lla/audit)

DML01 — cannot load starting set header — (lla/audit)

DML02 — cannot unload starting set header — (lla/audit)

DML03 — cannot load ending set header — (lla/audit)

DML04 — cannot unload ending set header — (lla/audit)

DML05 — unable to find starting set header — (lla/audit)

DML06 — unable to find ending set header — (lla/audit)

- DML07 — cannot load starting rid block — (lla/audit)
- DML08 — cannot unload starting rid block — (lla/audit)
- DML09 — cannot load ending rid block — (lla/audit)
- DML10 — cannot unload ending rid block — (lla/audit)
- DML11 — unable to find starting rid block — (lla/audit)
- DML12 — unable to find ending rid block — (lla/audit)
- DML13 — cannot unload DML header structure — (lla/audit)
- DML14 — unable to find rid block that links to starting rid block — (lla/audit)
- DML15 — unable to find rid block that links to ending rid block — (lla/audit)
- DML16 — unable to find set header that links to starting set header — (lla/audit)
- DML17 — unable to find set header that links to ending set header — (lla/audit)
- DMLAUD00 — unable to load DMLHEAD structure with SDP — (lla/audit)
- DMLAUD01 — DML sanity id at itemid = 0 found to have incorrect value 0x%x — (lla/audit)
- DMLAUD02 — DML page size found to have incorrect value %d, should be %d — (lla/audit)
- DMLAUD03 — number of rids in ridblk per page found to have incorrect value %d, should be %d — (lla/audit)
- DMLAUD04 — start of set header list found to have incorrect value 0x%x — (lla/audit)
- DMLAUD05 — set start points to bad set header at 0x%x — (lla/audit)
- DMLAUD06 — end of set header list found to have incorrect value 0x%x — (lla/audit)
- DMLAUD07 — set end points to bad set header at 0x%x — (lla/audit)
- DMLAUD08 — set start itemid being reset to 0x%x from 0x%x — (lla/audit)
- DMLAUD09 — set end itemid being reset to 0x%x from 0x%x — (lla/audit)
- DMLAUD10 — start of rid block list found to have incorrect value 0x%x — (lla/audit)
- DMLAUD11 — rid block start points to bad rid block header at 0x%x — (lla/audit)
- DMLAUD12 — end of rid block list found to have incorrect value 0x%x — (lla/audit)

- DMLAUD13 — rid block end points to bad rid block at 0x%lx — (Ila/audit)
- DMLAUD14 — rid block start itemid being reset to 0x%lx from 0x%lx — (Ila/audit)
- DMLAUD15 — rid block end itemid being reset to 0x%lx from 0x%lx — (Ila/audit)
- DMLAUD16 — SPACE page size found to have incorrect value %d, should be %d — (Ila/audit)
- DMLAUD17 — SPACE curleft being reset from 0x%lx to 0x%lx — (Ila/audit)
- DMLAUDR0 — start and end rid block links being reset from 0x%lx, 0x%lx to 0x%lx,0x%lx — (Ila/audit)
- DMLAUDR1 — start and end set header links being reset from 0x%lx, 0x%lx to 0x%lx,0x%lx — (Ila/audit)
- DMLR00 — error trying to find a rid block — (Ila/audit)
- DMLR01 — error trying to find start and end rid block — (Ila/audit)
- DMLR02 — unable to load DMLHEAD to reset rid block links — (Ila/audit)
- DMLR03 — unable to write updated rid block links — (Ila/audit)
- DMLR04 — error trying to find a set header — (Ila/audit)
- DMLR05 — error trying to find start and end set headers — (Ila/audit)
- DMLR06 — unable to load DMLHEAD to reset set header links — (Ila/audit)
- DMLR07 — unable to write updated set header links — (Ila/audit)
- DP01 — cannot allocate space for cursor — (Ila/dml)
- DS00 — fprintf error writing database info. — (Ila/porter)
- DS01 — cannot change record status — (Ila/dml)
- DS02 — cannot delete non-existing owner — (Ila/dml)
- DS03 — cannot change record status — (Ila/dml)
- DS04 — cannot load set member block — (Ila/dml)
- DS05 — cannot unload set member block — (Ila/dml)
- DS06 — cannot redeem space for old set member block — (Ila/dml)

- DS07 — cannot change record status — (lla/dml)
- DS08 — cannot write set member block — (lla/dml)
- DS09 — cannot delete non-existing owner — (lla/dml)
- DS10 — inconsistent record header — (lla/dml)
- DS11 — cannot delete non-existent owner — (lla/dml)
- DSK001 — internal error; transaction not allowed for dfcopt — (ecd/rcv/integ)
- DSK002 — internal error; transaction not allowed for mhdopt — (ecd/rcv/integ)
- DST00 — cannot create eaiopt record — (ecd/rcv/ftamecd)
- DST01 — cannot delete eaiopt from ecd — (ecd/rcv/ftamecd)
- DST02 — cannot read eaiopt in ecd — (ecd/rcv/ftamecd)
- DST03 — cannot update eaiopt in ecd — (ecd/rcv/ftamecd)
- DSTRAUD1400 — offset to DML structure being reset from 0x%lx to 0x%lx —
(ecd/ecdaudit/aud2)
- DSTRAUD1401 — offset to DML structure found to be 0x%lx but should be 0x%lx —
(ecd/ecdaudit/aud2)
- DSTRAUD1402 — disk database sanity id being reset from 0x%lx to 0x%lx —
(ecd/ecdaudit/aud2)
- DSTRAUD1403 — disk database sanity id found to be 0x%lx and should be 0x%lx —
(ecd/ecdaudit/aud2)
- DSTRAUD1404 — offset to sdp SPACE structure being reset from 0x%lx to 0x%lx —
(ecd/ecdaudit/aud2)
- DSTRAUD1405 — offset to sdp SPACE structure found to be 0x%lx but should be
0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1406 — sanity id for DML structure found to be 0x%lx in %s but should
be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1407 — either user rootid in STARTUP(=0x%lx) or DMLHEAD(=0x%lx) is
bad, use fsdb to patch — (ecd/ecdaudit/aud2)
- DSTRAUD1408 — disk database size in STARTUP being reset from 0x%lx to 0x%lx —
(ecd/ecdaudit/aud2)

- DSTRAUD1409 — disk database size in STARTUP found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1410 — disk database size in sdp SPACE structure being reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1411 — disk database size in sdp SPACE structure found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1412 — number of sdp stax being reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1413 — number of sdp stax found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1414 — sdp SPACE structure page size reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1415 — sdp SPACE structure found to have page size 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1416 — address of file information structure being reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1417 — address of file information structure found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1418 — address of stax information structure being reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1419 — address of stax information structure found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1420 — address of environment structure reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1421 — address of environment structure found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1422 — address of head name reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1423 — address of head name found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1424 — address of manager structure reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1425 — address of manager structure found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1426 — number of frames in environment structure reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1427 — number of frames in environment found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1428 — frame pointer address reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1429 — frame pointer address found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1430 — manager[0] space pointer reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1431 — manager[0] space pointer found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1432 — manager[0] page id reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1433 — manager[0] page id found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1434 — manager[0] use count reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1435 — manager[0] use count found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1436 — manager[0] forward pointer reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1437 — manager[0] forward pointer found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1438 — manager[0] backward pointer reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1439 — manager[0] backward pointer found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)

DSTRAUD1440 — manager[1] page id reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)

- DSTRAUD1441 — manager[1] page id found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1442 — manager[1] use count reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1443 — manager[1] use count found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1444 — manager[1] forward pointer reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1445 — manager[1] forward pointer found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1446 — manager[1] backward pointer reset from 0x%lx to 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1447 — manager[1] backward pointer found to be 0x%lx but should be 0x%lx — (ecd/ecdaudit/aud2)
- DSTRAUD1448 — no corrections made — (ecd/ecdaudit/aud2)
- DSTRAUD1449 — no corrections made — (ecd/ecdaudit/aud2)
- DU01 — cannot load specified item — (lla/dml)
- DU02 — cannot unload specified item — (lla/dml)
- DU03 — not specified item type — (lla/dml)
- DU04 — cannot load record — (lla/dml)
- DU05 — cannot unload bad record — (lla/dml)
- DU06 — bad record header — (lla/dml)
- DUDYN00 — bad rid stored in symbol table — (ecd/rcv/ftamecd)
- DUOPT00 — cannot create duopt record — (ecd/rcv/ftamecd)
- DUOPT01 — cannot delete duopt from ecd — (ecd/rcv/ftamecd)
- DUOPT02 — cannot read duopt in ecd — (ecd/rcv/ftamecd)
- DUOPT03 — cannot update duopt in ecd — (ecd/rcv/ftamecd)
- Db_load — cross form check violated — (ecd/rcv/prelinks)

- Db_load — dump file format does not match mask format — (ecd/rcv/prelinks)
- EAILOPT00 — cannot create eaiopt record — (ecd/rcv/ftamecd)
- EAILOPT01 — cannot delete eaiopt from ecd — (ecd/rcv/ftamecd)
- EAILOPT02 — cannot read eaiopt in ecd — (ecd/rcv/ftamecd)
- EAILOPT03 — cannot update eaiopt in ecd — (ecd/rcv/ftamecd)
- EAILOPT04 — cannot get EAILOPT record — (ecd/rcv/ftamecd)
- EAILOPT05 — changing min_psswd_chg is not allowed — (ecd/rcv/ftamecd)
- EAILOPT06 — changing max_psswd_chg is not allowed — (ecd/rcv/ftamecd)
- EAILOPT07 — changing dsable_attempt is not allowed — (ecd/rcv/ftamecd)
- EAILOPT08 — You are about to turn off password enhancement — (ecd/rcv/ftamecd)
- EB01 — unable to use current node — (lla/am)
- EB02 — unable to unuse current node in leaf — (lla/am)
- EB03 — unable to reuse current node in branch — (lla/am)
- EB04 — unable to unuse current node in branch — (lla/am)
- EB05 — unable to reuse node in branch — (lla/am)
- EB06 — bad return from coalesce — (lla/am)
- EB07 — unable to unuse current node — (lla/am)
- EB08 — unable to use child — (lla/am)
- EB09 — unable to unuse child — (lla/am)
- EB10 — unable to use left child — (lla/am)
- EB11 — unable to use right child — (lla/am)
- EB12 — unable to use left next — (lla/am)
- EB13 — unable to unuse left next — (lla/am)
- EB14 — unable to unuse left child — (lla/am)
- EB15 — unable to unuse right child — (lla/am)

EB16 — unable to redeem right child — (lla/am)

EB17 — unable to redeem child — (lla/am)

EB18 — unable to change left child — (lla/am)

EB19 — unable to change child — (lla/am)

EC01 — fail to find rid in btree — (lla/am)

EC02 — unable to use btree root — (lla/am)

EC03 — unable to redeem btree root — (lla/am)

EC04 — unable to unuse btree root — (lla/am)

EC05 — unknown access method — (lla/am)

ECDORG 0 — cannot change limp_mode in INCORE database — (ecd/rcv/ftamecd)

ECDORG00 — cannot create ECDORG record — (ecd/rcv/ftamecd)

ECDORG01 — cannot read ecdorg in ecd — (ecd/rcv/ftamecd)

ECDORG02 — cannot update ecdorg in ecd — (ecd/rcv/ftamecd)

ECDORG03 — cannot break old links — (ecd/rcv/ftamecd)

ECDORG04 — illegal database value for h_dlm — (ecd/rcv/ftamecd)

ECDORG05 — illegal form value for limp_mode — (ecd/rcv/ftamecd)

EH01 — error return from open_hash — (lla/am)

EH02 — error return from next_hash — (lla/am)

EH03 — error return from re_place — (lla/am)

EH04 — unable to find rid in hash — (lla/am)

EH05 — unable to use bucket — (lla/am)

EH06 — unable to unuse bucket — (lla/am)

EH07 — unable to use main bucket — (lla/am)

EH08 — unable to find hash value in main bucket — (lla/am)

EH09 — unable to use overflow — (lla/am)

EH10 — unable to find hash value in overflow — (lla/am)
EH11 — unable to find hash value in overflow — (lla/am)
EH12 — unable to unuse main bucket — (lla/am)
EH13 — unable to unuse overflow — (lla/am)
EH14 — unable to unuse bucket — (lla/am)
EQ01 — empty queue — (lla/am)
EQ02 — unable to use head id structure initially — (lla/am)
EQ03 — unable to unuse head id structure initially — (lla/am)
EQ04 — mismatched rid in queue — (lla/am)
EQ05 — unable to use head id before redeem — (lla/am)
EQ06 — unable to redeem head id — (lla/am)
EQ07 — unable to unuse head id after redeem — (lla/am)
EQ08 — unknown position — (lla/am)
ER01 — cannot load set — (lla/dml)
ER02 — cannot unload set — (lla/dml)
ER03 — cannot redeem clustered record — (lla/dml)
ER04 — cannot redeem record — (lla/dml)
ERR00A — sdp_house error returned — (ecd/dgen)
ERR00B — sdp_connect error returned — (ecd/dgen)
ERR00 — sdp_use error returned — (ecd/dgen)
ERR01A — negative growth — (ecd/dgen)
ERR01C — unable to allocate — (ecd/dgen)
ERR01 — sdp_unuse error returned — (ecd/dgen)
ERR02 — unable to allocate — (ecd/dgen)
ERR03A — lseek error — (ecd/dgen)

ERR03A — sdp_use error — (ecd/dgen)
ERR03B — error in writing — (ecd/dgen)
ERR03 — error in writing loadfile header — (ecd/dgen)
ERR04 — unable to allocate — (ecd/dgen)
ERR05 — unable to allocate — (ecd/dgen)
ERR06 — unable to allocate — (ecd/dgen)
ERR07 — error in writing load file — (ecd/dgen)
ERR08A — error in writing loadfile — (ecd/dgen)
ERR08 — error in writing load file — (ecd/dgen)
ERR09 — error in opening data file — (ecd/dgen)
ERR10 — lseek error — (ecd/dgen)
ERR11 — error in reading — (ecd/dgen)
ERR12 — error in writing — (ecd/dgen)
ERR13 — growth size error — (ecd/dgen)
ERR14 — error in writing — (ecd/dgen)
ERRCNT00 — could not retrieve ucb with errcount — (ecd/rcv/ftamecd)
ERRCNT01 — unable to locate errcount record — (ecd/rcv/ftamecd)
ERRCNT02 — vector size exceeds form size — (ecd/rcv/ftamecd)
ERRF00 — cannot open %s: errno = %d — (ecd/rcv/prelinks)
ERRF01 — cannot create %s: errno = %d — (ecd/rcv/prelinks)
ERRF02 — cannot open %s: errno = %d — (ecd/rcv/prelinks)
ETHLD00 — cannot create errthld record — (ecd/rcv/ftamecd)
ETHLD01 — cannot delete errthld from ecd — (ecd/rcv/ftamecd)
ETHLD02 — cannot read errthld in ecd — (ecd/rcv/ftamecd)
ETHLD03 — cannot update errthld in ecd — (ecd/rcv/ftamecd)

- ETHLD04 — cannot get old errthld reference count — (ecd/rcv/ftamecd)
- ETHLD05 — cannot load errthld record — (ecd/rcv/ftamecd)
- ETHLD06 — cannot open sequence through ucbs — (ecd/rcv/ftamecd)
- ETHLD07 — cannot sequence through ucbs — (ecd/rcv/ftamecd)
- ETHLD08 — cannot update ucb l_errc — (ecd/rcv/ftamecd)
- ETHLD09 — cannot delete old related errcount record — (ecd/rcv/ftamecd)
- FA01 — unable to find next frame to use — (lla/sdp)
- FA02 — error return from writeframe — (lla/sdp)
- FA03 — unknown ref/change value %dt%ot%d — (lla/sdp)
- FA04 — error return from readpage — (lla/sdp)
- FD001 — illegal value specified, probably a null field — (ecd/rcv/prelinks)
- FD00 — cannot convert character vector to encoded — (ecd/rcv/prelinks)
- FD01 — cannot convert encoded structure to character vector — (ecd/rcv/prelinks)
- FDD00 — cannot create & open request file — (ecd/rcv/ftamsp)
- FDRV01 — Invalid database operation — (ecd/rcv/fiop)
- FDRV02 — Invalid request from process — (ecd/rcv/fiop)
- FDRV03 — Invalid exit from infinite loop — (ecd/rcv/fiop)
- FDRV04 — Cannot read screen from process — (ecd/rcv/fiop)
- FDRV05 — Cannot malloc 1 in init_form — (ecd/rcv/fiop)
- FDRV06 — Cannot modinit in init_form — (ecd/rcv/fiop)
- FDRV07 — Cannot malloc 2 in init_form — (ecd/rcv/fiop)
- FDRV08 — Sending xfr_entry — (ecd/rcv/fiop)
- FDRV09 — Getting req_get_key formarea — (ecd/rcv/fiop)
- FDRV10 — Sending ack_get_key — (ecd/rcv/fiop)
- FDRV11 — Sending ack_get_key formarea — (ecd/rcv/fiop)

- FDRV12 — Sending ack_fields — (ecd/rcv/fiop)
- FDRV13 — Sending xfr_field — (ecd/rcv/fiop)
- FDRV14 — Sending xfr_field fieldname — (ecd/rcv/fiop)
- FDRV16 — Could not make tempfile for create — (ecd/rcv/fiop)
- FDRV17 — Could not creat tempfile — (ecd/rcv/fiop)
- FDRV18 — Could not write tempfile — (ecd/rcv/fiop)
- FDRV19 — Could not send ack_create — (ecd/rcv/fiop)
- FDRV20 — Illegal value to MXset_value from xfr_field — (ecd/rcv/fiop)
- FDRV21 — Illegal return status in xfr_status — (ecd/rcv/fiop)
- FDRV22 — Could not get fieldname for xfr_field — (ecd/rcv/fiop)
- FDRV23 — Could not get fielddata for xfr_field — (ecd/rcv/fiop)
- FDRV24 — Could not get fieldnumber for xfr_field — (ecd/rcv/fiop)
- FDRV25 — Could not get message text for xfr_message — (ecd/rcv/fiop)
- FDRV26 — Bad msgtype for xfr_message — (ecd/rcv/fiop)
- FDRV27 — Message exceeds maximum size — (ecd/rcv/fiop)
- FDRV28 — Could not get formarea to execute — (ecd/rcv/fiop)
- FDRV29 — Could not send ack_execute — (ecd/rcv/fiop)
- FDRV30 — Could not send ack_execute — (ecd/rcv/fiop)
- FDRV31 — Could not send ack_execute — (ecd/rcv/fiop)
- FDRV32 — Could not get formarea for update — (ecd/rcv/fiop)
- FDRV33 — Could not send ack_update — (ecd/rcv/fiop)
- FDRV34 — Could not send ack_update — (ecd/rcv/fiop)
- FDRV35 — Could not get formarea for insert — (ecd/rcv/fiop)
- FDRV36 — Could not send ack_insert — (ecd/rcv/fiop)
- FDRV37 — Could not send ack_insert — (ecd/rcv/fiop)

- FDRV38 — Could not get formarea for delete — (ecd/rcv/fiop)
- FDRV39 — Could not send ack_delete — (ecd/rcv/fiop)
- FDRV40 — Could not send ack_delete — (ecd/rcv/fiop)
- FDRV41 — Could not get prompt message — (ecd/rcv/fiop)
- FDRV42 — Bad return from tpread in prompt — (ecd/rcv/fiop)
- FDRV43 — Could not send ack_prompt — (ecd/rcv/fiop)
- FDRV44 — Could not send response — (ecd/rcv/fiop)
- FDRV45 — Could not get default spec 1 — (ecd/rcv/fiop)
- FDRV46 — Could not get default spec 2 — (ecd/rcv/fiop)
- FDRV47 — Could not send ack_default — (ecd/rcv/fiop)
- FDRV48 — Could not get formarea for default form — (ecd/rcv/fiop)
- FDRV49 — Could not send ackdefreturn — (ecd/rcv/fiop)
- FDRV50 — Could not send formarea for default return — (ecd/rcv/fiop)
- FDRV51 — Could not send ack_deq — (ecd/rcv/fiop)
- FDRV52 — Failure from MXerror_msg — (ecd/rcv/fiop)
- FDRV53 — Unknow message type from MXerror_msg — (ecd/rcv/fiop)
- FDRV54 — Could not send ackdeqmsg — (ecd/rcv/fiop)
- FDRV55 — Could not send msgtxt — (ecd/rcv/fiop)
- FDRV56 — Could not get formarea for check — (ecd/rcv/fiop)
- FDRV57 — Could not send ack_check — (ecd/rcv/fiop)
- FDRV58 — Could not perform implicit trend — (ecd/rcv/fiop)
- FDRV59 — Could not send ack_deq_relay — (ecd/rcv/fiop)
- FDRV60 — Could not calloc pgm array — (ecd/rcv/fiop)
- FDRV61 — Could not get pgm_inst — (ecd/rcv/fiop)
- FDRV62 — NULL returned from what_form — (ecd/rcv/fiop)

- FDRV63 — Could not malloc operand1 — (ecd/rcv/fiop)
- FDRV64 — Could not get operand1 — (ecd/rcv/fiop)
- FDRV65 — Could not malloc operand2 — (ecd/rcv/fiop)
- FDRV66 — Could not get operand2 — (ecd/rcv/fiop)
- FDRV67 — Bad fieldname for MXnumber — (ecd/rcv/fiop)
- FDRV68 — Could not set value with MXset_value — (ecd/rcv/fiop)
- FDRV70 — Could not send ackgetsearch — (ecd/rcv/fiop)
- FDRV71 — Could not get value with MXvalue — (ecd/rcv/fiop)
- FDRV72 — Form not in formtable — (ecd/rcv/fiop)
- FDRV73 — Could not make tmpfile name — (ecd/rcv/fiop)
- FDRV74 — Could not creat tmpfile — (ecd/rcv/fiop)
- FDRV75 — Error in kl_tab — (ecd/rcv/fiop)
- FDRV76 — Could not reopen klistfile — (ecd/rcv/fiop)
- FDRV77 — Read error of klistfile — (ecd/rcv/fiop)
- FDRV78 — Could not get form — (ecd/rcv/fiop)
- FDRV79 — Error from MXnumber — (ecd/rcv/fiop)
- FDRV80 — Error from MXvalue — (ecd/rcv/fiop)
- FDRV81 — Write error to formfile — (ecd/rcv/fiop)
- FDRV82 — Could not send ackgetsearch — (ecd/rcv/fiop)
- FDRV83 — Execution stack overflow — (ecd/rcv/fiop)
- FDRV84 — Execution stack underflow — (ecd/rcv/fiop)
- FDRV85 — Could not send xfr_screen — (ecd/rcv/fiop)
- FDRV86 — Could not send screen image — (ecd/rcv/fiop)
- FDRV87 — Bad opcode — (ecd/rcv/fiop)
- FDRV88 — Could not malloc in drv_default — (ecd/rcv/fiop)

FDRV89 — Could not open journal file — (ecd/rcv/fiop)
FDRV90 — Could not close journal file — (ecd/rcv/fiop)
FDRV91 — Could not delete journal file — (ecd/rcv/fiop)
FDRV92 — implicit transaction abort failed — (ecd/rcv/fiop)
FH00 — sdp_sync error — (lla/dml)
FH01 — error return from writeframe %o — (lla/sdp)
FH01 — unable to use directory — (lla/am)
FH02 — unable to unuse directory — (lla/am)
FH02 — unknown ref/chng flag %ot%d — (lla/sdp)
FH03 — error return from wtSPACE %o — (lla/sdp)
FH03 — illegal search type — (lla/am)
FH04 — space not connected %o — (lla/sdp)
FH04 — unknown search type — (lla/am)
FHAU — audit failed in sdp_flush — (lla/sdp)
FLC00 — cannot open file to copy it — (ecd/rcv/ftamaux)
FLC01 — cannot get size of file to copy — (ecd/rcv/ftamaux)
FLC02 — cannot create new contiguous file — (ecd/rcv/ftamaux)
FLC03 — cannot open existing contiguous file — (ecd/rcv/ftamaux)
FLC04 — cannot get size of destination file — (ecd/rcv/ftamaux)
FLC05 — cannot recreate contiguous file — (ecd/rcv/ftamaux)
FLC06 — cannot copy file — (ecd/rcv/ftamaux)
FLC07 — cannot read file to copy — (ecd/rcv/ftamaux)
FLC08 — unable to update destination file — (ecd/rcv/ftamaux)
FLC09 — cannot malloc buffer for read — (ecd/rcv/ftamaux)
FO00 — unable to get next page to scan — (lla/audit)

- FO01 — cannot load set — (lla/dml)
- FO01 — unable to unuse page with match — (lla/audit)
- FO02 — cannot unload set — (lla/dml)
- FO02 — unable to unuse page without any matches — (lla/audit)
- FO03 — inconsistent record header — (lla/dml)
- FP 0 — associated pack form must be defined before fp form — (ecd/rcv/ftamsg)
- FP 1 — can not delete fp while associated fpb exist — (ecd/rcv/ftamsg)
- FP00 — cannot get pack information — (ecd/rcv/ftamsg)
- FP01 — cannot create new file partition — (ecd/rcv/ftamsg)
- FP02 — cannot get file partition data — (ecd/rcv/ftamsg)
- FP03 — cannot update file partition data — (ecd/rcv/ftamsg)
- FP04 — cannot delete file partition record — (ecd/rcv/ftamsg)
- FPB00 — cannot get pack data — (ecd/rcv/ftamsg)
- FPB01 — cannot get partition data — (ecd/rcv/ftamsg)
- FPB02 — cannot get file partition body record — (ecd/rcv/ftamsg)
- FPB03 — cannot create ftbody record or update part_info record — (ecd/rcv/ftamsg)
- FPB04 — cannot update part_info record for ftbody deletion — (ecd/rcv/ftamsg)
- FPB05 — cannot delete ftbody record — (ecd/rcv/ftamsg)
- FPB06 — cannot update ftbody record — (ecd/rcv/ftamsg)
- FPB0 — associated pack form must be defined before fpb form — (ecd/rcv/ftamsg)
- FPB1 — associated part form must be defined before fpb form — (ecd/rcv/ftamsg)
- FPB2 — another fpb specified for this partition — (ecd/rcv/ftamsg)
- FPB3 — no fpb for specified partition — (ecd/rcv/ftamsg)
- FRE00 — unable to load current rid block — (lla/audit)
- FRE01 — unable to set dmlhead to write status — (lla/audit)

- FRE02 — unable to unload current rid block — (lla/audit)
- FREAUD00 — rid block 0x%lx being put on free rid block list at head — (lla/audit)
- FREAUD01 — rid block 0x%lx on free list has no free slots — (lla/audit)
- FRL00 — unable to load current rid block — (lla/audit)
- FRL01 — unable to load next free rid block — (lla/audit)
- FRL02 — unable to unload next free rid block — (lla/audit)
- FRL03 — unable to unload current rid block — (lla/audit)
- FRL04 — unable to load rid block at start of free list — (lla/audit)
- FRL05 — unable to unload rid block at start of free list — (lla/audit)
- FRLAUD00 — rid block 0x%lx next free link found to have incorrect value 0x%lx — (lla/audit)
- FRLAUD01 — dmlhead link to start of rid block free list found to have incorrect value 0x%lx — (lla/audit)
- FRLAUDR00 — rid block 0x%lx next free link being reset from 0x%lx to 0x%lx — (lla/audit)
- FRLAUDR01 — resetting free rid block id in dmlhead from 0x%lx to 0x%lx — (lla/audit)
- FRLR00 — unable to load current rid block — (lla/audit)
- FRLR01 — unable to unload current rid block — (lla/audit)
- FRLR02 — unable to set dmlhead to write status — (lla/audit)
- FS 0 — associated pack form must be defined before fs form — (ecd/rcv/ftamsg)
- FS 1 — can not delete fs while associated fsb(s) exist — (ecd/rcv/ftamsg)
- FS00 — cannot get pack information — (ecd/rcv/ftamsg)
- FS01 — cannot create new file system partition — (ecd/rcv/ftamsg)
- FS02 — cannot get file system data — (ecd/rcv/ftamsg)
- FS03 — cannot update file system data — (ecd/rcv/ftamsg)
- FSB00 — cannot get pack data — (ecd/rcv/ftamsg)
- FSB01 — cannot get partition data — (ecd/rcv/ftamsg)

- FSB02 — cannot get preceding fsbody record — (ecd/rcv/ftamsg)
- FSB03 — cannot get specified fsbody record — (ecd/rcv/ftamsg)
- FSB04 — cannot create specified fsb record — (ecd/rcv/ftamsg)
- FSB05 — cannot update specified fsb record — (ecd/rcv/ftamsg)
- FSB06 — cannot delete specified fsb record — (ecd/rcv/ftamsg)
- FSB07 — cannot create fsbody set & update owner — (ecd/rcv/ftamsg)
- FSB08 — cannot remove fsb from set — (ecd/rcv/ftamsg)
- FSB09 — cannot remove empty set — (ecd/rcv/ftamsg)
- FSB10 — cannot update partrec if empty set — (ecd/rcv/ftamsg)
- FSB11 — cannot update fsb if empty set — (ecd/rcv/ftamsg)
- FSB1 — associated pack form must be defined before fsb form — (ecd/rcv/ftamsg)
- FSB2 — associated fs form must be defined before fsb form — (ecd/rcv/ftamsg)
- FSB3 — invalid path name to fsb — (ecd/rcv/ftamsg)
- FSB4 — could not find indicated file system body — (ecd/rcv/ftamsg)
- FSB5 — may not change type of directory with members — (ecd/rcv/ftamsg)
- FSB6 — may not delete directory with members — (ecd/rcv/ftamsg)
- FTMI00 — cannot create FTAMINOPT record — (ecd/rcv/ftamecd)
- FTMI01 — cannot delete FTAMINOPT from ecd — (ecd/rcv/ftamecd)
- FTMI02 — cannot read FTAMINOPT in ecd — (ecd/rcv/ftamecd)
- FTMI03 — cannot update FTAMINOPT in ecd — (ecd/rcv/ftamecd)
- FTMO00 — cannot create FTAMOPT record — (ecd/rcv/ftamecd)
- FTMO01 — cannot delete FTAMOPT record — (ecd/rcv/ftamecd)
- FTMO02 — cannot read FTAMOPT record — (ecd/rcv/ftamecd)
- FTMO03 — cannot update FTAMOPT record — (ecd/rcv/ftamecd)
- FTMU00 — cannot create FTAMUSER record — (ecd/rcv/ftamecd)

FTMU01 — cannot delete FTAMUSER record — (ecd/rcv/ftamecd)

FTMU02 — cannot read FTAMUSER record — (ecd/rcv/ftamecd)

FTMU03 — cannot update FTAMUSER record — (ecd/rcv/ftamecd)

G01 — cannot locate rid — (lla/dml)

G02 — cannot load record — (lla/dml)

GBT00 — cannot open /dev/pmem to get boot info — (ecd/rcv/ftamaux)

GBT01 — unsuccessful read of /dev/pmem to get boot info — (ecd/rcv/ftamaux)

GD02 — form checks skipped — (ecd/rcv/prelinks)

GD03 — See error file, dump error rc=%d, formcode = %s, record: — (ecd/rcv/prelinks)

GD03 — error loading data for form: — (ecd/rcv/prelinks)

GD03 — forms with warnings only — (ecd/rcv/prelinks)

GD04 — form with range or intra-form errors — (ecd/rcv/prelinks)

GD05 — duplicate form encountered — (ecd/rcv/prelinks)

GD06 — error reading form — (ecd/rcv/prelinks)

GE01 — fail to allocate header %s — (lla/sdp)

GE02 — fail to allocate for file structure %s — (lla/sdp)

GE03 — fail to allocate stacks structure %s — (lla/sdp)

GE04 — fail to build name %s — (lla/sdp)

GE05 — fail to create header file %s — (lla/sdp)

GE06 — error return from wtSPACE %s — (lla/sdp)

GEAU — audit failed in sdp_generate — (lla/sdp)

GEN00 — no database specified — (ecd/rcv/links)

GEN01 — high level feature is not available: hlmode parameter is illegal! — (ecd/rcv/links)

GEN02 — no masks specified — (ecd/rcv/links)

GEN03 — bad input parameter %s — (ecd/rcv/links)

- GEN04 — cannot default mask directory — (ecd/rcv/links)
- GEN05 — mask pathname too long — (ecd/rcv/links)
- GEN06 — Unable to set path to bin, verify OFC and HOST variables are set — (ecd/rcv/links)
- GEN07 — cannot malloc mask_dir — (ecd/rcv/links)
- GEN09 — cannot determine high level feature availability — (ecd/rcv/links)
- GETTY00 — cannot create GETTY record — (ecd/rcv/ftamecd)
- GETTY01 — cannot delete getty from ecd — (ecd/rcv/ftamecd)
- GETTY02 — cannot read getty in ecd — (ecd/rcv/ftamecd)
- GETTY03 — cannot update getty in ecd — (ecd/rcv/ftamecd)
- GFSA00 — cannot create gfsappl record — (ecd/rcv/ftamecd)
- GFSA01 — cannot delete gfsappl from ecd — (ecd/rcv/ftamecd)
- GFSA02 — cannot read gfsappl in ecd — (ecd/rcv/ftamecd)
- GFSA03 — cannot update gfsappl in ecd — (ecd/rcv/ftamecd)
- GFSA04 — cannot get gfsappl record — (ecd/rcv/ftamecd)
- GFSAPL_0 — User is not allowed to change the alw_prefix_chng field from 'n' to
- GFSAPL_1 — When alw_prefix_chng field is set to 'n', user cannot change prefix list — (ecd/rcv/ftamecd)
- GFSAPL_2 — User is not allowed to have duplicate or overlapping pathname prefixes — (ecd/rcv/ftamecd)
- GFSC00 — cannot create gfscom record — (ecd/rcv/ftamecd)
- GFSC01 — cannot delete gfscom from ecd — (ecd/rcv/ftamecd)
- GFSC02 — cannot read gfscom in ecd — (ecd/rcv/ftamecd)
- GFSC03 — cannot update gfscom in ecd — (ecd/rcv/ftamecd)
- GFSC04 — FTAM1 objectid could not be linked — (ecd/rcv/ftamecd)
- GFSC05 — FTAM2 objectid could not be linked — (ecd/rcv/ftamecd)

GFSC06 — FTAM3 objectid could not be linked — (ecd/rcv/ftamecd)
GFSC07 — FTAM4 objectid could not be linked — (ecd/rcv/ftamecd)
GFSC08 — cannot load gfscom record — (ecd/rcv/ftamecd)
GFSC09 — error with old objectid linkage — (ecd/rcv/ftamecd)
GFSI00 — cannot create gfsinit record — (ecd/rcv/ftamecd)
GFSI01 — cannot delete gfsinit from ecd — (ecd/rcv/ftamecd)
GFSI02 — cannot read gfsinit in ecd — (ecd/rcv/ftamecd)
GFSI03 — cannot update gfsinit in ecd — (ecd/rcv/ftamecd)
GFSR00 — cannot create gfsresp record — (ecd/rcv/ftamecd)
GFSR01 — cannot delete gfsresp from ecd — (ecd/rcv/ftamecd)
GFSR02 — cannot read gfsresp in ecd — (ecd/rcv/ftamecd)
GFSR03 — cannot update gfsresp in ecd — (ecd/rcv/ftamecd)
GFSS00 — cannot create gfssec record — (ecd/rcv/ftamecd)
GFSS01 — cannot delete gfssec from ecd — (ecd/rcv/ftamecd)
GFSS02 — cannot read gfssec in ecd — (ecd/rcv/ftamecd)
GFSS03 — cannot update gfssec in ecd — (ecd/rcv/ftamecd)
GR01 — cannot load set — (lla/dml)
GR02 — accessid not in set — (lla/dml)
GR03 — cannot locate rid — (lla/dml)
GR04 — cannot unload set — (lla/dml)
GR05 — bad accessid specified — (lla/dml)
GRB01 — cannot load record or ridblk — (lla/dml)
GRB02 — cannot unload record or ridblk — (lla/dml)
GTCP00 — cannot load cpopt record — (ecd/rcv/ftamaux)
GTCP01 — illegal cmhopt.chcprot value in variant — (ecd/rcv/ftamaux)

GTEA00 — cannot load eaiopt record — (ecd/rcv/ftamaux)

GTEA01 — illegal value in variant — (ecd/rcv/ftamaux)

HASH00 — cannot open file to calculate hashsum — (ecd/rcv/ftamaux)

HASH01 — hashsum calculation error — (ecd/rcv/ftamaux)

HASH02 — cannot read file to calculate hashsum — (ecd/rcv/ftamaux)

HASH03 — cannot store hashsum in file — (ecd/rcv/ftamaux)

HASH04 — cannot allocate buffer — (ecd/rcv/ftamaux)

HASH05 — cannot read startup structure — (ecd/rcv/ftamaux)

HI01 — unable to close stream %o — (lla/sdp)

HI02 — unable to close previous stream %o %s — (lla/sdp)

HI03 — unable to open history file %s — (lla/sdp)

HIAU — audit failed in sdp_hist — (lla/sdp)

HM00 — bad PLACE sanity id — (lla/audit)

HO01 — illegal value of first parameter %d — (lla/sdp)

HO02 — fail to allocate environment — (lla/sdp)

HO03 — fail to allocate managers — (lla/sdp)

HO04 — fail to allocate frame %d — (lla/sdp)

HOAU — audit failed in sdp_house — (lla/sdp)

IA00 — Cannot get incore ecd into address space — (ecd/rcv/ftamaux)

IA00 — cannot malloc space for DATADIR — (lla/audit)

IA00 — cannot malloc space for DATADIR — (lla/dml)

IA01 — Cannot remove incore ecd from address space — (ecd/rcv/ftamaux)

IA01 — cannot malloc space for SPACE — (lla/audit)

IA01 — cannot malloc space for SPACE — (lla/dml)

IA02 — cannot malloc space for ENVIRON — (lla/audit)

IA02 — cannot malloc space for ENVIRON — (Ila/dml)

IA03 — user specified data area not loadf generated — (Ila/audit)

IA03 — user specified data area not loadf generated — (Ila/dml)

IA04 — not all cursors released — (Ila/dml)

IA05 — cannot malloc space for MANAGER — (Ila/audit)

IA05 — cannot malloc space for MANAGER — (Ila/dml)

IA06 — usecount of MANAGER[0] was < 1 and is being reset to 1 — (Ila/dml)

IB01 — insert at root failed — (Ila/am)

IB02 — init node failed — (Ila/am)

IB03 — unable to use new root — (Ila/am)

IB04 — unable to unuse new root — (Ila/am)

IB05 — unable to split old root — (Ila/am)

IC01 — key too long — (Ila/am)

IC02 — unable to allot space for key — (Ila/am)

IC03 — unable to use allotted space for key — (Ila/am)

IC04 — unable to unuse allotted space for key — (Ila/am)

IC05 — unknown access method — (Ila/am)

IDAT00 — no available slots in dbs table — (Ila/rcv)

IDAT01 — unsupported libfile functions — (Ila/rcv)

IDAT02 — no incore database starting address — (Ila/rcv)

IDAT03 — cannot open audit database — (Ila/rcv)

IDAT04 — cannot close audit database — (Ila/rcv)

IDAT05 — audit database seek error — (Ila/rcv)

IDAT06 — audit database write error — (Ila/rcv)

IH01 — unable to use directory — (Ila/am)

- IH02 — unable to unuse directory — (lla/am)
- IH03 — unknown overflow status — (lla/am)
- IH04 — unknown bucket status — (lla/am)
- IH05 — target returned as error — (lla/am)
- II01 — error return from CALC index — (lla/am)
- II02 — index out of bounds — (lla/am)
- II03 — unable to use index slot — (lla/am)
- II04 — unable to unuse index slot — (lla/am)
- II05 — index position taken — (lla/am)
- IMS001 — internal error; transaction not allowed for ucb — (ecd/rcv/integ)
- IMS002 — Cannot retrieve ucb record from database — (ecd/rcv/integ)
- IMS003 — RC/V did not get response from the IMS driver — (ecd/rcv/integ)
- IMS004 — Two processes %d and %d are connected to IMS ports — (ecd/rcv/integ)
- IMS005 — Changes to record %s %d are not accepted by imsdrv — (ecd/rcv/integ)
- IMS006 — ERROR!! Unable to notify IMS DRIVER of ucb change — (ecd/rcv/integ)
- IQ01 — unable to create queue/stack structure — (lla/am)
- IQ02 — unable to use tail id — (lla/am)
- IQ03 — unable to create next structure — (lla/am)
- IQ04 — unable to unuse old tail id — (lla/am)
- IQ05 — unable to use new tail id — (lla/am)
- IQ06 — unable to unuse tail id — (lla/am)
- JRNL00 — jrnlbeg routine aborted — (ecd/rcv/prelinks)
- JRNL01 — jrnlend routine aborted — (ecd/rcv/prelinks)
- JRNL02 — jrnlend routine aborted — (ecd/rcv/ftamaux)
- JRNL03 — temp journal file open error — (ecd/rcv/prelinks)

JRNL04 — journal file open error — (ecd/rcv/prelinks)
JRNL05 — temp journal file open error — (ecd/rcv/prelinks)
JRNL06 — temp journal file size is zero — (ecd/rcv/prelinks)
JRNL07 — journal file concatenation error — (ecd/rcv/prelinks)
JRNL08 — journal file move error — (ecd/rcv/prelinks)
JRNL09 — journal file move error — (ecd/rcv/prelinks)
KC01 — illegal key length — (lla/am)
KC02 — unable to use key id — (lla/am)
KC03 — unable to unuse key id — (lla/am)
KC04 — unable to use indirect tkey — (lla/am)
KC05 — unable to unuse indirect tkey — (lla/am)
KDAT00 — no available slots in dbs table — (lla/rcv)
KDAT01 — unsupported libfile function — (lla/rcv)
KDAT02 — close error — (lla/rcv)
KDAT03 — open error — (lla/rcv)
KDAT04 — seek error — (lla/rcv)
KDAT05 — read error — (lla/rcv)
KDAT06 — read error — (lla/rcv)
KDAT07 — seek error — (lla/rcv)
KDAT08 — write error — (lla/rcv)
KDAT09 — write error — (lla/rcv)
KDAT10 — read error — (lla/rcv)
KDAT11 — seek error — (lla/rcv)
KDU01 — rid not multiple of four — (lla/dml)
KDU02 — sanity check mismatch — (lla/dml)

KDU03 — local rid not multiple of four — (lla/dml)
KDU04 — bad record header — (lla/dml)
KDU05 — itemid out of range — (lla/dml)
KDU06 — indirect rid out of range — (lla/dml)
KGR00 — setid not multiple of long — (lla/dml)
KGR01 — setid greater than maxpage — (lla/dml)
KGR02 — bad set header block — (lla/dml)
KGR03 — user accessid out of bounds — (lla/dml)
KGR04 — bad accessid — (lla/dml)
KGR05 — calc index error — (lla/dml)
KGR06 — user index value out of bound — (lla/dml)
KGR07 — indexid bad — (lla/dml)
KGR08 — unsupported access type specified — (lla/dml)
KGR09 — hash access error — (lla/dml)
KGR10 — queue, stack access error — (lla/dml)
KGR11 — btree access error — (lla/dml)
KH01 — unable to allot for directory — (lla/am)
KH02 — unable to create bucket — (lla/am)
KH03 — unable to use directory — (lla/am)
KH04 — unable to unuse directory — (lla/am)
KL00 — unable to open key file — (ecd/rcv/ftamp)
KLT00 — cannot write out read entry to request file — (ecd/rcv/prelinks)
KLT01 — cannot get specific record — (ecd/rcv/prelinks)
KLT02 — cannot open cursor through symbols or set — (ecd/rcv/prelinks)
KLT03 — cannot build read request — (ecd/rcv/prelinks)

- KLT04 — cannot write out read entry to klist file — (ecd/rcv/prelinks)
- KLT05 — fatal cursor error through symbols or set — (ecd/rcv/prelinks)
- KLT06 — cannot get formname — (ecd/rcv/prelinks)
- KLT07 — cannot get record key — (ecd/rcv/prelinks)
- KLT08 — cannot malloc memory for key — (ecd/rcv/prelinks)
- KRG00 — rid not multiple of four — (lla/dml)
- KRG01 — bad ridblk itemid — (lla/dml)
- KRG02 — bad record header — (lla/dml)
- KRG03 — rid larger than maximum ecd address — (lla/dml)
- KRG04 — ridblk ptr out of range — (lla/dml)
- KRG05 — subsid doesnot match record type — (lla/dml)
- KRG06 — cannot copy variable length record — (lla/dml)
- KRG07 — displ. to pointer error — (lla/dml)
- KRG08 — record id. is negative — (lla/dml)
- KUP00 — user rid not multiple of long — (lla/dml)
- KUP01 — user rid out of range — (lla/dml)
- KUP02 — ridblk ptr not multiple of long — (lla/dml)
- KUP03 — ridblk ptr out of range — (lla/dml)
- KUP04 — bad record header — (lla/dml)
- KUP05 — subsid doesnot match record type — (lla/dml)
- KUP06 — record set_id not multiple of long — (lla/dml)
- KUP07 — bad set member block — (lla/dml)
- KUP08 — dml built with max record size (FULBUFFLEN) < user record — (lla/dml)
- KUP09 — d2p error — (lla/dml)
- KUP10 — dml built with max record size (FULLBUFFLEN) < user record — (lla/dml)

KUP11 — p2d error — (lla/dml)

KUP12 — cannot build larger variable length record — (lla/dml)

KUP13 — d2p error — (lla/dml)

KUP14 — setid not multiple of long — (lla/dml)

KUP15 — bad set header — (lla/dml)

KUP16 — key_breakout error on oldfullrec — (lla/dml)

KUP17 — key_breakout error on fullrec — (lla/dml)

LAPU00 — cannot update aphuser to aphopt link — (ecd/rcv/ftamaux)

LAPU02 — cannot read LOGDEV record — (ecd/rcv/ftamaux)

LAPU03 — cannot update partition and mdct in aphuser — (ecd/rcv/ftamaux)

LAPU04 — associated logdev does not exist — (ecd/rcv/ftamaux)

LAPU04 — associated logdev %s does not exist — (ecd/rcv/ftamaux)

LAPU05 — cannot get mdct record — (ecd/rcv/ftamaux)

LAPU06 — No mdct associated with logdev %s — (ecd/rcv/ftamaux)

LB01 — fail to malloc — (lla/sdp)

LB02 — close error: %d %d — (lla/sdp)

LB03 — creat error: %d %st%o — (lla/sdp)

LB04 — fail to close in creat routine: %d %st%d — (lla/sdp)

LB05 — open error: %d %st%d — (lla/sdp)

LB06 — read error: %d %dt%ot%d — (lla/sdp)

LB07 — lseek error: %d %dt%ldt — (lla/sdp)

LB08 — unlink error: %d %s — (lla/sdp)

LB09 — write error: %d %dt%ot%d — (lla/sdp)

LB10 — size of name exceeds maximum length %st%d — (lla/sdp)

LB11 — fail to allocate space for name — (lla/sdp)

- LC01 — sdp_generate error — (lla/dml)
- LC02 — sdp_house error — (lla/dml)
- LC03 — sdp_connect error — (lla/dml)
- LC04 — cannot alloc dmlhead — (lla/dml)
- LC05 — cannot load dmlhead — (lla/dml)
- LC06 — cannot write dmlhead — (lla/dml)
- LC07 — cannot disconnect dmlhead — (lla/dml)
- LDB01 — Field overflow flagged by ???s in form picture; original value = —
(ecd/rcv/prelinks)
- LDB02 — Dump file supplies more fields than mask allows — (ecd/rcv/prelinks)
- LDB03 — Mask file requires more fields than dump file supplies — (ecd/rcv/prelinks)
- LK00 — cannot open cursor through database — (ecd/rcv/prelinks)
- LK01 — unable to get record info — (ecd/rcv/prelinks)
- LK02 — cannot load record — (ecd/rcv/prelinks)
- LK03 — cannot update record links — (ecd/rcv/prelinks)
- LKAUD00 — cannot update audrec links — (ecd/rcv/ftamaux)
- LKAUD01 — cannot load audrec record — (ecd/rcv/ftamaux)
- LKCI00 — cannot update ciopt links — (ecd/rcv/ftamaux)
- LKCI01 — cannot load ciopt record — (ecd/rcv/ftamaux)
- LKECD00 — cannot update ecdorg links — (ecd/rcv/ftamaux)
- LKECD01 — cannot load ecdorg record — (ecd/rcv/ftamaux)
- LKFTMI00 — cannot update ftaminopt-ftamuser links — (ecd/rcv/ftamaux)
- LKFTMO00 — cannot update ftamopt-objectid link — (ecd/rcv/ftamaux)
- LKFTMO01 — cannot update ftamopt-ftaminopt link — (ecd/rcv/ftamaux)
- LKFTMU00 — cannot update ftamuser-objectid link — (ecd/rcv/ftamaux)

LKGFSA00 — cannot update gfsappl-logdev links — (ecd/rcv/ftamaux)
LKGFSA01 — cannot update gfsappl-gfscom link — (ecd/rcv/ftamaux)
LKGFSA02 — cannot update gfsappl-gfsresp link — (ecd/rcv/ftamaux)
LKGFSA03 — cannot update gfsappl-gfsinit link — (ecd/rcv/ftamaux)
LKGFSA04 — cannot update gfsappl-ftamuser link — (ecd/rcv/ftamaux)
LKGFSC00 — cannot update gfscom-objectid link — (ecd/rcv/ftamaux)
LKGFSC01 — cannot update gfscom-ftamopt link — (ecd/rcv/ftamaux)
LKGFSR00 — cannot update gfsresp-gfssec links — (ecd/rcv/ftamaux)
LKGFSS00 — cannot update gfssec-objectid links — (ecd/rcv/ftamaux)
LKLD00 — cannot update logdev links — (ecd/rcv/ftamaux)
LKLD01 — cannot load logdev record — (ecd/rcv/ftamaux)
LKLD02 — Mdct pointed to by logdev does not exist — (ecd/rcv/ftamaux)
LKLD03 — Could not read mdct logical options for delinking — (ecd/rcv/ftamaux)
LKLD04 — Cannot update mdct record — (ecd/rcv/ftamaux)
LKMDCT00 — cannot update mdct links — (ecd/rcv/ftamaux)
LKMDCT01 — cannot access mdct record — (ecd/rcv/ftamaux)
LKMDCT01 — cannot load mdct record — (ecd/rcv/ftamaux)
LKMDCT02 — cannot load mdct referenced ucb — (ecd/rcv/ftamaux)
LKMDCT03 — cannot update l_minhd field of ucb — (ecd/rcv/ftamaux)
LKRUNA00 — cannot update runa links — (ecd/rcv/ftamaux)
LKSCSD00 — cannot update scsdopt links — (ecd/rcv/ftamaux)
LKUCB00 — cannot update ucb links — (ecd/rcv/ftamaux)
LKUCB01 — cannot load error threshold record — (ecd/rcv/ftamaux)
LKUCB02 — cannot create error counter record — (ecd/rcv/ftamaux)
LKUCB03 — cannot update error counter record — (ecd/rcv/ftamaux)

LKUCB04 — cannot load ucb — (ecd/rcv/ftamaux)

LOG00 — cannot read LOGDEV record — (ecd/rcv/ftamecd)

LOG01 — Mdct specified on LOGDEV form does not exist — (ecd/rcv/ftamecd)

LOG02 — Unable to read logical options structure in mdct — (ecd/rcv/ftamecd)

LOG03 — Mdct specified in LOGDEV record does not exist — (ecd/rcv/ftamecd)

LOG04 — Unable to read logical options structure in mdct — (ecd/rcv/ftamecd)

LOG05 — Getty & chanid currently set for this partition — (ecd/rcv/ftamecd)

LOG06 — Cannot update mdct record — (ecd/rcv/ftamecd)

LOG07 — Cannot create LOGDEV record — (ecd/rcv/ftamecd)

LOG09 — ** WARNING **, special device file inode not removed — (ecd/rcv/ftamecd)

LOG10 — Cannot update logdev record — (ecd/rcv/ftamecd)

LOG12 — Cannot delete logdev record — (ecd/rcv/ftamecd)

LOG13 — cannot open set of APHUSER records — (ecd/rcv/ftamecd)

LOG14 — cannot get record from APHUSER set — (ecd/rcv/ftamecd)

LOG15 — Could not get aphuser record — (ecd/rcv/ftamecd)

LOG16 — Cannot update aphuser record — (ecd/rcv/ftamecd)

LOG17 — Can not get mdct record — (ecd/rcv/ftamecd)

LOG18 — Partition number too large for a non tape device — (ecd/rcv/ftamecd)

LOG19 — Can not have a channel_id for a tape device — (ecd/rcv/ftamecd)

LPPU00 — cannot update pphuser to pphopt link — (ecd/rcv/ftamaux)

LR01 — disk to memory copy failure — (lla/dml)

MD00 — error on initial read of request entry — (ecd/rcv/prelinks)

MD01 — read error on request entry — (ecd/rcv/prelinks)

MD02 — write error on initial dump entry — (ecd/rcv/prelinks)

MD03 — write error on dump entry — (ecd/rcv/prelinks)

MD04 — write error on final dump entry — (ecd/rcv/prelinks)

MD05 — dump error rc = %d, on form type = %s for record: — (ecd/rcv/prelinks)

MDCT00 — cannot create MDCT record — (ecd/rcv/ftamecd)

MDCT01 — cannot delete mdct from ecd — (ecd/rcv/ftamecd)

MDCT02 — cannot read mdct in ecd — (ecd/rcv/ftamecd)

MDCT03 — cannot update mdct in ecd — (ecd/rcv/ftamecd)

MDCT05 — could not delink mdct — (ecd/rcv/ftamecd)

MHDOPT00 — cannot create MHDOPT record — (ecd/rcv/ftamecd)

MHDOPT01 — cannot delete mhdopt from ecd — (ecd/rcv/ftamecd)

MHDOPT02 — cannot read mhdopt in ecd — (ecd/rcv/ftamecd)

MHDOPT03 — cannot update mhdopt in ecd — (ecd/rcv/ftamecd)

MINHD 0 — mdct %s active ucb (%d) already active — (ecd/rcv/ftamaux)

MKD00 — unable to create cursor on pack set — (ecd/rcv/ftamaux)

MKD01 — unable to get next pack in set — (ecd/rcv/ftamaux)

MKD02 — unable to buildread pack — (ecd/rcv/ftamaux)

MKD03 — unable to create cursor on partition set — (ecd/rcv/ftamaux)

MKD04 — unable to get next pack in set — (ecd/rcv/ftamaux)

MKD05 — unable to buildread file partition — (ecd/rcv/ftamaux)

MKD06 — unable to buildread file partition body — (ecd/rcv/ftamaux)

MKD07 — unable to buildread file system — (ecd/rcv/ftamaux)

MKD08 — unable to create cursor on file system body set — (ecd/rcv/ftamaux)

MKD09 — unable to get next fsb in set — (ecd/rcv/ftamaux)

MKD10 — unable to buildread file system body — (ecd/rcv/ftamaux)

MN00 — unable to attach user database — (lla/porter)

MN01 — cannot release user database — (lla/porter)

- MN02 — cannot open output file for writing — (Ila/porter)
- MR01 — cannot erase old record — (Ila/dml)
- MR02 — cannot create new record — (Ila/dml)
- MR03 — cannot load new record — (Ila/dml)
- MR04 — cannot write new record — (Ila/dml)
- MR05 — cannot load rid block — (Ila/dml)
- MR06 — cannot rewrite updated rid block — (Ila/dml)
- MSG01 — DGEN completed. The size of the flat file is %ld bytes which requires that YOU have allocated at least %ld segment(s) for the database. — (ecd/dgen)
- NB01 — unable to use current node — (Ila/am)
- NB01 — unable to use current node — (Ila/porter)
- NB02 — unknown search type — (Ila/am)
- NB02 — unknown search type — (Ila/porter)
- NB03 — unable to unuse current node — (Ila/am)
- NB03 — unable to unuse current node — (Ila/porter)
- NB04 — unable to copy key into cursor — (Ila/porter)
- NC01 — time stamp out of date — (Ila/am)
- NC01 — unable to use bucket — (Ila/am)
- NC02 — bucket search fails — (Ila/am)
- NC02 — unknown access method — (Ila/am)
- NC03 — unable to create overflow — (Ila/am)
- NC04 — bckt_full returns unknown value — (Ila/am)
- NC05 — unable to unuse bucket — (Ila/am)
- NH01 — unable to use bucket (PHY) — (Ila/am)
- NH01 — unable to use bucket (PHY) — (Ila/porter)

NH02 — unable to unuse bucket (PHY) — (lla/am)
NH02 — unable to unuse bucket (PHY) — (lla/porter)
NH03 — unable to use bucket (EQ) — (lla/am)
NH03 — unable to use bucket (EQ) — (lla/porter)
NH04 — unable to unuse bucket (EQ) — (lla/am)
NH04 — unable to unuse bucket (EQ) — (lla/porter)
NH05 — illegal search type — (lla/am)
NH05 — illegal search type — (lla/porter)
NH06 — unknown search type — (lla/am)
NH06 — unknown search type — (lla/porter)
NH07 — illegal return from bckt_look — (lla/am)
NH07 — unable to copy key into cursor — (lla/porter)
NI01 — unknown search type — (lla/am)
NI02 — unable to use index slot — (lla/am)
NI03 — unable to unuse index slot — (lla/am)
NO00 — internal error; called for other than update — (ecd/rcv/integ)
NO01 — cannot retrieve ucb — (ecd/rcv/integ)
NO02 — Two processes %d and %d are connected to the ims ports — (ecd/rcv/integ)
NPH00 — cannot create NPH record — (ecd/rcv/ftamecd)
NPH01 — cannot delete NPH from ecd — (ecd/rcv/ftamecd)
NPH02 — cannot read NPH in ecd — (ecd/rcv/ftamecd)
NPH03 — cannot update NPH in ecd — (ecd/rcv/ftamecd)
NPH04 — cannot break old links — (ecd/rcv/ftamecd)
NQ01 — unable to use queue/stack structure — (lla/am)
NQ02 — unknown search type — (lla/am)

NQ03 — unable to unuse queue/stack structure — (lla/am)

NS01 — cannot locate record — (lla/dml)

NS02 — cannot load record — (lla/dml)

NSR01 — cannot load set — (lla/dml)

NSR02 — next controller error — (lla/dml)

NSR03 — cannot unload set — (lla/dml)

NSR04 — cannot load rid block — (lla/dml)

NSR05 — cannot unload rid block — (lla/dml)

NSR06 — cannot unload rid block — (lla/dml)

OB01 — unable to use node — (lla/am)

OB02 — unable to unuse node — (lla/am)

OB03 — unable to reuse node — (lla/am)

OB04 — unable to reunuse node — (lla/am)

OB05 — unknown search type — (lla/am)

OB06 — unable to use first leaf node — (lla/am)

OB07 — unable to unuse first leaf node — (lla/am)

OBJID00 — cannot create OBJECTID record — (ecd/rcv/ftamecd)

OBJID01 — cannot delete OBJECTID from ecd — (ecd/rcv/ftamecd)

OBJID02 — cannot read OBJECTID in ecd — (ecd/rcv/ftamecd)

OBJID03 — cannot update OBJECTID in ecd — (ecd/rcv/ftamecd)

OC01 — unable to use bucket — (lla/am)

OC01 — unknown access method — (lla/am)

OC02 — unable to create overflow — (lla/am)

OC03 — hash value not found — (lla/am)

OC04 — unable to use overflow — (lla/am)

OC05 — unable to unuse overflow — (lla/am)
OC06 — unable to unuse bucket — (lla/am)
OD00 — trbegin must be issued — (ecd/rcv/prelinks)
OD01 — bad dbtype — (ecd/rcv/prelinks)
OD02 — trend must be issued — (ecd/rcv/prelinks)
OI00 — specified ordinal doesnot match any record — (lla/dml)
OI01 — bad search type — (lla/am)
OP01 — Cannot allocate space for cursor — (lla/dml)
OPF00 — cannot open print file — (ecd/rcv/links)
OPF01 — cannot open echo file — (ecd/rcv/links)
OPF02 — cannot open journal file — (ecd/rcv/links)
OQ01 — unable to use queue/stack structure (HEAD) — (lla/am)
OQ02 — unable to unuse queue/stack structure (HEAD) — (lla/am)
OQ03 — unable to use queue/stack structure (TAIL) — (lla/am)
OQ04 — unable to unuse queue/stack structure (TAIL) — (lla/am)
OQ05 — unknown search type — (lla/am)
OR00 — cannot load next frame to be scanned — (lla/audit)
OR01 — cannot load possible rid block — (lla/audit)
OR02 — cannot unload rid block — (lla/audit)
OR03 — cannot load possible set header — (lla/audit)
OR04 — cannot unload set header — (lla/audit)
OR05 — cannot load possible set member block — (lla/audit)
OR06 — cannot unload set member block — (lla/audit)
OR07 — cannot load rid block for correction — (lla/audit)
OR08 — cannot write updated rid block back — (lla/audit)

- OR09 — cannot unload current frame being scanned — (Ila/audit)
- ORPAUD00 — resetting ridblk slot at 0x%lx to 0x%lx from 0x%lx — (Ila/audit)
- OS01 — cannot allocate space for cursor — (Ila/dml)
- OS01 — unable to use bucket — (Ila/am)
- OS02 — cannot load set — (Ila/dml)
- OS02 — hashval not found — (Ila/am)
- OS03 — accessid out of bounds — (Ila/dml)
- OS03 — unable to use source bucket — (Ila/am)
- OS04 — bad accessid specified — (Ila/dml)
- OS04 — count not consistent for split — (Ila/am)
- OS05 — open controller error — (Ila/dml)
- OS05 — unable to create overflow — (Ila/am)
- OS06 — cannot unload set — (Ila/dml)
- OS06 — unable to use destination bucket — (Ila/am)
- OS07 — bad search option for UNIVERSAL set — (Ila/dml)
- OS07 — unable to unuse bucket — (Ila/am)
- OS08 — unable to unuse source bucket — (Ila/am)
- OS09 — unable to unuse destination bucket — (Ila/am)
- OT01 — unable to use target — (Ila/am)
- OT02 — unable to unuse target — (Ila/am)
- PACK 0 — pack still contains some active file partitions or systems that must be deleted —
(ecd/rcv/ftamsg)
- PACK00 — cannot create PACK record — (ecd/rcv/ftamsg)
- PACK01 — cannot delete pack from sg — (ecd/rcv/ftamsg)
- PACK02 — cannot read pack in sg — (ecd/rcv/ftamsg)

PACK03 — cannot update pack in sg — (ecd/rcv/ftamsg)

PACK04 — cannot reference partition set — (ecd/rcv/ftamsg)

PACK05 — cannot remove set of partitions — (ecd/rcv/ftamsg)

PERRC00 — cannot retrieve ucb record — (ecd/rcv/ftamaux)

PO01 — error return from framesearch — (lla/sdp)

PO02 — unable to push remainder of popped area — (lla/sdp)

PPHO01 — cannot create PPHOPT record — (ecd/rcv/ftamecd)

PPHO02 — cannot delete PPHOPT record — (ecd/rcv/ftamecd)

PPHO03 — cannot read PPHOPT record — (ecd/rcv/ftamecd)

PPHO04 — cannot update PPHOPT record — (ecd/rcv/ftamecd)

PPHO05 — cannot open set of PPHOPT records — (ecd/rcv/ftamecd)

PPHO06 — cannot get record from PPHOPT set — (ecd/rcv/ftamecd)

PPHU01 — cannot create PPHUSER record — (ecd/rcv/ftamecd)

PPHU02 — cannot delete PPHUSER record — (ecd/rcv/ftamecd)

PPHU03 — cannot read PPHUSER record — (ecd/rcv/ftamecd)

PPHU04 — cannot update PPHUSER record — (ecd/rcv/ftamecd)

PPHU05 — cannot get PPHUSER record — (ecd/rcv/ftamecd)

PPHU06 — cannot delink the old PPHUSER link — (ecd/rcv/ftamecd)

PPHU07 — cannot open set of PPHUSER records — (ecd/rcv/ftamecd)

PPHU08 — loc_ssap_sel must not be changed with an update — (ecd/rcv/ftamecd)

PRCSS00 — cannot create PROCESS record — (ecd/rcv/ftamsg)

PRCSS01 — cannot delete process from sg — (ecd/rcv/ftamsg)

PRCSS02 — cannot read process in sg — (ecd/rcv/ftamsg)

PRCSS03 — cannot update process in sg — (ecd/rcv/ftamsg)

PRCSS04 — cannot open process sequence — (ecd/rcv/ftamsg)

- PRCSS05 — cannot get next process record — (ecd/rcv/ftamsg)
- PRL00 — cannot load mask for formtype: — (ecd/rcv/prelinks)
- PRL01 — cannot print form instance without having preceding form code known — (ecd/rcv/prelinks)
- PRL02 — key greater than buffer — (ecd/rcv/prelinks)
- PRL03 — cannot retrieve formtype = %s, rc = %d, record: — (ecd/rcv/prelinks)
- PROCESS 0 — Internal error, specified form instance missing — (ecd/rcv/ftamsg)
- PSCOPT00 — cannot retrieve scsdopt record — (ecd/rcv/ftamaux)
- PSCOPT01 — cannot build scsdbody request — (ecd/rcv/ftamaux)
- PU00 — lla_nsgrid error in audit ecd — (ecd/rcv/ftamaux)
- PU00 — unable to split redeemed space into 2 pieces — (lla/sdp)
- PU01 — error return from stk2search — (lla/sdp)
- PU01 — lla_rinfo error in audit ecd — (ecd/rcv/ftamaux)
- PU02 — error return from fault — (lla/sdp)
- PU02 — lla_rdget error on affected record in audit ecd — (ecd/rcv/ftamaux)
- PU03 — lla_uprid error on affected record in audit ecd — (ecd/rcv/ftamaux)
- RBL00 — cannot find previous rid block processed — (lla/audit)
- RBL01 — cannot load current rid block — (lla/audit)
- RBL02 — cannot load next rid block — (lla/audit)
- RBL03 — cannot find next rid block — (lla/audit)
- RBL04 — cannot unload next rid block — (lla/audit)
- RBL05 — cannot find next rid block — (lla/audit)
- RBL06 — cannot unload current rid block — (lla/audit)
- RBLAUD00 — rid block sanity id at 0x%lx found to have incorrect value 0x%lx — (lla/audit)
- RBLAUD01 — previous link field in rid block at 0x%lx being reset from 0x%lx to 0x%lx — (lla/audit)

RBLAUD02 — next link field in rid block at 0x%lx being reset from 0x%lx to 0x%lx — (lla/audit)

RC00 — Unsupported request mode for this form code — (ecd/rcv/prelinks)

RC01 — Form instance has outstanding references — (ecd/rcv/prelinks)

RC02 — Key of requested form exists on another form type — (ecd/rcv/prelinks)

RCT01 — cannot get record information — (ecd/rcv/prelinks)

RCT02 — cannot get set owner count — (ecd/rcv/prelinks)

RCV01 — cannot close database — (ecd/rcv/links)

RCV02 — could not set globals — (ecd/rcv/links)

RE01 — space not connected %o — (lla/sdp)

RE02 — no permission to redeem %o — (lla/sdp)

RE03 — id out of range %lot%lo — (lla/sdp)

RE04 — id not aligned %ot%lo — (lla/sdp)

RE05 — fail to push %ot%lo — (lla/sdp)

REAU — audit failed in sdp_redeem — (lla/sdp)

RF00 — sdp_refresh error — (lla/dml)

RF01 — error return from readpage %o — (lla/sdp)

RF02 — frame with change status encountered by sdp_refresh %o — (lla/sdp)

RF03 — unknown ref/chng flag %ot%d — (lla/sdp)

RF04 — error rereading SPACE control information %o — (lla/sdp)

RF05 — space not connected %o — (lla/sdp)

RF06 — unable to ALLOC new SPACE structure — (lla/sdp)

RFAU — audit failed in sdp_refresh — (lla/sdp)

RG00 — cannot load record — (lla/dml)

RG01 — cannot unload ridblk entry — (lla/dml)

- RG02 — cannot load actual record — (lla/dml)
- RG03 — bad record sanity id — (lla/dml)
- RG04 — subsid doesnot match record type — (lla/dml)
- RG05 — cannot load entire variable length record — (lla/dml)
- RG06 — displacement to pointer error — (lla/dml)
- RG07 — cannot load entire large record — (lla/dml)
- RG08 — cannot unload record — (lla/dml)
- RG09 — record id. is negative — (lla/dml)
- RM01 — cannot open sequence — (lla/dml)
- RM02 — cannot get next recrod in set — (lla/dml)
- RM03 — cannot delete record — (lla/dml)
- RM04 — cannot load set — (lla/dml)
- RM05 — cannot remove clustered extents — (lla/dml)
- RM06 — cannot unload set — (lla/dml)
- RM07 — cannot redeem set instance — (lla/dml)
- RM08 — cannot destory access structure — (lla/dml)
- RM09 — cannot change dmlhead status — (lla/dml)
- RM10 — cannot load previous sethead — (lla/dml)
- RM11 — cannot write previous sethead — (lla/dml)
- RM12 — cannot change dmlhead status — (lla/dml)
- RM13 — cannot load next sethead — (lla/dml)
- RM14 — cannot write next sethead — (lla/dml)
- RP01 — attempt to read id beyond maximum allotment %lot%lo — (lla/sdp)
- RP02 — attempt to read beyond file configuration %dt%d — (lla/sdp)
- RP03 — fail to build file name %s%d — (lla/sdp)

- RP04 — error return from open %s — (lla/sdp)
- RP05 — fail to seek %s — (lla/sdp)
- RP06 — fail to read %s — (lla/sdp)
- RP07 — unexpected number of bytes read %d — (lla/sdp)
- RS00 — cannot attach audit copy of ecd — (ecd/rcv/ftamaux)
- RS00 — cannot open cursor through all sets — (lla/porter)
- RS01 — cannot open universal set in audit ecd — (ecd/rcv/ftamaux)
- RS01 — fail to allocate for header file %s — (lla/sdp)
- RS01 — lla_seqset error — (lla/porter)
- RS02 — cannot close audit ecd after resetting required fields — (ecd/rcv/ftamaux)
- RS02 — error return from buildname %s — (lla/sdp)
- RS02 — unable to load next set header block — (lla/porter)
- RS03 — cannot calculate and store hashsum in the audit ecd — (ecd/rcv/ftamaux)
- RS03 — cannot unload current set header block — (lla/porter)
- RS03 — fail to open header file %s — (lla/sdp)
- RS04 — cannot open cursor through set — (lla/porter)
- RS04 — fail to position header file %s — (lla/sdp)
- RS05 — error dumping records within a set — (lla/porter)
- RS05 — fail to read header file %s — (lla/sdp)
- RS06 — cannot open cursor through UNIVERSAL set — (lla/porter)
- RS06 — fail to allocate for file information %s — (lla/sdp)
- RS07 — error dumping records in UNIVERSAL set — (lla/porter)
- RS07 — fail to position header file for file information %s — (lla/sdp)
- RS08 — fail to read header file for file information %s — (lla/sdp)
- RS08 — fprintf error writing final record info. — (lla/porter)

- RS09 — fail to allocate for stack information %s — (lla/sdp)
- RS10 — fail to position header file for stack information %s — (lla/sdp)
- RS11 — fail to read stack information %s — (lla/sdp)
- RS12 — fail to close header file %s — (lla/sdp)
- RS13 — error returned from setSPACE %s — (lla/sdp)
- RSAUD00 — cannot access audrec record — (ecd/rcv/integ)
- RSAUD01 — cannot reset audrec values — (ecd/rcv/integ)
- RSAUD02 — cannot update audrec record — (ecd/rcv/integ)
- RSAUDI00 — cannot access audinst record — (ecd/rcv/integ)
- RSAUDI01 — cannot reset audinst values — (ecd/rcv/integ)
- RSAUDI02 — cannot update audinst record — (ecd/rcv/integ)
- RSECD00 — cannot access ecdorg record — (ecd/rcv/integ)
- RSECD01 — cannot reset ecdorg values — (ecd/rcv/integ)
- RSECD02 — cannot update ecdorg record — (ecd/rcv/integ)
- RSEERR00 — cannot access error count record — (ecd/rcv/integ)
- RSEERR01 — cannot reset error count values — (ecd/rcv/integ)
- RSEERR02 — cannot update error count record — (ecd/rcv/integ)
- RSSCSC01 — cannot reset scsdmap values — (ecd/rcv/integ)
- RSSCSD00 — cannot access scsdmap record — (ecd/rcv/integ)
- RSSCSD02 — cannot update scsdmap record — (ecd/rcv/integ)
- RSSIM00 — cannot access simcntl record — (ecd/rcv/integ)
- RSSIM01 — cannot reset simcntl values — (ecd/rcv/integ)
- RSSIM02 — cannot update simcntl record — (ecd/rcv/integ)
- RST01 — cannot load record — (lla/dml)
- RST02 — record member of only UNIVERSAL set — (lla/dml)

RST03 — cannot load set member block — (lla/dml)
RST04 — cannot unload set member block — (lla/dml)
RST05 — cannot unload record — (lla/dml)
RSUCB00 — cannot access ucb record — (ecd/rcv/integ)
RSUCB01 — cannot reset ucb values — (ecd/rcv/integ)
RSUCB02 — cannot update ucb record — (ecd/rcv/integ)
RT00 — unable to set write permissions on DML head structure — (lla/dml)
RT01 — cannot load record — (lla/dml)
RT02 — cannot unload record — (lla/dml)
RUNA00 — cannot create RUNA record — (ecd/rcv/ftamecd)
RUNA01 — cannot delete RUNA from ecd — (ecd/rcv/ftamecd)
RUNA02 — cannot read RUNA in ecd — (ecd/rcv/ftamecd)
RUNA03 — cannot update RUNA in ecd — (ecd/rcv/ftamecd)
RUNA04 — cannot break old links — (ecd/rcv/ftamecd)
RUNALK01 — cannot load runa record — (ecd/rcv/ftamaux)
RV00 — recover unable to release itemid — (lla/dml)
RV200 — recov2 unable to release first itemid — (lla/dml)
RV201 — recov2 unable to release second itemid — (lla/dml)
S201 — stack sizes incorrectly configured — (lla/sdp)
SC01 — open_hash failed — (lla/am)
SC01 — user specified set_type not a DDLSETTYPE pointer — (lla/dml)
SC02 — open_quest failed — (lla/am)
SC02 — requested sethead will not fit on creation physical page — (lla/dml)
SC03 — cannot allocate space for SETHEAD — (lla/dml)
SC03 — open_btree failed — (lla/am)

- SC04 — cannot load initialized SETHEAD — (lla/dml)
- SC04 — unknown access method — (lla/am)
- SC05 — cannot create access structure — (lla/dml)
- SC06 — cannot write initialized SETHEAD — (lla/dml)
- SC07 — cannot load previous set header — (lla/dml)
- SC08 — cannot write previous set header — (lla/dml)
- SC09 — cannot change dmlhead status — (lla/dml)
- SC10 — DML primitives built with maximum key size (KEYMAX) less than actual key — (lla/dml)
- SCBA00 — cannot retrieve scsdopt record — (ecd/rcv/ftamecd)
- SCBA01 — cannot update scsdopt record — (ecd/rcv/ftamecd)
- SCBA02 — cannot break old link — (ecd/rcv/ftamecd)
- SCBA03 — scsdbody %s duplicates earlier sc_list entry — (ecd/rcv/ftamecd)
- SCBF00 — logical group table full — (ecd/rcv/ftamecd)
- SCBF01 — scan point table not big enough — (ecd/rcv/ftamecd)
- SCBF02 — distributor point table not big enough — (ecd/rcv/ftamecd)
- SCMAP00 — cannot retrieve scsdopt record — (ecd/rcv/ftamecd)
- SCOPT00 — cannot create scsdopt record — (ecd/rcv/ftamecd)
- SCOPT01 — cannot retrieve scsdopt record — (ecd/rcv/ftamecd)
- SCOPT02 — cannot update scsdopt record — (ecd/rcv/ftamecd)
- SCOPT03 — cannot delete scsdopt record — (ecd/rcv/ftamecd)
- SCOPT04 — cannot break old links — (ecd/rcv/ftamecd)
- SCSDMAP00 — record contains more logical groups than form — (ecd/rcv/ftamecd)
- SDF00 — cannot get logdev record — (ecd/rcv/integ)
- SDF01 — cannot remove special device file — (ecd/rcv/integ)

SDF02 — mknod failed — (ecd/rcv/integ)

SDF03 — chmod failed — (ecd/rcv/integ)

SDF04 — chown failed — (ecd/rcv/integ)

SDF05 — internal error: called for invalid transaction — (ecd/rcv/integ)

SDF06 — invalid special device file type — (ecd/rcv/integ)

SDP00 — unable to reset previous stack entry with bad next id — (lla/audit)

SDP01 — unable to load current stack entry — (lla/audit)

SDP02 — unable to reset previous stack entry — (lla/audit)

SDP03 — unable to unload current stack entry — (lla/audit)

SDP04 — unable to load stack entry to be reset — (lla/audit)

SDP05 — unable to unload revised stack entry — (lla/audit)

SDP06 — unable to load previous stack entry — (lla/audit)

SDP07 — unable to unload previous stack entry — (lla/audit)

SDP08 — unable to load previous 4 byte stack entry — (lla/audit)

SDP09 — unable to unload previous 4 byte stack entry — (lla/audit)

SDP10 — unable to reset previous 4 byte stack entry — (lla/audit)

SDP11 — unable to load next 4 byte stack entry — (lla/audit)

SDP12 — unable to unload next 4 byte stack entry — (lla/audit)

SDP13 — unable to load previous 4 byte entry for correction — (lla/audit)

SDP14 — unable to unload corrected previous 4 byte entry — (lla/audit)

SDPAUD00 — stack entry 0x%lx next id found to have incorrect value 0x%lx — (lla/audit)

SDPAUD01 — stack entry 0x%lx on %d list size %ld out of range — (lla/audit)

SDPAUD02 — resetting depth of %d control stack from %ld to %ld — (lla/audit)

SDPAUD03 — resetting next id of %d control stack from 0x%lx to 0x%lx — (lla/audit)

SDPAUD04 — stack entry 0x%lx next id being reset from 0x%lx to 0x%lx — (lla/audit)

- SDPAUD05 — resetting next id of %d control stack from 0x%lx to 0x%lx — (lla/audit)
- SDPAUD06 — four byte stack entry 0x%lx top found to have incorrect value 0x%lx — (lla/audit)
- SDPAUD07 — resetting top itemid of 4 byte stack from 0x%lx to 0x%lx — (lla/audit)
- SDPAUD08 — resetting depth of 4 byte stack from %ld to %ld — (lla/audit)
- SDPAUD09 — resetting top of 4 byte stack from 0x%lx to 0x%lx — (lla/audit)
- SDPAUD10 — 4 byte stack entry 0x%lx next id being reset from 0x%lx to 0x%lx — (lla/audit)
- SEM01 — cannot set or clear semaphore in ecdroot — (ecd/rcv/ftamaux)
- SEM02 — cannot flush ecd with lla_flush — (ecd/rcv/ftamaux)
- SEM02 — cannot flush sg with lla_flush — (ecd/rcv/ftamaux)
- SET00 — cannot find previous set header processed — (lla/audit)
- SET01 — cannot load current set header — (lla/audit)
- SET02 — cannot load next set header — (lla/audit)
- SET03 — cannot find next set header — (lla/audit)
- SET04 — cannot unload next set header — (lla/audit)
- SET05 — cannot find next set header — (lla/audit)
- SET06 — cannot unload current set header — (lla/audit)
- SETAUD00 — set header sanity id at 0x%lx found to have incorrect value 0x%lx — (lla/audit)
- SETAUD01 — previous link field in set header at 0x%lx being reset from 0x%lx to 0x%lx — (lla/audit)
- SETAUD02 — next link field in set header at 0x%lx being reset from 0x%lx to 0x%lx — (lla/audit)
- SGEN200 — cannot create sgen2 record — (ecd/rcv/ftamsg)
- SGEN201 — cannot read sgen2 in sg — (ecd/rcv/ftamsg)
- SGEN202 — cannot update sgen2 in sg — (ecd/rcv/ftamsg)

- SHARE 0 — Internal error, specified form instance missing — (ecd/rcv/ftamsg)
- SHARE00 — cannot create SHARE record — (ecd/rcv/ftamsg)
- SHARE01 — cannot delete share from sg — (ecd/rcv/ftamsg)
- SHARE02 — cannot read share in sg — (ecd/rcv/ftamsg)
- SHARE03 — cannot open share sequence — (ecd/rcv/ftamsg)
- SHARE04 — cannot get next share record — (ecd/rcv/ftamsg)
- SHD00 — unable to load set header — (lla/audit)
- SHD01 — unable to unload set header — (lla/audit)
- SHDAUD00 — set 0x%lx ordinal in both header & accessid bad — (lla/audit)
- SHDAUD01 — set 0x%lx ordinal being reset from 0x%x to 0x%x — (lla/audit)
- SHDAUD02 — set 0x%lx number of access methods being reset from %d to %d — (lla/audit)
- SHDAUD03 — set 0x%lx cluster flag being reset from 0x%x to 0x%x — (lla/audit)
- SHDAUD04 — set 0x%lx, access method %d, accessid being reset from 0x%lx to 0x%lx — (lla/audit)
- SHDAUD05 — set 0x%lx, access method %d, pagesize being reset from 0x%x to 0x%x — (lla/audit)
- SHDAUD06 — set 0x%lx, access method %d, keylength being reset from %d to %ld — (lla/audit)
- SHL00 — 'hlmode' invalid field on 'rcvparams' form — (ecd/rcv/fiop)
- SHL01 — cannot initialize 'rcvparams' form — (ecd/rcv/fiop)
- SI01 — cannot load set — (lla/dml)
- SI01 — error return from calc index — (lla/am)
- SI02 — cannot unload set — (lla/dml)
- SI02 — index out of bounds — (lla/am)
- SI03 — unable to use index slot — (lla/am)
- SI04 — unable to unuse index slot — (lla/am)

SIG00 — RC/V killed by termination signal %d* — (ecd/rcv/links)

SIM00 — cannot insert SIM record — (ecd/rcv/ftamecd)

SIM01 — cannot read SIM record — (ecd/rcv/ftamecd)

SIM02 — cannot update SIM record — (ecd/rcv/ftamecd)

SN01 — cannot load record — (lla/dml)

SN02 — cannot load set header — (lla/dml)

SN03 — cannot unload set header — (lla/dml)

SN04 — cannot unload record — (lla/dml)

SPD00 — cannot open request file or create dump file — (ecd/rcv/prelinks)

SPD01 — dump cannot be done from the incore database — (ecd/rcv/prelinks)

SPHO01 — cannot create SPHOPT record — (ecd/rcv/ftamecd)

SPHO02 — cannot read SPHOPT record — (ecd/rcv/ftamecd)

SPHO03 — cannot update SPHOPT record — (ecd/rcv/ftamecd)

SPL00 — cannot open load file — (ecd/rcv/prelinks)

SPL01 — cannot fopen error file — (ecd/rcv/prelinks)

SPL02 — skeleton database previously modified — (ecd/rcv/prelinks)

SPL03 — must be in sysgen mode to do a dbload — (ecd/rcv/prelinks)

SPLI00 — cannot insert SPLI record — (ecd/rcv/ftamecd)

SPLI01 — cannot read SPLI record — (ecd/rcv/ftamecd)

SPLI02 — cannot update SPLI record — (ecd/rcv/ftamecd)

SPP00 — cannot open key file — (ecd/rcv/prelinks)

SPP01 — cannot open print file — (ecd/rcv/prelinks)

SS00 — cannot open cursor through all sets — (lla/porter)

* See SYSERR-APP in the System Error Codes Output Appendix for the termination signal number definition.

- SS01 — Cannot load set — (lla/dml)
- SS01 — lla_seqset error — (lla/porter)
- SS02 — Cannot not unload set — (lla/dml)
- SS02 — bad set header block — (lla/porter)
- SS03 — bad access type in set access block — (lla/porter)
- SS04 — cannot open physical cursor through set — (lla/porter)
- SS05 — cannot release set header block — (lla/porter)
- SS06 — lla_nsqid error — (lla/porter)
- SS07 — fprintf error writing set member entry — (lla/porter)
- SS08 — unable to get record info. — (lla/porter)
- SS09 — unable to dump non-recorded based key — (lla/porter)
- SS10 — fprintf error writing final set info. — (lla/porter)
- SS11 — cannot get set name given set ordinal — (lla/porter)
- SS12 — cannot unload set header block — (lla/porter)
- ST01 — attempt to take statistics on an unhoued enviroment %o — (lla/sdp)
- STAU — audit failed in sdp_stats — (lla/sdp)
- STRAUD00 — incore database sanity id found to have incorrect value 0x%lx — (lla/audit)
- STRAUD01 — offset to sdp SPACE structure being reset from 0x%lx to 0x%x — (lla/audit)
- STRAUD02 — offset to user data section being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD03 — user rootid in DMLHEAD(=0x%lx) was reinitialized — (lla/audit)
- STRAUD04 — startup database size being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD05 — space maxpage being reset from 0x%lx to 0x%lx — (lla/audit)

- STRAUD06 — number of SDP space allocation stacks being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD07 — pointer to SDP stacks being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD08 — SDP space pointer to environ being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD09 — ENVIRON manager pointer being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD10 — SDP manager frame pointer being reset from 0x%lx to 0x%lx — (lla/audit)
- STRAUD11 — SDP manager page id being reset from 0x%lx to 0x0 — (lla/audit)
- STRAUD12 — SDP manager use count being reset from 0x%lx to 0x1 — (lla/audit)
- STRAUD13 — SDP manager next index being reset from 0x%lx to 0x1 — (lla/audit)
- STRAUD14 — SDP manager previous index being reset from 0x%lx to 0x1 — (lla/audit)
- STRAUD15 — SDP manager page id being reset from 0x%lx to IDNULL — (lla/audit)
- STRAUD16 — SDP manager use count being reset from 0x%lx to 0x0 — (lla/audit)
- STRAUD17 — SDP manager next index being reset from 0x%lx to 0x0 — (lla/audit)
- STRAUD18 — SDP manager previous index being reset from 0x%lx to 0x0 — (lla/audit)
- SY01 — error return from sdp_flush %o — (lla/sdp)
- SY02 — error return from sdp_refresh %o — (lla/sdp)
- SYAU — audit failed in sdp_sync — (lla/sdp)
- SYM00 — cannot get symbol record — (ecd/rcv/prelinks)
- SYM01 — cannot create symbol record — (ecd/rcv/prelinks)
- SYM02 — cannot update symbol record — (ecd/rcv/prelinks)
- SYM03 — cannot delete symbol record — (ecd/rcv/prelinks)
- SYM04 — illegal link value — (ecd/rcv/prelinks)

SYM05 — symbol already exists in the database — (ecd/rcv/prelinks)

SYM06 — symbol not found — (ecd/rcv/prelinks)

SYM07 — cannot open set of symbol records — (ecd/rcv/prelinks)

SYM08 — cannot get symbol record — (ecd/rcv/prelinks)

SYM09 — cannot malloc — (ecd/rcv/prelinks)

SYM10 — cannot delete symbol record — (ecd/rcv/prelinks)

SYNC00 — cannot close incore database — (ecd/rcv/ftamaux)

SYNC01 — cannot attach boot ecd — (ecd/rcv/ftamaux)

SYNC02 — cannot copy ecd boot file to ecd delta file — (ecd/rcv/ftamaux)

SYNC03 — cannot copy ecd delta file to audit ecd — (ecd/rcv/ftamaux)

SYNC04 — cannot clear semaphore in boot ecd — (ecd/rcv/ftamaux)

SYNC05 — cannot close boot ecd — (ecd/rcv/ftamaux)

SYNC06 — cannot attach incore ecd — (ecd/rcv/ftamaux)

SYNC07 — cannot clear semaphore in incore ecd — (ecd/rcv/ftamaux)

SYNC08 — cannot replace audit ecd — (ecd/rcv/ftamaux)

SYNC09 — cannot attach audit ecd — (ecd/rcv/ftamaux)

SYNC10 — cannot clear semaphore in audit ecd — (ecd/rcv/ftamaux)

SYNC11 — cannot close audit ecd — (ecd/rcv/ftamaux)

TB00 — unable to close database — (ecd/rcv/ftamaux)

TB01 — unable to reopen database with transaction blocks —
(ecd/rcv/ftamaux)

TBAB0 — no associated transaction block defined — (lla/tb)

TBAB1 — unable to restore SDP buffer to original state — (lla/tb)

TBB0 — error seeking to start of delta control list — (lla/tb)

TBB1 — error writing start of delta control list — (lla/tb)

- TBB2 — changes indicated on other than start of Dlt_cntl list — (I/a/tb)
- TBB3 — error seeking to start of Dlt_cntl list — (I/a/tb)
- TBB4 — error reading start of Dlt_cntl list — (I/a/tb)
- TBC0 — unable to close file — (I/a/tb)
- TBCM00 — unable to flush all new or changed pages out — (I/a/tb)
- TBCM01 — cannot find associated transaction block — (I/a/tb)
- TBCM02 — error doing special control file processing — (I/a/tb)
- TBCM03 — error doing special data file processing — (I/a/tb)
- TBCM04 — cannot ALLOC scratch buffer — (I/a/tb)
- TBCM05 — cannot open file with changes on it — (I/a/tb)
- TBCM06 — error seeking in delta file — (I/a/tb)
- TBCM07 — error reading from delta file — (I/a/tb)
- TBCM08 — data read must be multiple of a page — (I/a/tb)
- TBCM09 — error seeking in underlying database file — (I/a/tb)
- TBCM10 — error writing changes to underlying file — (I/a/tb)
- TBCM11 — error seeking next Dlt_cntl structure — (I/a/tb)
- TBCM12 — error reading next Dlt_cntl structure — (I/a/tb)
- TBCM13 — error closing SDP control file — (I/a/tb)
- TBCM14 — error closing data file — (I/a/tb)
- TBCP0 — error closing associated datafile — (I/a/tb)
- TBCP1 — error closing associated SDP control file — (I/a/tb)
- TBCP2 — error closing delta file — (I/a/tb)
- TBCP3 — error unlinking delta file — (I/a/tb)
- TBCR0 — creation of SDP control file not currently supported — (I/a/tb)
- TBCR1 — number of available file slots exceeded — (I/a/tb)

- TBCR2 — error duplicating data file name — (I/a/tb)
- TBG0 — error seeking to start of Dlt_cntl list — (I/a/tb)
- TBG1 — error writing start of Dlt_cntl list — (I/a/tb)
- TBG2 — error seeking to next Dlt_cntl structure — (I/a/tb)
- TBG3 — error reading next Dlt_cntl structure — (I/a/tb)
- TBM0 — unable to ALLOC memory to duplicate string — (I/a/tb)
- TBO00 — first open must be for a SDP control file — (I/a/tb)
- TBO01 — unable to ALLOC new TB control structure — (I/a/tb)
- TBO02 — cannot open SDP control file — (I/a/tb)
- TBO03 — bad delta file path name — (I/a/tb)
- TBO04 — cannot create delta file — (I/a/tb)
- TBO05 — cannot open delta file — (I/a/tb)
- TBO06 — cannot duplicate delta file name — (I/a/tb)
- TBO07 — cannot duplicate SDP control file name — (I/a/tb)
- TBO08 — cannot ALLOC data file control structures — (I/a/tb)
- TBO09 — cannot ALLOC buffer for paging Dlt_cntl structures — (I/a/tb)
- TBO10 — trying to reopen an open SDP control file — (I/a/tb)
- TBO11 — error opening a SDP control file — (I/a/tb)
- TBO12 — unable to duplicate datafile name — (I/a/tb)
- TBO13 — trying to reopen an open data file — (I/a/tb)
- TBO14 — error opening a data file — (I/a/tb)
- TBO15 — more file being opened than slots available — (I/a/tb)
- TBO16 — second open has no associated space structure — (I/a/tb)
- TBO17 — MAX_CRTRYS unsuccessful attempts at creating delta file — (I/a/tb)

TBP0 — cannot create new datafile — (lla/tb)

TBP1 — cannot unlink file — (lla/tb)

TBP2 — cannot open newly created data file — (lla/tb)

TBR0 — last seek and current read not for same file — (lla/tb)

TBR1 — read request greater than framesize — (lla/tb)

TBR2 — fatal error in get_delta — (lla/tb)

TBR3 — error seeking requested frame — (lla/tb)

TBR4 — error reading requested frame — (lla/tb)

TBS0 — seeking in an unopened file — (lla/tb)

TBU0 — currently unsupported facility in TBLIB — (lla/tb)

TBV00 — terminal type not supported — (ecd/rcv/links)

TBV01 — terminal initialization error — (ecd/rcv/links)

TBV02 — echo file error — (ecd/rcv/links)

TBV03 — could not open database — (ecd/rcv/links)

TBV04 — could not construct high level form/process table — (ecd/rcv/links)

TBV05 — offline boot may be running — (ecd/rcv/links)

TBW0 — last seek and current write not for same file — (lla/tb)

TBW1 — write request greater than framesize — (lla/tb)

TBW2 — can build new delta entry — (lla/tb)

TBW3 — fatal error in get_delta — (lla/tb)

TBW4 — error seeking in delta file — (lla/tb)

TBW5 — error writing to delta file — (lla/tb)

TE00 — cannot close delta files — (ecd/rcv/ftamaux)

TE01 — cannot commit ecd delta file — (ecd/rcv/ftamaux)

TE01 — cannot commit sg delta file — (ecd/rcv/ftamaux)

- TE02 — cannot reopen database — (ecd/rcv/ftamaux)
- TE03 — cannot delete delta files — (ecd/rcv/ftamaux)
- TE04 — cannot apply transaction — (ecd/rcv/ftamaux)
- TE05 — cannot reset to default values in audit copy — (ecd/rcv/ftamaux)
- TE06 — cannot disable prepump triggers — (ecd/rcv/ftamaux)
- TE07 — cannot release ecd trigger associated resources — (ecd/rcv/ftamaux)
- TE08 — cannot attach for concurrency check — (ecd/rcv/ftamaux)
- TE09 — Unable to release after concurrency check, ucbs may be reserved — (ecd/rcv/ftamaux)
- TE10 — cannot apply transaction because of reserved ucb - try again — (ecd/rcv/ftamaux)
- TE11 — concurrency check error — (ecd/rcv/ftamaux)
- TE12 — ucbs may be reserved — (ecd/rcv/ftamaux)
- TE13 — *UNIX* RTR ucb major state change — (ecd/rcv/ftamaux)
- TE14 — cannot enable postpump triggers — (ecd/rcv/ftamaux)
- TE15 — cannot disable postpump triggers — (ecd/rcv/ftamaux)
- TE16 — cannot close audit database — (ecd/rcv/ftamaux)
- TE17 — cannot disable prepump triggers — (ecd/rcv/ftamaux)
- TE18 — cannot disable post triggers — (ecd/rcv/ftamaux)
- TE19 — cannot disable linkpump triggers — (ecd/rcv/ftamaux)
- TE20 — cannot enable actvpump triggers — (ecd/rcv/ftamaux)
- TE21 — cannot disable actvpump triggers — (ecd/rcv/ftamaux)
- TE22 — cannot close incore database - act_db() — (ecd/rcv/ftamaux)
- TE23 — cannot delete delta files — (ecd/rcv/ftamaux)
- TE24 — onaudit() function failed — (ecd/rcv/ftamaux)
- TE25 — offaudit() function failed — (ecd/rcv/ftamaux)

TEE00 — mtlog_line returned FAIL — (ecd/rcv/textlinks)

TINFO00 — cannot insert TINFO set — (ecd/rcv/ftamecd)

TINFO01 — cannot create TINFO record — (ecd/rcv/ftamecd)

TINFO02 — cannot delete TINFO record — (ecd/rcv/ftamecd)

TINFO03 — cannot read TINFO record — (ecd/rcv/ftamecd)

TINFO04 — cannot update TINFO record — (ecd/rcv/ftamecd)

TINFO05 — cannot open set of APHUSER records — (ecd/rcv/ftamecd)

TINFO06 — cannot get record from APHUSER set — (ecd/rcv/ftamecd)

TINFO07 — cannot read SPHOPT record — (ecd/rcv/ftamecd)

TINFO08 — cannot get TINFO record — (ecd/rcv/ftamecd)

TM00 — unable to open verify file %s — (ecd/rcv/textlinks)

TM01 — could not determine terminal name — (ecd/rcv/textlinks)

TM02 — unable to open log file %s — (ecd/rcv/textlinks)

TM03 — could not process parameter — (ecd/rcv/textlinks)

TM04 — cannot open journal file — (ecd/rcv/textlinks)

TM05 — could not open database — (ecd/rcv/textlinks)

TM06 — FATAL return from ODIN — (ecd/rcv/textlinks)

TM07 — could not close database — (ecd/rcv/textlinks)

TM08 — cannot open verify file for reading — (ecd/rcv/textlinks)

TM09 — cannot open temporary file — (ecd/rcv/textlinks)

TM10 — cannot open verify file for reading — (ecd/rcv/textlinks)

TM11 — cannot open temporary file — (ecd/rcv/textlinks)

TM12 — mtlog_line returned FAIL — (ecd/rcv/textlinks)

TM13 — offline boot may be running — (ecd/rcv/textlinks)

TPF00 — could not malloc space for formname — (ecd/rcv/textlinks)

TPF01 — FATAL RETURN — (ecd/rcv/textlinks)

TPF02 — mtlog_line returned FAIL — (ecd/rcv/textlinks)

TPHOPT00 — cannot create TPHOPT record — (ecd/rcv/ftamecd)

TPHOPT01 — cannot delete tphopt from ecd — (ecd/rcv/ftamecd)

TPHOPT02 — cannot read tphopt in ecd — (ecd/rcv/ftamecd)

TPHOPT03 — cannot update tphopt in ecd — (ecd/rcv/ftamecd)

TPP00 — could not get space for argument to keyword %s —
(ecd/rcv/textlinks)

TPP01 — mask pathname too long — (ecd/rcv/textlinks)

TPP02 — FATAL RETURN — (ecd/rcv/textlinks)

TPP03 — mtlog_line returned FAIL — (ecd/rcv/textlinks)

TR01 — cannot malloc memory for new TR_BLK — (lla/tb)

TR02 — unknown TR_ACTION value — (lla/tb)

TR03 — fatal error during prcss_chng — (lla/tb)

TR04 — no slots remain in conntab — (lla/tb)

TR05 — malloc failure — (lla/tb)

TR06 — specific DATADIR * does not match — (lla/tb)

TR07 — freeslots count bad — (lla/tb)

TR08 — commit error — (lla/tb)

TR09 — no transaction block for specified datadir * — (lla/tb)

TR10 — cannot get record set type info — (lla/tb)

TR11 — cannot create file for trigger information — (lla/tb)

TR12 — cannot write database connection information — (lla/tb)

TR13 — cannot write trigger information — (lla/tb)

TR14 — cannot write subschema name used in trigger — (lla/tb)

TR15 — cannot close file with trigger information — (lla/tb)

TR16 — EOF while reading string — (lla/tb)

TR17 — error while reading string — (lla/tb)

TR18 — cannot malloc space for string read — (lla/tb)

TR19 — string being read greater than 160 characters long — (lla/tb)

TR20 — cannot malloc trigger condition block — (lla/tb)

TR21 — triggers not setup for old trigger DATADIR — (lla/tb)

TR22 — cannot open trigger information file — (lla/tb)

TR23 — cannot read old connection DATADIR pointer — (lla/tb)

TR24 — cannot read old database file name — (lla/tb)

TR25 — cannot read old trigger control information — (lla/tb)

TR26 — cannot read old subschema name — (lla/tb)

TR27 — unable to save trigger control entry — (lla/tb)

TRAPHU00 — cannot access aphuser record — (ecd/rcv/integ)

TRAPHU01 — cannot update aphuser rids — (ecd/rcv/integ)

TRAUD00 — cannot load audrec record — (ecd/rcv/integ)

TRAUD01 — cannot update audrec rids — (ecd/rcv/integ)

TRCI00 — cannot access ciopt record — (ecd/rcv/integ)

TRCI01 — cannot update ciopt rids — (ecd/rcv/integ)

TRECD00 — cannot access ecdorg record — (ecd/rcv/integ)

TRECD01 — cannot update ecdorg rids — (ecd/rcv/integ)

TRFTI00 — cannot access ftaminopt record — (ecd/rcv/integ)

TRFTI01 — cannot update ftaminopt rids — (ecd/rcv/integ)

TRFTO00 — cannot access ftamopt record — (ecd/rcv/integ)

TRFTO01 — cannot update ftamopt rids — (ecd/rcv/integ)

TRFTU00 — cannot access ftamuser record — (ecd/rcv/integ)
TRFTU01 — cannot update ftamuser rids — (ecd/rcv/integ)
TRGFSA00 — cannot access gfsappl record — (ecd/rcv/integ)
TRGFSA01 — cannot update gfsappl rids — (ecd/rcv/integ)
TRGFSC00 — cannot access gfscom record — (ecd/rcv/integ)
TRGFSC01 — cannot update gfscom rids — (ecd/rcv/integ)
TRGFSR00 — cannot access gfsresp record — (ecd/rcv/integ)
TRGFSR01 — cannot update gfsresp rids — (ecd/rcv/integ)
TRGFSS00 — cannot access gfssec record — (ecd/rcv/integ)
TRGFSS01 — cannot update gfssec rids — (ecd/rcv/integ)
TRLD00 — cannot access logdev record — (ecd/rcv/integ)
TRLD01 — cannot update logdev rids — (ecd/rcv/integ)
TRMDCT00 — cannot access mdct record1 — (ecd/rcv/integ)
TRMDCT01 — cannot update mdct rids — (ecd/rcv/integ)
TRPPHU00 — cannot access pphuser record — (ecd/rcv/integ)
TRPPHU01 — cannot update pphuser rids — (ecd/rcv/integ)
TRRUNA00 — cannot access runa record — (ecd/rcv/integ)
TRRUNA01 — cannot update runa rids — (ecd/rcv/integ)
TRSCSD00 — cannot access scsdopt record — (ecd/rcv/integ)
TRSCSD01 — cannot update scsdopt rids — (ecd/rcv/integ)
TRUCB00 — cannot access ucb record — (ecd/rcv/integ)
TRUCB01 — cannot update ucb rids — (ecd/rcv/integ)
TSIG00 — RC/V killed by termination signal %d — (ecd/rcv/textlinks)
TSM00 — cannot open log file for reading — (ecd/rcv/textlinks)
TSM01 — cannot open temporary file — (ecd/rcv/textlinks)

- TSM02 — cannot open log file for reading — (ecd/rcv/textlinks)
- TSM03 — cannot open temporary file — (ecd/rcv/textlinks)
- TTOPT00 — cannot create TTOPT record — (ecd/rcv/ftamecd)
- TTOPT01 — cannot delete ttopt from ecd — (ecd/rcv/ftamecd)
- TTOPT02 — cannot read ttopt in ecd — (ecd/rcv/ftamecd)
- TTOPT03 — cannot update ttopt in ecd — (ecd/rcv/ftamecd)
- UA00 — unable to load set member block — (lla/dml)
- UA01 — bad set member block sanity id — (lla/dml)
- UA02 — unable to load set header — (lla/dml)
- UA03 — bad set header sanity id — (lla/dml)
- UA04 — erase controller error — (lla/dml)
- UA05 — insert controller error — (lla/dml)
- UA06 — unable to write set header — (lla/dml)
- UA07 — unable to unload set member block — (lla/dml)
- UA08 — cannot unload set member block during error recovery — (lla/dml)
- UA09 — cannot unload set header during error recovery — (lla/dml)
- UB00 — cannot close incore ecd — (ecd/rcv/ftamaux)
- UB01 — cannot attach boot ecd — (ecd/rcv/ftamaux)
- UB02 — cannot clear semaphore in boot ecd — (ecd/rcv/ftamaux)
- UB03 — cannot close boot ecd — (ecd/rcv/ftamaux)
- UB04 — cannot attach incore ecd — (ecd/rcv/ftamaux)
- UCB 0 — cannot change ucb with ACT, STBY, or INIT major states — (ecd/rcv/ftamecd)
- UCB 10 — cannot synchronize disk and incore database images — (ecd/rcv/ftamecd)

- UCB 12 — cannot change the major state to GROW when CU status is ACT — (ecd/rcv/ftamecd)
- UCB 13 — change not allowed on RNG type ucb when ACT — (ecd/rcv/ftamecd)
- UCB 14 — cannot change rexinh value when t_rexinh on ucbdyn is set to y — (ecd/rcv/ftamecd)
- UCB 15 — 256 are valid for 3B21D MASC CU — (ecd/rcv/ftamecd)
- UCB 16 — equipage values must be divisible by 8 for 3B21D MASC CU — (ecd/rcv/ftamecd)
- UCB 17 — could not read up pointer — (ecd/rcv/ftamecd)
- UCB 18 — %s cannot be NULL for 3B21D when device_type is SMT, DAT, or MHD — (ecd/rcv/ftamecd)
- UCB 19 — only a|A, b|B, or c|C are valid for 3B21D aisle — (ecd/rcv/ftamecd)
- UCB 1 — major state change from %s to %s is not valid — (ecd/rcv/ftamecd)
- UCB 20 — could not read down pointer — (ecd/rcv/ftamecd)
- UCB 21 — could not read side pointer — (ecd/rcv/ftamecd)
- UCB 22 — when CUs for channel %d are in GROW or UNEQIP — (ecd/rcv/ftamecd)
- UCB 23 — Both controllers must not be UNEQIP when unit is GROW — (ecd/rcv/ftamecd)
- UCB 24 — Lower unit must be UNEQIP or GROW when unit is GROW — (ecd/rcv/ftamecd)
- UCB 25 — Lower unit must be UNEQIP when unit is UNEQIP — (ecd/rcv/ftamecd)
- UCB 26 — Above controllers must not be UNEQIP when unit is GROW — (ecd/rcv/ftamecd)
- UCB 27 — Side unit must be UNEQIP or GROW when unit is GROW — (ecd/rcv/ftamecd)
- UCB 28 — Side unit must be UNEQIP when unit is UNEQIP — (ecd/rcv/ftamecd)
- UCB 29 — an SBUS is needed — (ecd/rcv/ftamecd)

- UCB 2 — cannot get ucblst node — (ecd/rcv/ftamecd)
- UCB 3 — must do a trend before another major state transition on this ucb — (ecd/rcv/ftamecd)
- UCB 4 — only major state and ucb linkage changes allowed in current state — (ecd/rcv/ftamecd)
- UCB 5 — can only insert/delete a ucb in UNEQIP on INCORE ecd — (ecd/rcv/ftamecd)
- UCB 6 — can only insert/delete a ucb in GROW, OOS, UNEQIP on any DISK ecd — (ecd/rcv/ftamecd)
- UCB 7 — internal error: bad entry in ucbchanges table — (ecd/rcv/ftamecd)
- UCB 8 — must do a trend before deleting this ucb — (ecd/rcv/ftamecd)
- UCB00 — cannot create UCB record — (ecd/rcv/ftamecd)
- UCB01 — cannot delete ucb from ecd — (ecd/rcv/ftamecd)
- UCB02 — cannot read ucb in ecd — (ecd/rcv/ftamecd)
- UCB03 — cannot update ucb in ecd — (ecd/rcv/ftamecd)
- UCB04 — cannot break old links — (ecd/rcv/ftamecd)
- UCB05 — cannot read ecdorg record — (ecd/rcv/ftamecd)
- UCBDYN00 — bad rid stored in symbol table — (ecd/rcv/ftamecd)
- ULARP00 — cannot insert ULARP set — (ecd/rcv/ftamecd)
- ULARP01 — cannot create ULARP record — (ecd/rcv/ftamecd)
- ULARP02 — cannot delete ULARP record — (ecd/rcv/ftamecd)
- ULARP03 — cannot read ULARP record — (ecd/rcv/ftamecd)
- ULARP04 — cannot update ULARP record — (ecd/rcv/ftamecd)
- ULARP05 — Too many arguments in proc_args field, (max. of nine) — (ecd/rcv/ftamecd)
- ULEV0 — Cannot get ucb record — (ecd/rcv/integ)
- ULEV1 — Cannot get next higher ucb record — (ecd/rcv/integ)

ULEV2 — Cannot update ucb record — (ecd/rcv/integ)

UN02 — id out of range %lot%lo — (lla/sdp)

UN03 — attempt to unuse an item not core resident %ot%lo — (lla/sdp)

UN04 — whichframe returned an impossible value %ot%dt%d — (lla/sdp)

UN05 — no permissions for WRTN %o — (lla/sdp)

UN06 — unknown permissions %ot%d — (lla/sdp)

UN07 — attempt to unuse an unused item %o %O — (lla/sdp)

UN08 — compilation made without either LFU or LRU defined — (lla/sdp)

UNAU — audit failed in sdp_unuse — (lla/sdp)

UP00 — unable to load record — (lla/dml)

UP01 — unable to unload ridblk entry — (lla/dml)

UP02 — unable to load actual record — (lla/dml)

UP03 — bad record sanity id — (lla/dml)

UP04 — bad subsid for record specified — (lla/dml)

UP05 — unable to load set member block — (lla/dml)

UP06 — bad set member block sanity id — (lla/dml)

UP07 — dml built with smaller record size (FULLBUFFLEN) than user record — (lla/dml)

UP08 — displacement to pointer error — (lla/dml)

UP09 — dml built with smaller record size (FULLBUFFLEN) than user record — (lla/dml)

UP10 — pointer to displacement error — (lla/dml)

UP11 — unable to relocate enlarged record — (lla/dml)

UP12 — displacement to pointer error with new record — (lla/dml)

UP13 — unable to load set header — (lla/dml)

UP14 — bad set header sanity id — (lla/dml)

- UP15 — key breakout error with old record — (lla/dml)
- UP16 — key breakout error with new record — (lla/dml)
- UP17 — erase controller error — (lla/dml)
- UP18 — insert controller error — (lla/dml)
- UP19 — unable to write set header — (lla/dml)
- UP20 — unable to unload set member block — (lla/dml)
- UP21 — unable to write record — (lla/dml)
- UP22 — cannot unload set member block during error recovery — (lla/dml)
- UP23 — cannot unload record during error recovery — (lla/dml)
- UP24 — cannot unload set header during error recovery — (lla/dml)
- UP25 — cannot copy old largerec — (lla/dml)
- UP26 — cannot copy large, fix-length record — (lla/dml)
- UP27 — cannot write large, fix-length record — (lla/dml)
- UP28 — record header bad, rectype not FIXSIZE or VARIABLESIZE
— (lla/dml)
- UP29 — cannot write revised large, variable-length record — (lla/dml)
- UP30 — cannot redeem space at end of variable-length record — (lla/dml)
- UPDLK 0 — reference %.20s in %s record still
unresolved — (ecd/rcv/prelinks)
- UPDLK02 — invalid link value 0x%x from %s to %s — (ecd/rcv/prelinks)
- UPDLK03 — Invalid %s link(%.20s) on %s — (ecd/rcv/prelinks)
- UPDLK04 — invalid link value 0x%x from %s to %s — (ecd/rcv/prelinks)
- UPLK00 — cannot update link — (ecd/rcv/prelinks)
- UPLK01 — cannot increment reference count — (ecd/rcv/prelinks)
- US01 — space not connected %o — (lla/sdp)
- US02 — no permissions for WRTN %o — (lla/sdp)

US03 — unknown permissions %ot%d — (lla/sdp)

US04 — id out of range %ot%lot%lo — (lla/sdp)

US05 — id not aligned to %d-byte boundary %ot%lo — (lla/sdp)

US06 — framesearch returns impossible value %ot%dt%d — (lla/sdp)

USAU — audit failed in sdp_use — (lla/sdp)

VA01 — attempt to vacate a connected environment %o — (lla/sdp)

VA02 — attempt to vacate unhoused environment %o — (lla/sdp)

VAAU — audit failed in sdp_vacate — (lla/sdp)

VIOP0 — Cannot get ucb record — (ecd/rcv/integ)

VIOP1 — Device type is not defined in iop table — (ecd/rcv/integ)

VIOP2 — Could Not read up pointer — (ecd/rcv/integ)

VIOP3 — Could Not read down pointer — (ecd/rcv/integ)

VIOP4 — ucb record %s %d and %s %d have identical pcid field —
(ecd/rcv/integ)

VIOP5 — ucb records %s %d and %s %d have identical pcsdid fields —
(ecd/rcv/integ)

VIOP6 — Could Not read side pointer — (ecd/rcv/integ)

VIOP7 — ucb record %s %d and %s %d have same pcid field —
(ecd/rcv/integ)

VIOP8 — ucb record %s %d and %s %d has same pcsdid field —
(ecd/rcv/integ)

WF00 — unknown form type encountered name = %s — (ecd/rcv/prelinks)

WF01 — found id beyond file limit %lot%dt%d — (lla/sdp)

WF02 — fail to build file name for creation %s%d — (lla/sdp)

WF03 — fail to create file %s — (lla/sdp)

WF04 — fail to build file name for opening %s%d — (lla/sdp)

WF05 — fail to open file %s — (lla/sdp)

- WF06 — fail to position file %d — (lla/sdp)
- WF07 — fail to write %d — (lla/sdp)
- WHRS01 — cannot create WORKHRS record — (ecd/rcv/ftamecd)
- WHRS02 — cannot read WORKHRS record — (ecd/rcv/ftamecd)
- WHRS03 — cannot update WORKHRS record — (ecd/rcv/ftamecd)
- WK00 — cannot walk directory hierarchy — (ecd/rcv/ftamsg)
- WK01 — cannot walk last directory in hierarchy — (ecd/rcv/ftamsg)
- WL00 — bad PLACE sanity id — (lla/audit)
- WN00 — bad PLACE sanity identifier — (lla/audit)
- WN01 — bad w_status in PLACE structure — (lla/audit)
- WS01 — fail to open header file %s — (lla/sdp)
- WS02 — fail to position header file %s — (lla/sdp)
- WS03 — fail to write header file %s — (lla/sdp)
- WS04 — fail to position header file %s — (lla/sdp)
- WS05 — fail to write header file %s — (lla/sdp)
- WS06 — fail to position header file %s — (lla/sdp)
- WS07 — fail to write header file %s — (lla/sdp)
- WS08 — fail to close header file %s — (lla/sdp)
- WS09 — fail to sync database %s — (lla/sdp)
- YN00 — sdp_sync error — (lla/dml)
- a00_aud — audrec %s %d must have at least one associated audinst —
(ecd/rcv/integ)
- a10_aphu — aphuser record %s %s does not have an associated tsapinfo
record — (ecd/rcv/integ)
- a10_bparm — for non 3B21D model, %d is an invalid pages value —
(ecd/rcv/integ)

- a10_ftamopt — ftamopt record %s does not have an associated gfscom record — (ecd/rcv/integ)
- a10_ftamu — ftamuser record %s does not have an associated tsapinfo record — (ecd/rcv/integ)
- a10_gfsa — gfsappl %s has logdev at position %2d duplicated at gfsappl %s with logdev at position %2d — (ecd/rcv/integ)
- a10_gfsc — gfsappl form '%s' is linked to a gfscom form where the file access action '%s' conflicts with a gfssec form — (ecd/rcv/integ)
- a10_gfsc — gfsappl form '%s' is linked to a gfscom form where the gfsresp form passwords are invalid for the FTAM88 protocol — (ecd/rcv/integ)
- a10_gfsr — gfsappl form '%s' is linked to a gfscom form where the file access action '%s' conflicts with a gfssec form — (ecd/rcv/integ)
- a10_gfsr — gfsappl form '%s' is linked to a gfscom form where the gfsresp form passwords are invalid for the FTAM88 protocol — (ecd/rcv/integ)
- a10_gfss — gfsappl form '%s' is linked to a gfscom form where the file access action '%s' conflicts with a gfssec form — (ecd/rcv/integ)
- a10_mdct — not all ucb's pointed to by mdct %s are of the same type — (ecd/rcv/integ)
- a10_pack — partitions for pack %2.2s %d define too many blocks — (ecd/rcv/integ)
- a10_part — partitions for pack %2.2s %d define too many blocks — (ecd/rcv/integ)
- a10_ucb — ucb %s %u %s %u points up to a device of wrong type — (ecd/rcv/integ)
- a10_vchnum — ucb %s %u %s %u channel_num does not match the controller ucb %s %u %s %u channel_num — (ecd/rcv/integ)
- a10_vdid — ucb %s %u %s %u has the same internal_device ID %u as existing ucb %s %u %s %u ID %u — (ecd/rcv/integ)
- a11_mdct — mdct %s, of type %2.2s, is linked to ucb's of a wrong type — (ecd/rcv/integ)
- a11_pack — unsupported packsize specified on pack form %2.2s %d — (ecd/rcv/integ)

- a11_part — fs<->fp backup found: pack %2.2s %d partitions %d %d
— (ecd/rcv/integ)
- a11_ucb — ucb %s %u %s %u points up to ucb %s %u %s %u,
which has no down pointer — (ecd/rcv/integ)
- a11_vdid — ucb %s %u %s %u has the same internal_device ID %u as
existing ucb %s %u %s %u ID %u — (ecd/rcv/integ)
- a12_gfsa — gfsappl %s with vfs_id %d has no pphuser form —
(ecd/rcv/integ)
- a12_mdct — mdct %s has unknown type %2.2s: no check against ucb type
performed — (ecd/rcv/integ)
- a12_pack — bootdisk requires lboot or lboot21,bboot(pri),& root - missing
%s — (ecd/rcv/integ)
- a12_part — fs/fp backup not mutual: pack %2.2s %d partitions %d %d
— (ecd/rcv/integ)
- a12_ucb — ucb %s %u %s %u not found in side chain
search — (ecd/rcv/integ)
- a13_gfsa — gfsappl form '%s' is linked to a gfscom form where the file
access action '%s' conflicts with a gfssec form — (ecd/rcv/integ)
- a13_pack — bootdisk partition size %d exceeds maximum of %d —
(ecd/rcv/integ)
- a13_part — fs/fp backup has none or both primary: pack %2.2s %d partitions
%d %d — (ecd/rcv/integ)
- a13_ucb — ucb %s %u %s %u was not found in side chain search, possible
loop may exist — (ecd/rcv/integ)
- a14_gfsa — gfsappl form '%s' duplicates a virtual filestore id with gfsappl form
- a14_part — missing backup partition: pack %2.2s %d partition %d
— (ecd/rcv/integ)
- a14_ucb — ucb %s %u %s %u points down to a device of wrong type —
(ecd/rcv/integ)
- a15_gfsa — gfsappl form '%s' is linked to a gfscom form where the gfsresp
form passwords are invalid for the FTAM88 protocol — (ecd/rcv/integ)
- a15_pack — unsupported packsize specified on pack form %2.2s %d —
(ecd/rcv/integ)

a15_ucb — ucb %s %u %s %u points down to ucb %s %u %s %u, which does not point back up properly — (ecd/rcv/integ)

a16_gfsa — gfsappl %s has ATD active and no tape initiator — (ecd/rcv/integ)

a16_part — bootable disk - partition size %d exceeds %d blocks — (ecd/rcv/integ)

a16_ucb — ucb %s %u %s %u has incorrect up0 pointer for its side chain — (ecd/rcv/integ)

a17_gfsa — gfsappl %s has ADLD active and no data link process — (ecd/rcv/integ)

a17_part — boot disk requires lboot or lboot21 ,bboot(pri),& root - missing %s — (ecd/rcv/integ)

a17_ucb — ucb %s %u %s %u has incorrect up1 pointer for its side chain — (ecd/rcv/integ)

a18_part — for boot disk ppn of 1 is reserved for mkdsk use — (ecd/rcv/integ)

a18_ucb — ucb %s %u %s %u points down to beginning of a side chain containing possible loop — (ecd/rcv/integ)

a19_ucb — ucb %s %u %s %u points to the beginning of a side chain containing possible loop — (ecd/rcv/integ)

a20_ftamopt — ftamopt %s pointed to by gfscom %s must have non-zero ckpt_window with restart fu set — (ecd/rcv/integ)

a20_ftamopt — ftamopt %s pointed to by gfscom %s cannot have ftam_qos of NORECOV with restart or recovery fu set — (ecd/rcv/integ)

Acronyms

A

ACP

Alarm Control Process

ACSE

Associated Control Service Element

ACU

Automatic Calling Unit

ADP

Automatic Diagnostic Process

AFT

Application File Transfer

AIM

Application Integrity Monitor

AMA

Automatic Message Accounting

APH

Application Protocol Handler

API

Attached Processor Interface

APS

Attached Processor System

ASCII

American Standard Code for Information Interchange

ATP

All Tests Pass

AUDERR

Audit Error Recovery

AUDHIST

Audit Instance History

AUDINST

Audit Instance Information

AUDREC

Audit Record

AUDSTAT
Audit Status

B

BOT
Beginning of Tape

BXSLH
BX.25 Session Layer Handler

BXSLINFO
X.25 Session Layer Information

BXSLLOC
X.25 Session Layer Local Users Information

BXSLRMT
X.25 Session Layer Remote Users Information

C

CC
Central Control

CCIS
Common Channel Interoffice Signaling

CCITT
International Telegraph and Telephone Consultative Committee

CDOPT
Control Display Option Block

CHs
Channels

CHOPT
Option Block for IOP Controller

CIOPT
Craft Interface Option Block

CLASSDEF
Class Definition Record

CMON
Craft Interface Integrity Monitor

CN
Change Notice

CONFIG
Configuration Control

CPBLX3
BX.25 Datalink Option

CPPHNONE
BX.25 Datalink Option

CPU
Central Processing Unit

CR
Carriage Return

CRC
Cyclic Redundancy Check

CRT
Cathode Ray Tube

CSOP
Coordinator of Spooler Output Processes

CSU
Cache Store Unit

CU
Control Unit

CTIP
Customer Training and Information Product

D

DAP
Display Administration Process

DAT
Digital Audio Tape

DBDUMP
Database Dump

DBINFO
Database Information

DBLOAD
Database Load

DBMS

Database Management System

DCE

Data Circuit-terminating Equipment

DCI

Dual-Serial Channel (DSCH) Computer Interconnect

DCN

Device Control Number

DSCH

Dual-Serial Channel

DEVICE

Device Definition Record

DFC

Disk File Controller

DIOP

Disk Independent Operation

DIAMON

Diagnostic Monitor

DMA

Direct Memory Access

DMAC

Direct Memory Access Controller

DMERT

Duplex Multienvironment Real-Time

DSCH

Dual Serial Channel

DT

Data Terminal

DTE

Data Terminal Equipment

DTP

Data Transfer Process

DU Data Unit

DUI

Direct User Interface

DUIC

Direct User Interface Controller

DUOPT

Direct User Interface Option

DUOPTDYN

Direct User Interface Option Dynamic Data

E

EAI

Emergency Action Interface

EAILOPT

Emergency Action Interface Option Block

EBCDIC

Extended Binary Coded Decimal Interchange Code

ECD

Equipment Configuration Database

ECDMAN

ECD Manager Process

ECDORG

ECD Origination Record

EIA

Electronic Industries Association

EIR

Enhanced Information Report

EOT

End of Tape

EQL

Equipment Location

ERRCOUNT

Error Counter

ERRLOG

Error Logger

ERRPORT

Error Report

ERRTHLD

Error Threshold

F

FADU

File Access Data Unit

FIFO

First-In First-Out

FLDUPD

Field Update

FP

File Partition

FPB

File Partition Body

FPDU

FTAM Protocol Data Unit

FS

File System

FSB

File System Body

FSL

FTAM Service Library

FTAM

File Transfer Access and Management

G

GACK

Generic Update Acknowledgment

GETTY

Get Shell Record

GFS

General File Server

GFT

General File Transfer

GMT

Greenwich Mean Time

GOS
Generic Update Output Spooler

H

HA-OMP
High Availability-Operations and Maintenance Platform

HDLC
High-level Data Link Control

HLMODE
High-Level Mode

I

id
identification

I/O
Input/Output

IOP
Input/Output Processor

IP
Information Product

ISO
International Standards Organization

K

KLIST
Key List

L

LANA

Local Area Network Address

LDIN

Logical Device Identifier

LDP

Processor Loader

LLA

Low-Level Access

LOGDEV

Logical Device Information

M

MAS

Main Store

MASC

Main Store Controller

MCRT

Maintenance CRT

MDCT

Minor Device Chain Table

MHD

Moving Head Disk

MIRA

Maintenance Input Request Administrator

MML

Man-Machine Language

MT

Magnetic Tape

MTC

Magnetic Tape Controller

MTTY

Maintenance TTY

MTTYC

Maintenance TTY Controller

MTTYPC

Maintenance Teletypewriter Peripheral Controller

N

NPH

Network Protocol Handler

NPHOPT

Network Protocol Handler Option

NSAP

Network Service Access Point

NTR

No Tests Run

O

ODIN

On-line Data Integrity

OFL

Off Line

OMDB

Output Message Database

OOS

Out of Service

OSI

Open Systems Interconnection

OSPS

Operator Services Position System

P

PC

Peripheral Controller

PCPEIH

Processor Controller Processor Error Interrupt Handler

PCSD

Peripheral Controller Subdevice

PDS

Program Documentation Standard

PDU

Protocol Data Unit

PF

Program/Programmable Function

PIC

Peripheral Interface Controller

PL

Presentation Layer *or* Programmable Link

PLC

Programmable Link Controller

PPH

ACSE/Presentation Layer Protocol Handler

PRM

Processor Recovery Message

PRTLST

Print List

PSAP

Presentation Service Access Point

PSI

Peripheral System Interface

PSM

Power Switch Monitor

PU

Peripheral Unit

PVC

Permanent Virtual Circuit

R

RC

Recent Change

RC/V

Recent Change and Verify

RCVPARAMS

RC/V Parameter Information

RID

Record Identifier

ROP

Read-Only Printer

RTS

Real-Time Status

RTSAP

Remote Transport Session Access Point

RUNA

Remote User Network Information

S

SAT

Store Address Translator

SBUS

Small Computer System Interface (SCSI) Bus

SC

Scanner, Session Connection, *or* Store Complete

SCC

Switching Control Center

SCDL

Switching Control Data Link

SCH

Serial Channel

SCM

Session Count Message

SCSD

Scanner and Signal Distributor

SCSDC

Scanner and Signal Distributor Controller

SCSDMAP

Scanner and Signal Distributor Map

SCSDOPT

Scanner and Signal Distributor Option

SCSI

Small Computer System Interface

SD

Signal Distributor

SDL

Synchronous Data Link

SDLC

Synchronous Data Link Controller

SDP

Software Demand Paging

SDSDA

SCSD Administrator

SG

System Generation

SGEN1

System Generation 1 (parameters)

SGEN2

System Generation 2 (boot processes)

SI

System Identification

SIM

System Integrity Monitor

SIMCNTL

System Integrity Monitor Control Record

SMD

Storage Module Drive

SOP

Spooler Output Process

SPDU

Session Protocol Data Unit

SPH

Session Protocol Handler

SPU

Small Computer System Interface (SCSI) Peripheral Unit

SPLRINFO

Spooler Information Record

SRESM

Session Resynchronization Message

SSAP

Session Service Access Point

STF

Some Tests Fail

SVC

Switched Virtual Circuit

T

TC

Transport Connection

TIM

Timer

TLP

Transport Layer Protocol or Trouble Location Process

TOS

TTY Output Spooler

TPDU

Transport Protocol Data Unit

TPH

Transport Protocol Handler

TRBEGIN

Transaction Begin

TREND

Transaction End

TSAP

Transport Service Access Point

TSDU

Transport Service Data Unit

TTOPT

TTY Terminal Option

TTY

Teletypewriter

TTYC

TTY Controller

U

UC

Utility Circuit

UCB

Unit Control Block

UCBDYN

Unit Control Block Dynamic

UDP

User Diagnostic Process

ULARP

User-Level Automatic Restart Process

UNAV

Unavailable

UNEQIP

Unequipped

V

VFS
Virtual File Store

VFY
Verify

VFYDFLT
Verify Default

VTOC
Volume Table of Contents

X

XEQ
Execute

Glossary

B

Batch Script

A file of input commands that is executed instead of input from the keyboard.

D

Degrowth

Procedure that allows deletion of the common hardware units of the 3B20D/3B21D computers.

Directory

A node that points to other directories and files.

Driver

A process that controls a hardware device.

E

Equipment Configuration Database

Database that describes the physical and logical configuration of the 3B20D/3B21D computers and peripheral hardware. The database contains configuration and control information for the audit and craft interface software systems.

F

File Partition

Data stored in a specific location on disk.

Flat File

File in which all Software Demand Paging (SDP) files are combined into a contiguous file.

Form

Template or mask used for input information.

Form Navigation

Control characters that allow you to move the cursor from field to field within a particular form or to change forms.

G

Growth

Procedure that allows addition of the common hardware units of the 3B20D/3B21D computers.

I

***Incore* Database**

Database read off of disk that is resident (*incore*) in the main memory of the 3B20D/3B21D computers. This is the database used by the operating system.

inode

Data structure that contains information about a file.

K

Key

A field that uniquely identifies a form instance.

L

Low-Level Access Database

Contains data required by the low-level access Database Management System (DBMS).

R

Recent Change and Verify System

Processes that allow the entry, modification, and deletion of data in the Equipment Configuration Database (ECD) and System Generation (SG) databases.

S

System Generation Database

Database that contains information needed to build a disk image and boot image of 3B20D/3B21D computers.

Index

/etc file system, [1-12](#)
2-level naming convention for UCBs, [6-299](#)
3B20D and 3B21D computers, [1-1](#)
3BGEN, [6-1](#)
3**bmkdsk** program, [6-344](#)

A

acknowledgment, [4-20](#)
activate, [6-5](#)
activate form, [5-15](#)
aftopt, [6-6](#)
American Standard Code for Information Interchange, [5-3](#)
aphopt, [6-8](#)
aphuser, [6-10](#)
Application File Transfer, [6-6](#)
Application File Transfer Option Record, [6-6](#)
auderr, [1-9](#) [6-15](#)
audhist, [6-20](#)
audinst, [6-22](#)
audit, [1-1](#) [1-9](#)
audit instance-independent information, [6-24](#)
audits, error categories for, [6-15](#)
audrec, [1-9](#) [6-24](#)
audstat, [6-31](#)
authdef, [6-34](#)
automatic restart process, [1-2](#) [1-9](#)

B

backup devices, [1-4](#)
batch input file, [5-5](#)
batch script, applying, [5-5](#)
black and white RC/V terminal, [5-8](#)
boot image, [1-1](#) [1-10](#) [1-14](#) [6-355](#)
boot time, [6-355](#)
boot time parameters, [1-10](#) [6-38](#) [6-44](#) [6-50](#) [6-57](#)
booting, [1-10](#) [1-13](#)
bslinfo, [6-63](#)
btihead, [6-36](#)
btparm, [1-10](#)
btparm [G2-R6.3, R6.5-R6.8], [6-38](#)

btparm [R21.1 & R21.3], [6-50](#)
btparm [R21.5 & Later], [6-57](#)
btparm [R6.4 Only], [6-44](#)
bxslloc, [6-64](#)
bxslrmt, [6-69](#)

C

cache store unit device, [3-6](#)
cdopt, [6-71](#)
central control device, [3-5](#)
ctfinfo, [6-74](#)
change mode prompt, [4-18](#)
chopt, [3-13](#)
chopt, [6-75](#)
ciopt, [6-78](#)
classdef, [6-81](#)
classdef, class definition record form, [5-10](#)
color RC/V, [5-6](#)
color RC/V terminal, [5-8](#)
color terminal, [5-6](#)
command line options, [4-6](#)
command line options, *rcvparams* form, [4-4](#)
Command, Filename, and Display Notations, [xiii](#)
commands, information, [4-11](#)
commands, list navigational, [4-11](#)
commands, list value, [4-10](#)
continuation character, [4-19](#) [4-21](#) [4-24](#)
control display page, [4-7](#)
control unit complex, [3-4](#)
 cache store unit device, [3-6](#)
 central control device, [3-5](#)
 control unit device, [3-4](#)
 direct memory access device, [3-7](#)
 dual serial channel device, [3-8](#)
 main store controller device, [3-5](#)
 serial channel device, [3-8](#)
 store address translator device, [3-6](#)
 utility circuit device, [3-7](#)
control unit device, [3-4](#)
conventions, [4-1](#)
Conventions Used,
 Command, Filename, and Display Notations, [xiii](#)
 Equipment Locations, [xv](#)
 Hexadecimal Notation, [xv](#)
 Safety Labels, [xv](#)
 Signal Designations, [xv](#)
correlating high-level forms to high-level processes, [7-14](#)

cpblx[6-683](#)
cpphone[6-692](#)
craft interface, [1-1](#) [1-9](#)
craft output, [1-9](#)
craft shell terminal, [4-3](#)
craft shells, [1-9](#)
CREATEECD command, [4-2](#)
CREATEECD, create skeleton ECD, [4-1](#)
CREATESG command, [4-2](#)
CREATESG, create skeleton SG database, [4-1](#)
CSU device, [3-6](#)
CSU *ucb* default values, standard, [3-6](#)
CU, [3-4](#)
CU complex, [3-5](#)
current error counts, *ucb*, [6-141](#)

D

database files, [1-12](#)
database free space, [5-6](#)
database space, [5-3](#)
database space, reclaiming fragmented,
 dbload, [5-3](#)
database tools,
 conventions, [4-1](#)
 CREATEECD command, [4-2](#)
 CREATESG command, [4-2](#)
 MML terminals, [4-1](#)
 PDS terminals, [4-1](#)
DB option, [4-2](#)
dbdump[6-6359](#)
dbinfo[6-699](#)
dbinfo form,
 finding all links to a specific form, [5-5](#)
 finding all UCBs under a specific UCB, [5-4](#)
 finding the RID of a specific form, [5-5](#)
 listing UCBs under a specific IOP, [5-4](#)
dbload[6-6360](#)
dciadd, adding a DCI, [7-9](#)
dcidel, deleting a DCI, [7-12](#)
dciopt[6-6107](#)
DCN, [2-2](#)
default file, specified, RC/V high-level [7-77](#)
default files, low-level ECD form [7-77](#)
default values, verifying low-level ECD form [7-77](#)
definitions, mnemonics, [6-4](#)
degrowth, [7-1](#)
delete operation, [4-16](#)
device[6-6110](#)

device control number, [2-2](#)
device layout information, [1-5](#)
dfcopt[6-6117](#)
diagnostic message, [4-27](#)
direct memory access device, [3-7](#)
direct user interface controller device [3-3](#) [16](#)
direct user interface device, [3-21](#)
disk,
 file partitions, [1-11](#)
 file systems, [1-11](#)
 logical size, [1-11](#)
 name, [1-11](#)
 partitions, [1-11](#)
disk ECDs, [1-13](#)
disk file controller complex, [3-9](#)
 disk file controller device, [3-9](#)
 moving head disk device, [3-10](#) [3-11](#)
 SCSI bus, [3-9](#)
disk file controller device, [3-9](#)
disk layout forms,
 fp, [1-111](#)
 fpb, [1-111](#)
 fs, [1-111](#)
 fsb, [1-111](#)
 pacf, [1-111](#)
disk layout information, [1-10](#)
display position, [4-6](#)
Display Special Device File, [3-18](#)
display terminal, [4-3](#) [44-19](#)
DMA device, [3-7](#)
DSCH device, [3-8](#)
dst, [6-6121](#)
dual serial channel device, [3-8](#)
DUI device, [3-21](#)
DUIC device, [3-16](#)
duopt[6-6126](#)
duoptdyr[6-6128](#)

E

eaiopt[6-6130](#)
ECD, [6-1](#)
ECD and SG database, building,
 DB option, [4-2](#)
 FORMTYPE option, [4-2](#)
ECD and SG RC/V forms, listing, [6-1](#)
ECD forms,
 activat[6-6-5](#)
 aftop[6-66](#)

ECD forms (Continued)

aphopt, [6-8](#)
aphuser, [6-10](#)
auderr, [6-15](#)
audhist, [6-20](#)
audinst, [6-22](#)
audrec, [6-24](#)
audstat, [6-31](#)
authdef, [6-34](#)
bslinfo, [6-63](#)
btihdev, [6-36](#)
btparm [G2-R6.3, R6.5-R6.8], [6-38](#)
btparm [R21.1 & R21.3], [6-50](#)
btparm [R21.5 & Later], [6-57](#)
btparm [R6.4 Only], [6-44](#)
bxslloc, [6-64](#)
bxslrmt, [6-69](#)
cdopt, [6-71](#)
cftinfo, [6-74](#)
chopt, [6-75](#)
ciopt, [6-78](#)
classdef, [6-81](#)
cpblx3, [6-83](#)
cpppone, [6-92](#)
dbinfo, [6-99](#)
dciopt, [6-107](#)
device, [6-110](#)
dfcopt, [6-117](#)
dst, [6-121](#)
duopt, [6-126](#)
duoptdyn, [6-128](#)
eaiopt, [6-130](#)
ecdorg, [6-137](#)
errcount, [6-141](#)
errthld, [6-144](#)
ftaminopt, [6-147](#)
ftamopt, [6-149](#)
ftamuser, [6-152](#)
getty, [6-158](#)
gfsappl, [6-162](#)
gfscom, [6-173](#)
gfsinit, [6-193](#)
gfsresp, [6-202](#)
gfssec, [6-211](#)
klist, [6-220](#)
logdev, [6-222](#)
mdct, [6-229](#)
mhdopt, [6-234](#)
nphopt, [6-237](#)
objectid, [6-252](#)
pphopt, [6-254](#)

ECD forms (Continued)

pphuser, [6-257](#)
rcvparams, [6-261](#)
runa, [6-263](#)
scsdbody, [6-265](#)
scsdmap, [6-268](#)
scsdopt, [6-272](#)
simcntl, [6-273](#)
sphopt, [6-277](#)
splrinfo, [6-279](#)
tphopt, [6-283](#)
tsapinfo, [6-287](#)
ttopt, [6-292](#)
ucb, [6-299](#)
ucbdyn, [6-322](#)
ularp, [6-326](#)
workhrs, [6-330](#)
ECD/SG forms,
dbdump, [6-359](#)
dbload, [6-360](#)
prtlist, [6-361](#)
trabort, [6-362](#)
trbegin, [6-363](#)
trend, [6-364](#)
ecdorg, [6-137](#)
Emergency Action Interface Option, [4-25](#) 6-130
Equipment Locations, [xv](#)
errcount, [1-8](#) [6-141](#)
error categories for audits, [6-15](#)
error codes, RC/V low-level access fatal, 8-1
error condition, [4-20](#)
error counter, [1-8](#)
error messages,
 acknowledgment, [4-20](#)
 growth space, [4-28](#)
 number of, [4-19](#)
 reading error messages, MSG
 keyword, [4-20](#)
 reference to list of fatal messages, [4-28](#)
 view or loose, [4-20](#)
 viewing, [4-20](#)
 where displayed, [4-16](#)
error threshold, [1-8](#)
errthld, [1-8](#) [3-1](#) [6-144](#)
ESCAPE, [6-4](#)
example RC/V session, [5-8](#)
execute-only form, [4-15](#) [4-22](#) [4-26](#)

F

fatal error codes, RC/V low-level access, [8-1](#)
fatal error file, [4-26](#)
fatal RC/V errors, [4-27](#)
field name, [4-21](#) [4-23](#)
field names, reference, [4-18](#)
field numbers, reference, [4-18](#)
field update, [1-1](#)
file manager, [2-3](#)
file partition, [3-10](#)
file partitions, [1-11](#)
file system, [3-10](#)
file systems, [1-11](#)
Finding all links to a specific form, [5-5](#)
finding all links to a specific form, [5-5](#)
finding all UCBs under a specific UCB, [5-4](#)
finding the RID of a specific form, [5-5](#)
flat-file, placing an ECD, [5-5](#)
form editor, [4-7](#)
form name, [4-14](#)
form navigation commands, [4-8](#)
form operation, [4-20](#)
form operation prompt, [4-9](#)
form selection prompt, [7-4](#)
formform, [7-14](#)
forms RC/V, [4-19](#)
forms, RC/V, general information, [6-1](#)
FORMTYPE option, [4-2](#)
fp, [1-11](#) [6-333](#)
fpb, [1-11](#) [6-337](#)
fragmented database space, reclaiming,
 dbload, [5-3](#)
fs, [1-11](#) [6-339](#)
fsb, [1-11](#) [6-344](#)
ftaminopt, [6-147](#)
ftamopt, [6-149](#)
ftamuser, [6-152](#)

G

general information, [6-1](#)
getty, [6-158](#)
gfsappl, [6-162](#)
gfscom, [6-173](#)
gfsinit, [6-193](#)

gfsresp, [6-202](#)
gfssec, [6-211](#)
growing and degrowing IOP devices, [7-4](#)
growth, [7-1](#)
growth space, [4-28](#)

H

hardware faults, [1-8](#)
help messages, [4-5](#) [4-7](#) [4-12](#)
HEX, [6-4](#)
HEX2D, [6-4](#)
Hexadecimal Notation, [xv](#)
high-level form, [7-2](#)
high-level forms processing acknowledgment
 messages, [7-5](#)
high-level forms to high-level processes,
 correlating, [7-14](#)
high-level mode explicit execution, [7-2](#)
high-level RC/V feature, [7-1](#)
high-level RC/V feature and text RC/V, [7-2](#)
high-level RC/V forms, [7-4](#)
 dciadd, adding a DCI, [7-9](#)
 dcidel, deleting a DCI, [7-12](#)
 formform, [7-14](#)
 iopadd, adding an IOP, [7-17](#)
 iopdel, deleting an IOP, [7-19](#)
 ioppc, review logical PC linkages, [7-21](#)
 iopslots, review physical PC slots, [7-23](#)
 mtadd, add a magnetic tape, [7-25](#)
 mtcadd, add magnetic tape controller, [7-28](#)
 mtcdel, delete magnetic tape controller, [7-31](#)
 mtdel, delete a magnetic tape, [7-33](#)
 rcvparams, RC/V parameters, [7-35](#)
 sccadd, add SCC link, [7-37](#)
 sccdcl, delete Switching Control Center
 (SCC) link, [7-42](#)
 scsiadd, add SCSI peripherals, [7-45](#)
 sdladd, add synchronous data link, [7-49](#)
 sdlcadd, add synchronous data link
 controller, [7-59](#)
 sdlcdcl, delete synchronous data link
 controller, [7-61](#)
 sdlldel, delete synchronous data link, [7-63](#)
 ttyadd, add TTY, [7-67](#)
 ttycadd, add TTY controller, [7-72](#)
 ttycdel, delete TTY controller, [7-74](#)
 ttydel, delete TTY, [7-76](#)
 vfydfit, verifies data in default file, [7-78](#)

high-level RC/V forms listing, [7-8](#)
high-level RC/V mode, [7-2](#)
high-level RC/V tools,
 RCVECD command options, [7-1](#)
 VFYDFLT command, [7-3](#)
How to Comment on This Information
Product, [xvii](#)

I

incore changes, making permanent,
 activate, [5-5](#)
 incore database, [4-3](#)
incore ECD copy, [6-5](#)
Information commands, [4-11](#)
informational commands, [4-7](#)
inode, [2-1](#) [2-3](#)
input field, [4-8](#)
input message area, [4-2](#)
input position, [4-6](#)
interface, color RC/V terminal, [5-7](#)
invoking **RCVECD**, [4-6](#)
IOP complex, [3-12](#)
 direct user interface controller device, [3-16](#)
 direct user interface device, [3-21](#)
 IOP device, [3-13](#)
 magnetic tape controller device, [3-15](#)
 magnetic tape device, [3-19](#)
 maintenance TTY and TTY device, [3-17](#)
 maintenance TTY controller device, [3-14](#)
 ROP device, [3-19](#)
 scanner and signal distributor controller
 device, [3-16](#)
 switching control data link device, [3-20](#)
 Synchronous Data Link Controller
 Device, [3-15](#)
 Synchronous Data Link Device, [3-20](#)
 TTY controller device, [3-14](#)
IOP device, [3-13](#)
IOP devices, growing and degrowing, [7-4](#)
IOP related devices,
 degrowth, [7-1](#)
 growth, [7-1](#)
IOP, listing all devices under, [7-7](#)
iopadd, adding an IOP, [7-17](#)
iopdel, deleting an IOP, [7-19](#)
ioppc, review logical PC linkages, [7-21](#)
iopslots, review physical PC slots, [7-23](#)

J

journal file, [4-5](#)

K

key field value, [4-21](#)
key-based forms, [6-349](#)
key-based forms instances, printing of, [6-361](#)
keystrokes file, RC/V session, [4-4](#)
keyword, [4-19](#)
keyword data pairs, [4-20](#)
keyword units, [4-19](#)
klist, [6-220](#) [6-349](#)
klist, listing keys, [5-3](#)

L

list navigational commands, [4-11](#)
list value commands, [4-10](#)
listing all devices under an IOP, [7-7](#)
listing and modes of operation, [6-1](#)
listing keys, *klist*, [5-3](#)
listing UCBs under a specific IOP, [5-4](#)
Listing UCBs under a specific IOP, *dbinfo*
form, [5-4](#)
LLA fatal error codes, [8-1](#)
logdev, [1-4](#) [2-1](#) [2-3](#)
logdev, [3-10](#)
logdev, [3-11](#) [3-17](#) [3-18](#) [6-222](#)
logical configurations,
 3B20D and 3B21D computers, [1-1](#)
 peripheral hardware, [1-1](#)
logical device, [1-4](#) [2-1](#) [2-2](#)
Logical Device Information, [6-222](#)
logical size, [1-11](#)
Logical-to-physical device translation, [1-2](#) [1-4](#)
logical-to-physical device translation, [2-1](#)
low-level access fatal error codes, [8-1](#)
low-level ECD form default files, [7-7](#)
low-level RC/V mode, [7-2](#)

M

machine instruction, [3-5](#)
magnetic tape controller device, [3-15](#)
magnetic tape device, [3-19](#)
main store controller device, [3-5](#)
maintenance TTY and TTY device, [3-17](#)
maintenance TTY controller device, [3-14](#)
maintenance, performing routine, [5-6](#)
Mandatory Customer Information, [xvii](#)
Manual,
 Organization, [xii](#)
 Purpose, [xi](#)
MASC device, [3-5](#)
MAXINT, [6-4](#)
mdct, [1-4](#) [2-1](#) [2-3](#) [6-229](#)
mhdopt, [6-234](#)
MKDSK, [6-1](#)
MML terminals, [4-1](#)
mnemonic name, [1-11](#) [1-12](#)
mnemonics definitions, [6-4](#)
Modifying RC/V database items, [5-1](#)
moving databases, [5-6](#)
moving head disk device, [3-10](#) [3-11](#)
Moving Head Disk Option Block Record, [6-234](#)
MT device, [3-19](#)
mtadd, add a magnetic tape, [7-25](#)
MTC device, [3-15](#)
mtcadd, add magnetic tape controller, [7-28](#)
mtcdel, delete magnetic tape controller, [7-31](#)
mtdel, delete a magnetic tape, [7-33](#)
MTTY and TTY Special Device File, [3-18](#)
MTTY device, [3-17](#)
MTTYC device, [3-14](#)

N

name, [1-11](#)
Navigational Commands, [4-9](#)
noncontinued line, [4-20](#)
nonexistent form, [4-16](#)
nphopt, [6-237](#)
null, [6-4](#)
number of, [4-19](#)

O

objectid, [6-252](#)
operating system generation information, [1-10](#)
option blocks, [1-4](#) [1-6](#) [1-8](#)
Organization, [xii](#)
Organization, Manual, [xii](#)
Output Message Database, [6-282](#)

P

pack, [1-11](#) [6-350](#)
pack data field, [6-4](#)
parameters, tunable for DAP, [6-74](#)
partitions, [1-11](#)
PDS terminals, [4-1](#)
peripheral hardware, [1-1](#)
permissions, modifying databases, [1-12](#)
PF keys states,
 display position, [4-6](#)
 input position, [4-6](#)
pphopt, [6-254](#)
pphuser, [6-257](#)
Printing forms, prtlist, [5-3](#)
printing key-based forms instances, [6-361](#)
process, [6-352](#)
process forms, [4-20](#)
process parameters, [4-20](#)
prompt, form selection, [7-4](#)
prtlist, [6-361](#)
prtlist, printing forms, [5-3](#)
Purpose, [xi](#)
Purpose, Manual, [xi](#)

Q

QCHAR, [6-4](#)

R

RC FORM operations, [4-22](#)
RC/V craft shell interface, [4-2](#) [4-3](#)
RC/V database, modifying items, [5-1](#)
RC/V ECD and SG forms, listing, [6-1](#)
RC/V feature, high-level, [7-1](#)
RC/V form, review only, [6-20](#) [6-31](#) [6-128](#)
[6-141](#) [6-268](#)
RC/V forms,
 3BGEN, [6-1](#)
 ECD, [6-1](#)
 general information, [6-1](#)
 listing and modes of operation, [6-1](#)
 MKDSK, [6-1](#)
 SG, [6-1](#)
RC/V forms, listing of high-level, [7-8](#)
RC/V forms, review-only, [7-7](#)
RC/V functions, [4-3](#)
RC/V LLA fatal error codes, [8-1](#)
RC/V low-level access fatal error codes, [8-1](#)
RC/V messages, SCC-originated, [4-19](#)
RC/V mnemonics,
 ESCAPE, [6-4](#)
 HEX, [6-4](#)
 HEX2D, [6-4](#)
 MAXINT, [6-4](#)
 null, [6-4](#)
 QCHAR, [6-4](#)
RC/V parameter information, [6-261](#) [6-353](#)
RC/V session, example, [5-8](#)
RC/V session, keystrokes file, [4-4](#)
RC/V terminal, black and white, [5-8](#)
RC/V Terminal, color, [5-8](#)
RCVECD command options, [7-1](#)
RCVECD, invoking, [4-6](#)
rcvparams, [6-261](#) [6-353](#)
rcvparams form, command line options, [4-4](#)
rcvparams, RC/V parameters, [7-35](#)
rcvparams, setting session parameters, [4-6](#)
reading error messages, MSG keyword, [4-20](#)
Reason for Issue, [xii](#)
Reclaiming fragmented database space,
 dbload, [5-3](#)
record identifier, [2-2](#)
recovery actions from errors, [1-9](#)
reference to list of fatal messages, [4-28](#)
Related Information Products, [xvi](#)
Remote Application Communication, [6-152](#)

Required Field, [4-8](#)
review-only form, [4-15](#)
review-only RC/V form, [6-20](#) [6-31](#) [6-128](#)
[6-141](#) [6-268](#)
RID, [2-2](#)
RID, finding, [5-5](#)
RID, finding for a specified form, *dbinfo*
 form, [5-5](#)
Root ECD, [6-5](#)
ROP, [3-19](#)
ROP device, [3-19](#)
ROP Devices, [2-3](#)
routine maintenance, performing, [5-6](#)
runa, [6-263](#)

S

Safety Labels, [xv](#)
SAT device, [3-6](#)
scanner and signal distributor controller
 device, [3-16](#)
sccadd, add SCC link, [7-37](#)
sccdel, delete Switching Control Center (SCC)
 link, [7-42](#)
SCDL device, [3-20](#)
SCH device, [3-8](#)
SCSD Scan Point Map, [6-268](#)
scsdbody, [6-265](#)
SCSDC device, [3-16](#)
scsdmap, [6-268](#)
scsdopt, [3-16](#) [6-272](#)
SCSI bus, [3-9](#)
SCSI or SMD MHD devices, [3-11](#)
scsiadd, add SCSI peripherals, [7-45](#)
SDL device, [3-20](#)
sdladd, add synchronous data link, [7-49](#)
SDLC device, [3-15](#)
sdlcadd, add synchronous data link
 controller, [7-59](#)
sdlcdel, delete synchronous data link
 controller, [7-61](#)
slddel, delete synchronous data link, [7-63](#)
SDP address space, [1-2](#) [1-13](#)
serial channel device, [3-8](#)
Session Layer Local Users List, [6-64](#)
Session Layer Remote Users List, [6-69](#)
setting session parameters, *rcvparams*, [4-6](#)
SG, [6-1](#)
SG and ECD RC/V forms, listing, [6-1](#)

SG forms,
 fp, [6-333](#)
 fpb, [6-337](#)
 fs, [6-339](#)
 fsb, [6-344](#)
 klist, [6-349](#)
 pack, [6-350](#)
 process, [6-352](#)
 rcvparams, [6-353](#)
 sgen2, [6-355](#)
 share, [6-358](#)
sgen2, [6-355](#)
share, [6-358](#)
Signal Designations, [xv](#)
simcntl, [6-273](#)
skeleton databases, [4-2](#)
software demand paging, [1-2](#) [1-11](#)
special device file, [1-4](#)
Special Device File, [3-18](#)
special device files, [2-1](#)
sphopt, [6-277](#)
split-screen effect, [3-17](#)
splinfo, [6-279](#)
spooler message, [4-20](#)
standard *ucb* default values,
 DSCH device, [3-8](#)
store address translator device, [3-6](#)
substitute mode, [4-18](#)
switching control data link device, [3-20](#)
Synchronous Data Link Controller Device, [3-15](#)
Synchronous Data Link Device, [3-20](#)
syntax, [4-19](#)
syntax, RC form operation, [4-22](#)

T

terminal control structure for control display
 terminal, [6-71](#)
terminal interface for color, RC/V, [5-7](#)
terminal, black and white, [5-8](#)
terminal, color, [5-8](#)
text RC/V, [4-19](#)
text RC/V session, [4-19](#)
The **LOADF3B** Command, [4-28](#)
tphopt, [6-283](#)
trabort, [6-362](#)
trabort, aborting a transaction, [5-2](#)
transaction begin form, *trbegin*, [5-9](#)
transaction end form, *trend*, [5-13](#)

transaction, RC/V, [5-1](#)
trbegin, [4-14](#) [4-15](#)
trbegin, [6-363](#)
trend, [4-14](#) [4-15](#)
trend, [6-364](#)
trend, transaction end form, [5-13](#)
tsapinfo, [6-287](#)
ttopt, [6-292](#)
TTY controller device, [3-14](#)
TTY device, [3-18](#)
ttyadd, add TTY, [7-67](#)
TTYC device, [3-14](#)
ttycadd, add TTY controller, [7-72](#)
ttycdel, delete TTY controller, [7-74](#)
ttysel, delete TTY, [7-76](#)
tunable parameters for DAP, [6-74](#)
typical craft shell commands, [4-3](#)
typical RC/V transaction, [5-8](#)

U

ucb, [1-4](#) [1-5](#) [1-7](#) [1-8](#) [2-1](#) [2-3](#)
UCB, [3-1](#)
ucb, [6-299](#)
ucb current error counts, [6-141](#)
ucb default values, standard,
 CSU device, [3-6](#)
 CU, [3-4](#)
 CU complex, [3-5](#)
 DMA device, [3-7](#)
 DUI device, [3-21](#)
 DUIC device, [3-16](#)
 IOP device, [3-13](#)
 MASC device, [3-5](#)
 MT device, [3-19](#)
 MTC device, [3-15](#)
 MTTY device, [3-17](#)
 MTTYC device, [3-14](#)
 ROP, [3-19](#)
 SAT device, [3-6](#)
 SCDL device, [3-20](#)
 SCH device, [3-8](#)
 SCSDC device, [3-16](#)
UCB default values, standard,
 SCSI or SMD MHD devices, [3-11](#)
ucb default values, standard,
 SDL device, [3-20](#)
 SDLC device, [3-15](#)
 TTY device, [3-18](#)

ucb default values, standard (Continued)
TTYC device, [3-14](#)
utility circuit device, [3-7](#)
ucbdyn, [6-322](#)
UCBs, 2-level naming convention, [6-299](#)
UCBs, finding all under a specific UCB, *dbinfo*
form, [5-4](#)
ULARP, [1-2](#) [1-9](#)
ularp, [6-326](#)
update operation, [4-16](#)
utility circuit device, [3-7](#)

V

verifying low-level ECD form default values, [7-7](#)
VFYDFLT command, [7-3](#)
vfydflt, verifies data in default file, [7-78](#)
view or loose, [4-20](#)
viewing, [4-20](#)

W

warning message, aborting a transaction, [5-2](#)
where displayed, [4-16](#)
workhrs, [6-330](#)