297-8991-304

DMS–100 Family

Software Delivery

ONP SNSE to SuperNode/ENET Conversion

BASE05 and up   PRELIMINARY Issue 01.09   February 1998

NORTEL
NORTHERN TELECOM

This page intentionally blank

DMS—100 Family

Software Delivery

ONP SNSE to SuperNode/ENET Conversion

Publication number: 297-8991-304
Document status: PRELIMINARY
Document release: 01.09
Date: February 16, 1998

This page intentionally blank

Publication history

February, 16 1998

Upissued to 01.09 to include steps to allow two-night process (as indicated by change bars)

June 4, 1997

Upissued to 01.08 to include additional steps to “ACTDUMP” SN tables and to matecopy installer files (as indicated by change bars)

May 6, 1997

Upissued to 01.07 to include additional MOP changes in PreSwact and SWACT/PostSWACT

April 18, 1997

Upissued to 01.06 to include process updates, revised instructions for migrating LIMs from SNSE to SN without referencing IM 78-5144

March 18, 1997

Upissued to 01.05 to include process updates

November 20, 1996

Upissued to 01.04 to reference document *ONP LPP Activity (IM 78-5144)*

October 7, 1996

Preliminary Issue 01.03, first use of new NTP number

July 10, 1996

Preliminary Issue 01.02

May 13, 1996

ONP SNSE to SuperNode/ENET Conversion, DRAFT Issue 01.01

This page intentionally blank

About this document

This document provides the method of procedure for the Software Application and ONP/Cutover phases of a DMS SuperNode SE (SNSE) to full DMS SuperNode Conversion (retrofit). These procedures also support a combined network upgrade from SNSE ENET to SuperNode ENET. (A combined upgrade from JNET to JNET/ENET will be supported in separate documents.)

The *ONP SNSE to SN/ENET Conversion MOP* included in this document utilizes One Night Process (ONP) software delivery tools and methods.

This document supports a software application only for SNSE Upgrades that meet the following requirements:

Supported

- DMS SuperNode SE (SNSE) 16K ENET/LIS converting to a full SuperNode ENET/LPP
- Same-to-same software levels
- Core currently operating on a Base05 platform PCL (for example LEC004) or higher, or for international on BCS36 or BCS40i
- LIS to LPP Upgrade
- LPP to LPP Upgrade (limited to maximum of 3 LPPs)

Not supported

- Combined upgrade from JNET to ENET (will be a separate document)
- Combined upgrade from JNET to JNET (will be a separate document)
- SNSE with 32K ENET (if later supported, will be a separate document)
- Upgrade from 3 LPPs and LIS/FLIS to 3 LPPs and LIS/FLIS
- Additional PMs may be installed as part of this procedure, but they will not be inservice until after Core upgrade is complete.

How this document is organized

The information in this document is organized in the following manner.

The *Introduction* chapter briefly describes how the SNSE to SuperNode software application and cutover works. It also explains the use of the Method of Procedure (MOP) sections in performing a software delivery. At the end of the chapter is list of general precautions for the MOP users.

The *Site preparation overview* chapter provides an overview of Telco/Carrier administrative and site responsibilities. This includes important planning and pre-cutover information.

This is followed by the *ONP SNSE to SN/ENET Conversion MOP* section which contain detailed procedure modules with steps to prepare for and deliver the new software load.

Appendix A: Command Summaries provides information on using the ONP features and commands: TABAUDIT, TABXFR, and BCSUPDATE. This section contains command syntax and example console sessions for these and other commands used throughout the software delivery process.

Appendix B: Supplementary Procedures contains additional procedures which may be referenced during the software delivery process.

Appendix C: Test Call Scripts provides generic guidelines for creating a test call plan for verifying the new software load. The site will be required to fill-in the test plan and test all applicable call types prior-to and following the software application.

Related documents

The Northern Telecom *Installation Method* documents specify Installation tasks that must be completed prior to and during the SNSE to SuperNode conversion. Method 78-54145 *SuperNode SE to SuperNode Upgrade Planning* contains an overview of the Conversion and a checklist of activities that must be completed and confirmed by the Nortel Installation and Operating Company representatives. That document also lists all the Installation Methods pertaining to the Conversion.

During the SNSE to SuperNode conversion the Nortel Installer will have available the *ONP LPP Activity* (IM 78-5144) document for migrating LIM from SNSE to SN . Parallel instructions are included in the PreSWACT and PostSWACT sections of this *Conversion MOP* .

A method of converting from NT40 to SuperNode is described in the *ONP SuperNode Conversion (ONPC)* ; the overall process for converting from SNSE to SN is similar.

This page intentionally blank

Contents

This page intentionally blank

Introduction

About the software delivery

The *ONP SNSE to SuperNode/ENET Conversion* details the steps necessary to perform hardware and software modifications required for a SuperNode SE (SNSE) to SuperNode conversion using the One Night Process (ONP) software delivery methods. The procedures contained in this document utilize the ONP software delivery tools and program modules to perform the front-end software conversion.

The events of the ONP can be divided into two main areas. The first area is the data transfer using TABXFR. It includes the table data move functions sometimes referred to as *dump and restore*. For a conversion, the TABXFR Retrofit tools will be used which are available on the TAS Non-Res tape. The second area is the software application using BCSUPDATE, which includes all the required application functions needed to activate the new software. The activation of the new software load on the SuperNode occurs with a Switch of Activity (SWACT), also called the *Cutover*. BCSUPDATE oversees the pre- and post-SWACT steps. The hardware “cutover” from the SNSE to the new SuperNode occurs simultaneously with the front end SWACT.

Typically, the *dump and restore* steps can be performed on the morning of the scheduled conversion. When this phase begins an office *data freeze* will go into effect . This mandates certain Data Modification Order (DMO) restrictions. Then, on the night of the conversion, the remainder of the software delivery (and cutover) can be completed in about 8 hours. The data freeze ends after the SWACT to the new SuperNode.

Using the MOP

First verify you are using the proper section of the MOP. Perform all procedures in the order given. The timeframe and responsibility for performing a step are indicated at the start of each section.

Throughout the document the term "Installer" refers to a Northern Telecom Installation Engineer at the DMS site. The term “Applicator” means the

Software Delivery Engineer who is certified to perform the conversion from SNSE to SuperNode (usually by way of remote dial-up). The term “Site” refers to Telephone/Carrier company personnel on site who can assist the Applicator during the conversion. The terms “conversion” and “retrofit” are often used interchangeably, as are the terms “application” and “cutover.”

Note: Throughout this document the term “to_PCL” refers to the PCL level of the new software load; and “from_PCL” refers to the old (or current) PCL level.

The term “Test Call Scripts” refers to the verification calls as predefined by the Telephone/Carrier Operating Company. These are test calls to be performed after activating the new software load in order to determine acceptance of the new load and operation of the SuperNode.

General precautions

Notice that the following precautions are subject to change throughout the conversion process. In addition, certain procedures may be changed during process development. Therefore, subsequent issues of the *ONP SNSE to SuperNode/ENET Conversion* will be reissued as needed as development progresses.

- Personnel responsible for performing any of the steps in this MOP must be thoroughly familiar with the complete procedure before starting it.
- Installation personnel will work with the Applicator at key points during the procedure to perform Installation tasks.
- Hardware problems are contributors to conversion aborts and reschedules; therefore, particular attention should be made to testing all memory cards and to monitoring CM and MS logs prior to the SuperNode conversion.
- It is recommended that site personnel responsible for assisting Northern Telecom in the actual process should review all sections of this document to ensure designated activities will be completed prior to or during the SNSE to SuperNode conversion process.

Site preparation overview

This section describes the site preparation tasks require to prepare the DMS for a front-end conversion from the SNSE-to-SuperNode. The detailed procedures to be performed by on-site Telco personnel are provided in the next chapter under *Site preparation procedures*.

Time-line of events

The following information provides the site with a summary of activities which will be performed prior to the SuperNode conversion. Personnel involved in the preparation, or the conversion process, or both, must be thoroughly familiar with this section.

Northern Telecom recommends that the actual conversion to SuperNode be scheduled to take place during low traffic periods to minimize any undesirable impact to the office.

Timeline overview

All references to days indicate the number of calendar days prior to the conversion from the SNSE to the SuperNode (IS date).

Table A — Process timeline

Days before the Conversion	**Site Preparation Activity**
— Determined by telephone co.	• Administrative functions (notification to operator services, control centers, repair service bureaus, etc.)
— 45 to 12 days to conversion	• First tape and documentation shipment arrives at site • Site starts checking CM logs daily • Begin loading peripherals with latest PM loads

—continued—

Table A — Process timeline (continued)

Days before the Conversion	Site Preparation Activity
— 15 days	• Ensure all needed hardware is installed and verified. • Nortel Patch Delivery downloads ONP process patches to SNSE. These must be applied to the SNSE prior to the Dump/Restore. • Begin data consistency checks (TABAUDIT).
— 10 days	• Test Image for data consistency check (only if requested). • Ship tape to Northern Telecom if required.
— 7 to 4 days	• Perform memory retention test (SNSE and SuperNode). • Perform CPU routine exerciser (REX) test (both). • Perform processor stability tests (both).
— 6 days	• Copy all SNSE and SuperNode files in SFDEV to backup tape. • If office is equipped with disk drive units (DDUs), on PMLOAD disks, erase all unwanted and unneeded disk files. Ensure latest PM loads and XPM patches are restored to proper disk volumes. • Monitor SNSE & SuperNode CM/MS logs through conversion day.
— 3 days	• Verify PMs and XPMs are loaded, patched and working. • Verify dial-ups for SNSE and SuperNode. • Verify SFDEV files are copied to tape for backup. • Ensure all Load tapes and Tools tapes are on-site. • Verify site is familiar with the procedures. • Verify site has prepared test calls. • Obtain name of site representative with authority to decide in case of an ABORT. • Verify patches will be down-loaded by Nortel patch department.
— 3 to 2 days	• Site receives final tape and documentation shipment. • Review all documentation. • Assure all (Site Prep & Installation) checklists are completed.
—	

Day of conversion	• Dump office image to tape or SLM disk for backup. • Perform office data dump (hardcopy of office data) on SNSE. • Nortel Dump/Restore will dial in at approximately 0500 site time. • D/R Engineer will load SuperNode with newest UDF image. • D/R Engineer will load Retrofit modules/execs on both SNSE & SN.

—continued—

Table A — Process timeline (continued)

Days before the Conversion	Site Preparation Activity
	• D/R Engineer will perform data move from SNSE to SuperNode. • Begin office data FREEZE until night of conversion.
Conversion night	• Conversion Application phase, Nortel Applicator will dial in at approximately 1000 site time to start Preswact. H/W Cutover occurs simultaneously with the software Application SWACT.

Table B — Conversion activities

NODE (front-end)	Conversion Activity
SNSE	Preliminary set-up and login
SN	Load new undatafilled image into the SuperNode
SNSE	Dump procedure on the SNSE
SN	Restore procedure on the SuperNode
SNSE	Pre-conversion on the SNSE
SN	Pre-conversion on the SuperNode
SNSE and SN	Conversion (switch of activity) to the SuperNode

SN	Post-conversion activities

Planning activities

It is imperative that the administrative functions outlined below are reviewed by all offices as soon as this document is received.

Administrative functions

The following administrative functions need to occur for all offices well in advance of the scheduled software delivery.

- In offices equipped with TOPS-OC (Operator Centralization), please refer to feature NC0152 (Host/Remote Networking by Queue Type) in the *Translation Guides*. This feature describes the BCS/PCL software delivery strategy for TOPS-OC offices.
- Since specific Telco policies may exist within various locations, an estimated time of SWACT should be established between Telco and the Nortel software delivery engineer.

  Note: Northern Telecom recommends that the front-end activity switch (SWACT) be scheduled to take place during low traffic periods to minimize the impact on the office.
- Advanced notification of the software update must be provided by the site to operator services, service control centers, repair bureau, and other special services.
- Advise the data transferal regional coordinator (or equivalent) when the software delivery update will occur.
- Offices equipped with Centralized Automatic Message Accounting (CAMA) or Local Automatic Message Accounting (LAMA) will arrange for the validation of an Automatic Message Accounting (AMA) test tape with the site billing center. Such testing may also include Station Message Detail Accounting (SMDR) or Other Common Carrier (OCC). The site billing center must be informed of this requirement four weeks prior to the software update. This test should be performed during POSTSWACT activities (execution of Test Call Scripts). Ensure that the *AMADUMP User’s Guide* (NTP 29C-1001-119) is readily available.
- Offices equipped with DPP or BMC actively collecting billing information may arrange with the downstream processing center to poll the billing information during *PRESWACT* and, optionally, during *POSTSWACT*.

Warnings

Ensure no additional hardware changes will be in progress during the SNSE to SuperNode Conversion. These activities are prohibited during the software Application and Cutover phases. Any affected hardware must be made INB (installation busy), and any further software changes must cease. Such activities would include, *but are not restricted to*, any of the following:

- Network extensions
- Memory extensions
- Peripheral additions or deletions

Special activities

The operating company may request special changes to office data which must be done separately from the SNSE to SuperNode Conversion. These requests must be identified ahead-of-time and the job scheduled separately. Such activities, referred to as Customer Special Request workarounds, can include *but are not restricted to*, any of the following:

- deleting entries in table LINEATTR or changing certain restricted fields
- LGC to LTC (or LTC to LGC) conversion

  Note: This request requires Telco to complete a workaround after CC SWACT. In *Appendix B* refer to Procedure B-2: Converting one PM to another.
- XPM to FXPM upgrade or Inservice FXPM relocation
- down-sizing of a switch involving deleting peripherals
- changing an MDC group to allow the installation of an attendant console
- changing table TRKGRP field SEL SEQ from MIDL or LIDL to ASEQ or DSEQ
- changing number of entries in TOPSDEV and TOPSPOS if certain pseudo-cllis or office parm TOPS_NUM_TRAFFIC_OFFICES has been maxed-out
- NPA splits
- increasing amount of ICI codes per ATTCONS in table CUSTCONS
- changing or deleting carrier names in TOPEACAR
- 100/200 splits
- deleting or changing Serving Translation Scheme (STS)
- DCH sparing in ISDN offices
- changing or deleting remote site names

Pre-application site activities

The site personnel responsible for assisting Northern Telecom during the software delivery should become familiar with all sections of this document to ensure designated activities are completed in a timely manner.

- Upon receiving this document, the Telco site should read over the entire MOP to get an idea of the activities that will be performed. To help in scheduling the needed site preparation work, consult the Confirmation Letter or a Site Notification Package for an exact schedule of events.
- Next, turn to the *Site preparation procedure* section of the MOP, and begin doing the procedures in that section.
- Site should continue performing all the procedures through the *Site responsibilities the day of the software delivery procedure* section, which completes the pre-application phase.

In addition to this document and related *Installation Methods*, the site must also be familiar with the *Peripheral Module Software Release Document* for the new PCL software release level.

Conversion hardware requirements

- 9-track tape drive (MTD) on both the SNSE and SuperNode front-ends
- Foreign exchange dial-ups into both the SNSE and SuperNode
- New SuperNode front-end installed and commissioned for service
- Master and remote switch boxes installed and tested

Returning tapes

After a software delivery has been completed, Telco may return for recycling the DATA CARTRIDGE TAPES for the previous software load. Keep all software tapes for the current PCL. Please do not return any 9-track magnetic tapes—recycle only data cartridges.

ONP SNSE to SN/ENET Conversion MOP

MOP contents

This page intentionally blank

Site preparation procedure (SNSE to SN/ENET)

Begin with this section when the First Shipment of tapes and documentation arrives at the office. Procedures may also be performed before the recommended completion interval, but must be completed by the completion interval.

Note that some procedures detailed in this document parallel those found in various Northern Telecom Installation Methods. This document is intended to work in tandem with the Installation Method and should not conflict with the procedures documented therein.

In order to qualify the office for a software upgrade, the procedures listed in the “Site Preparation Checklist” in this document (below) and the “Pre-Cutover Checklist” found in the Installation Method 78-5145 must be completed by the intervals listed.

> ***Note:*** **Failure to complete both checklists by the intervals listed will place the cutover date in jeopardy.**

Except where indicated otherwise, the Site Preparation procedures will be performed on the SuperNode SE (SNSE) switch.

Site Preparation Checklist

This checklist is provided to help assure that all site preparation steps have been completed. Please read the entire document before proceeding with the checklist. The completion intervals listed throughout this document should be observed to assure the cutover date.

It is required that the Customer Representative and the Nortel Installation Representative confirm that each step is completed.

Proc	**Description**	**Date Complete**	**NT**	**Site**
1	Responsibilities before pre-application checks			
2	Take image SNSE			
3	Route logs SNSE			
4	Processor tests SNSE			
5	Installer loads SuperNode with UDF Image			
6	Processor tests SuperNode (with Installer)			
7	Save site files			
8	Peripheral verification			
9	Table ACDGRP			
10	Fill in Test Call Scripts			
11	Using AutoTABAUDIT to run TABAUDIT			
12	Prepare test image			
13	Review table data checks results			
14	Restore CM and MS loads			
15	Preload both MSs			
16	Day zero checklist			
17	Patch verification			
18	Run DATADUMP			
19	FX switch access			
20	Network management control			

Procedure 1
Responsibilities before pre-application checks

1 **Site** Site is encouraged to verify all new software load and patch tapes received in a shipment.

Note: The following steps do not apply to a TAS NONRES tape which is used only by Nortel technical support to provide access to certain non-resident software tools.

a. INSERT (or MOUNT) and LIST each tape.

b. From the tape header or first file verify the tape is correct for the target load. For the PCL IMAGE tape also verify the image filename.
Verify a tape is good by listing the tape to the end without any errors.

c. If any problems are found notify your Nortel customer service representative immediately to request reshipment.

d. Keep the tapes on-site for use during the scheduled software update.

—continued—

Procedure 1
Responsibilities before pre-application checks (continued)

> **CAUTION**
> **Wait to upgrade the Message Switch until the new PCL IMAGE tape arrives on site**
>
> For MS preloading it is recommended that the MS be loaded using the new PCL IMAGE tape, and *not* the PM Loads (IN or STP) tape. This is because the MS load provided on the IMAGE tape contains a more current patch set. A recommended procedure for MS preloading is found in this MOP in the section “Updating loads in the Message Switch.”

2 **Site** When the PM load tapes arrive—Load and patch ALL PERIPHERAL MODULES with the new PM software according to the *Peripheral Module Software Release Document* (“Application Procedures” section). *Peripheral modules include all PMs, XPMs, DPP, MPC, and the various application processors associated with a DMS-SCP/STP/SSP such as the LPP (including EIUs, LIUs and LIMs) and the FP.*

Note: With the release of NA005B a new tool is available to automate a portion of the PM Upgrade Process. This tool is called PANTHER and is part of the PM Upgrade Manager Utility (a Non-res tool). Please refer to the *Peripheral Module Software Release Document*. In the “Overview of release” section of that document is an explanation of the PANTHER tool and how it is used to automate the upgrading of PMs.

---

3 **Site** Monitor front-end stability watching SNSE CM, MM, and MS logs through the day of the software delivery.

Procedure 2
Take image SNSE

1 **Site/ACT** Before beginning the front-end testing (processor tests), DUMP AN OFFICE IMAGE for backup—one SNSE image to SLM disk 1 (or to tape) and backup image to SLM tape cartridge.

Procedure 3
Route logs SNSE

This procedure ensures that specific logs are routed to an active log device and are not suppressed.

1 **Site/ACT**

```
> LOGUTIL

> LISTREPS SPECIAL
```

If specific logs are suppressed use

```
> RESUME <log>
```

If logs have threshold set use

```
> THRESHOLD 0 <log>
```

where <log> refers to specific CM, MS, SLM, and MM logs.

2

```
> LISTROUTE DEVICE <printer>
```

If critical logs are not routed use

```
> ADDREP <printer> <log>
> STOPDEV <printer>
```

Verify only critical logs are enabled on the device and are correctly routed.

3

```
> STARTDEV <printer>

> LEAVE
```

Procedure 4
Processor tests SNSE

To ensure front-end stability Site should complete the following tests before being contacted for the pre-application checks.

1 **Site** Ensure the CPUs are in SYNC and the inactive side is NOT jammed.

---

2 **ACT** Match the memory from the Memory level of the MAP.

```
> MAPCI;MTC;CM;MEMORY;MATCH ALL
> QUIT
```

---

3 **INACT** From the inactive Reset Terminal, jam the inactive CPU.

```
RTIF> \JAM
RTIF> YES
```
{for confirmation}

---

4 **ACT** Drop SYNC from the CM level of the MAP.

```
> DPSYNC
> YES
```
{for confirmation}

---

5 **INACT** Wait for the inactive CPU to return to flashing A1.

---

6 Test the CM stability with each of the following restarts on the *inactive Reset Terminal* ONLY.

a. **INACT** `RTIF> \RESTART WARM`

`RTIF> YES` *{for confirmation}*
Wait for a flashing A1.

b. **INACT** `RTIF> \RESTART COLD`

`RTIF> YES` *{for confirmation}*
Wait for a flashing A1.

c. **INACT** `RTIF> \RESTART RELOAD`

`RTIF> YES` *{for confirmation}*
Wait for a flashing A1.

---

7 **ACT** Test the memory cards from the Memory level of the MAP.

```
> MEMORY;TST ALL LONG
```
This test will take up to 15 minutes to test each memory card.

```
> YES
> QUIT
```
{for confirmation}

—continued—

Procedure 4
Processor tests SNSE (continued)

8 After completion of the tests, check the CM logs and verify that no CM112 logs have been reported during the test. If needed, resolve any problems and repeat step 7.

---

9 **ACT** SYNC the CPUs from the CM level of the MAP.

```
> SYNC
```

---

10 After receiving the “Synchronization Successful” message, verify no faults are displayed at the CM or Memory levels of the MAP (shows all dots and no Xs or fs).

---

11 **INACT** At the inactive Reset Terminal release the jam.

```
RTIF> \RELEASE JAM
```

---

12 **ACT** Switch activity of the CPUs from the CM level.

```
> SWACT
```

---

13 **INACT** Repeat steps 1 through 12 on the newly-inactive CPU.

---

14 Verify the CPUs remain in SYNC and the inactive side is NOT jammed.

---

15 **ACT** Match the memory from the Memory level of the MAP.

```
> MEMORY;MATCH ALL
```

```
> QUIT
```

---

16 **ACT** Perform a REX test long from the CM level. (Repeat with each CPU initially active.)

```
> REXTST LONG
```

```
> YES
```
{for confirmation}

CPU SYNC, Message Controller (MC), and Subsystem Clock (SSC) states will change. The SuperNode will be out of SYNC for at least 60 minutes.

---

17 **ACT** After completion of the test, verify the test results:

```
> QUERYCM REXRESULT
```

The CPUs should be back in SYNC with no REX alarms at the CM level or on the main MAP display header. If the test failed, contact the site supervisor to resolve any problems and repeat steps 16 and 17.

—continued—

Procedure 4
Processor tests SNSE (continued)

18 Repeat (with the other CPU active) steps 16 and 17.

---

19 **ACT** Perform an image test from the CMMNT level of the MAP.

```
> CMMNT
> IMAGE
> QUIT
```

---

20 After completion of the test, check for CM logs indicating pass or fail message. If test failed, clear the problem and repeat step 19.

---

21 **ACT** Busy the Slave MS from the MS level of the MAP.

```
> MS;BSY <x>
```

where <x> refers to the Slave MS (look under Clock field).

---

22 **ACT** Test the MS from the MS level.

```
> TST <x>
```

---

23 After completion of the test the results of the test are displayed. If the test failed, resolve any problems and repeat the previous step.

---

24 **ACT** Return the busied MS to service.

```
> RTS <x>
```

---

25 Wait 5 minutes to ensure the clocks are stable and to allow the hardware audit to run. Both MS should be inservice.

---

26 **ACT** Switch MS clock mastership.

```
> SWMAST
```

---

27 Wait an additional 10 minutes to allow MS clocks to completely stabilize.

---

28 Test the other MS by repeating steps 21 through 25.

---

29 **ACT** `> QUIT ALL`

After front-end Processor Tests are completed, continue to monitor logs until the Conversion. Take action to assure front-end stability should any problems arise. *An unstable processor could jeopardize the Cutover.*

Procedure 5
Processor tests SuperNode

Continue with this procedure *only after the SuperNode has been loaded by the Northern Telecom Installer*. When the First Shipment of tapes and documents arrives on site, the Installer will load the SuperNode using an undatafilled PCL image. (This image is considered the “Commissioning load.”)

Note: A procedure to load the SuperNode with the UDF image is in Appendix B. See procedure “SuperNode loading with UDF image.”

1. **Site/SN** It is recommended to repeat on the SuperNode the same Processor Tests that were performed above. However, the *testing must be coordinated with the Nortel Installer on-site.* ***Do not proceed without first clearing this with the Installer.***

2. When ready, *on the SuperNode switch* repeat **Procedures 2 through 4** above. These steps are exactly the same for either SNSE or SuperNode.

Procedure 6
Save site files

1 **Site/ACT** Copy all Telco/site-created files in store file (SFDEV) onto a labeled scratch tape. These files can be manually restored to SFDEV after the PCL application.
*Any patches and files downloaded for the s/w application need to remain where they were downloaded —**DO NOT ERASE THESE!***

Note: The purpose of this step is to clear SFDEV of any unnecessary files to increase space for ONP work.

Procedure 7
Peripheral verification

1 **Site/ACT** Verify the newest PM loads and patches are copied onto the appropriate PMLOAD disk volumes. If necessary, clear more disk space by erasing unwanted files.

> **CAUTION**
> **PM loads and their corresponding PM patches must reside on the same disk volume**
> This is a requirement for Auto Patching to work.

Note: Use the PMLOAD volumes that are normally used for loading and patching peripherals. (For example, for SL100 the Series II PM loads and patches are placed on separate disk volumes than the Series III PM loads and patches.)

2 Ensure peripheral modules (including DPPs) are loaded and patched with the newest PM software. (Refer to the *Peripheral Module Software Release Document*.)

Note: Procedures for preloading the MSs are in section *Updating loads in the Message Switch* of this MOP.

3 On the DDU used for primary billing collection (such as AMA SMDR OCC CDR), perform routine maintenance on the disk to ensure it is functioning properly. If excessive “bad blocks” are present, reformat the disk.

Procedure 8
Table ACDGRP

1 **App/ACT** Find all the holes in table ACDGRP and fill them with dummy tuples as follows. Otherwise, you may be unable to retrieve MIS reports from some ACDGRPs.

a.
```
> OMSHOW ACDGRP ACTIVE
```

b. Look for nonconsecutive keys. (Example: 0 2 3 5 6 has 1 and 4 missing.)

c. For any missing tuples, have translations personnel datafill dummy tuples in the key indexes. (This prevents wrong renumbering during the software update.)

d. Also provide datafill in table DNROUTE for each corresponding dummy tuple added in table ACDGRP.

Procedure 9
Fill in Test Call Scripts

1 **Site** Fill in and test the *Test Call Scripts*.

This is to provide a thorough test plan exercise for validating the new PCL. You will be asked to make your test calls after switching activity to the new software load.

TABAUDIT procedure

Perform this procedure 15 days before the software delivery date. This allows time to correct any table data problems that might be found. These procedure steps must be completed with all data errors resolved. This will ensure that the software application can begin on time and be completed with a minimum of interruptions.

The following AUTOTABAUDIT procedure is recommended over manual TABAUDIT because it is fast and will not occupy a terminal. (AUTOTABAUDIT runs in the background.)

If necessary, repeat TABAUDIT until all tables pass the data checks with no errors. Refer translation errors to your local translations department. Additional support, if required, can be obtained through your Northern Telecom regional representative.

CAUTION

Before attempting to correct any table errors be sure to review all TABAUDIT Bulletins and Notices.

These will identify any known problems or inconsistencies.

Note: The following procedure steps do *not* utilize all of the options available to the user. Additional information on TABAUDIT options is included in *Appendix A* under “Using TABAUDIT.”

- *Procedure 1* below is the *recommended* procedure for setting up AUTOTABAUDIT to run a scheduled TABAUDIT.
- Otherwise, to *manually* execute a TABAUDIT session, please refer to “Steps to execute manual TABAUDIT” in *Appendix B.*

CAUTION

TABAUDIT can take anywhere from 2 to 15+ hours to complete.

The total time will vary depending on the number and size of tables checked. AUTOTABAUDIT can be used to schedule the execution of TABAUDIT during user-defined time segments. Multiple sessions can be scheduled, but they cannot overlap.

CAUTION
TABAUDIT is not "image safe"
Look at table IMGSCHED before defining AUTOTABAUDIT. Do *not* schedule TABAUDIT to execute at the same time as Autoimage.

Procedure 1
Using AUTOTABAUDIT to run TABAUDIT (BCS36 and higher)

Note: It is beneficial to schedule AUTOTABAUDIT to *start after the completion of the AUTOIMAGE* and to *stop before the next scheduled AUTOIMAGE begins*. This should allow enough time for the AUTOIMAGE to complete.

1 **Site/ACT** Set-up for AUTOTABAUDIT.

a. Enter the automated level of the TABAUDIT increment (AUTOTABAUDIT) to enable the auto level commands.

```
> TABAUDIT
TABAUDIT:
```

```
> AUTO
AUTOTABAUDIT:
```

All commands issued from within this level apply to AUTOTABAUDIT. This level has its own includes and excludes list which must be specified from within the auto level. Two additional commands are TIMEFRAME and TERMINATE (explained below).

Note: AUTOTABAUDIT *included* and *excluded* lists do not reset when leaving the AUTOTABAUDIT level. The list remains until reset by the user.

b. Clear the *included* list of tables.

```
AUTOTABAUDIT:
> CLEAR INCLUDED
```

c. Clear the *scheduled* list of timeframes.

```
AUTOTABAUDIT:
> CLEAR SCHEDULE ALL
```

d. Define the list of tables to be tested.

```
AUTOTABAUDIT:
> INCLUDE ALL
```

This option will include all tables in the office.

Note: The *included* list of tables should include all the tables listed in table DART. The *excluded* list normally would not contain any tables.

—continued—

Procedure 1
Using AUTOTABAUDIT to run TABAUDIT (continued)

2 Define the days and times to run the TABAUDIT session.

The following steps define to AUTOTABAUDIT the timeframe *within* which TABAUDIT runs and the days *on* which TABAUDIT runs.

a.
```
AUTOTABAUDIT:
> TIMEFRAME <start time> [start date] <stop time>
  [stop date]
```

Note: A timeframe can not be less than 30 minutes. The date can be included as an option.

Example:

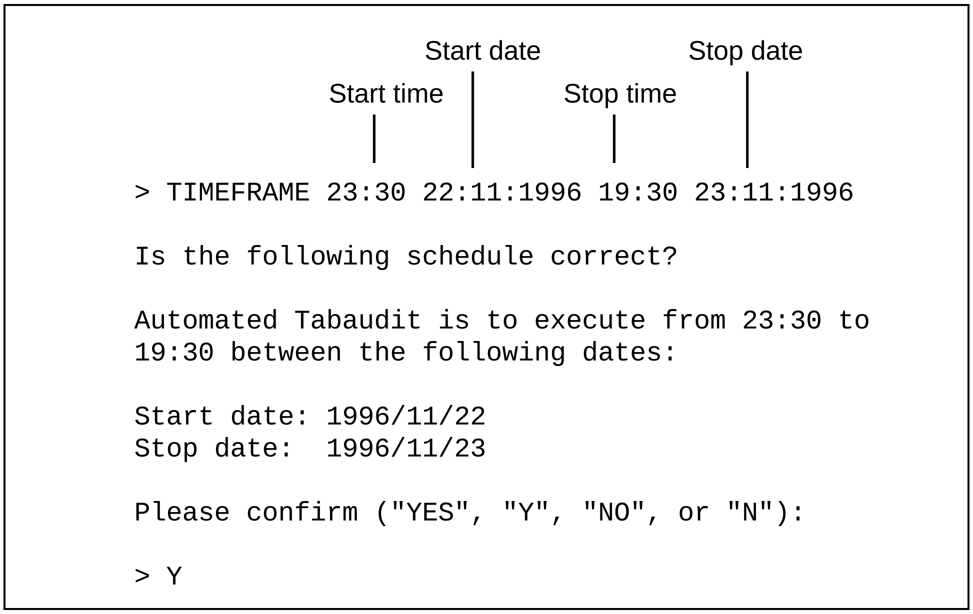
```
> TIMEFRAME 23:30 22:11:1996 19:30 23:11:1996
```

where	*23:30*	*is*	*start time*	*(hr/min)*
	22:11:1996	*is*	*start date*	*(d/mm/yy)*
	19:30	*is*	*stop time*	*(hr/min)*
	23:11:1996	*is*	*stop date*	*(dd/mm/yy)*

The above definition will schedule TABAUDIT to run for 20 hours from 23:30 p.m. to 19:30 p.m. on both November 22 and 23.

Note: Up to eight different sessions can be set up. Timeframe definitions, however, must not overlap one another.

Example timeframe for autotabaudit



```
> TIMEFRAME 23:30 22:11:1996 19:30 23:11:1996

Is the following schedule correct?

Automated Tabaudit is to execute from 23:30 to
19:30 between the following dates:

Start date: 1996/11/22
Stop date:  1996/11/23

Please confirm ("YES", "Y", "NO", or "N"):

> Y
```

—continued—

Procedure 1
Using AUTOTABAUDIT to run TABAUDIT (continued)

b. Verify AUTOTABAUDIT is set up and scheduled as intended.

```
AUTOTABAUDIT:
> STATUS
```

This displays the current AUTOTABAUDIT settings.

---

3 Activate the AUTOTABAUDIT schedule.

```
AUTOTABAUDIT:
> EXECUTE
```

This first shows a STATUS. If correct, confirm with “YES” when prompted.

Note: The scheduler will start AUTOTABAUDIT at the specified start time(s), and will stop testing at the specified stop time(s). Only the *included* tables will be tested in the order they are listed in table DART.

** TERMINATION—To stop the AUTOTABAUDIT session: from the AUTOTABAUDIT level, type the *TERMINATE* command.
This command halts AUTOTABAUDIT and resets the execution order of the tables back to the top of the included list.

The following example illustrates the correct use of the EXECUTE command of AUTOTABAUDIT.

Example of autotabaudit execute

```
AUTOTABAUDIT:
> EXECUTE

-----------------------------------------------------
|                                                   |
|             AUTOMATED TABAUDIT STATUS             |
|                                                   |
-----------------------------------------------------
|  Active Timeframe        |  Executing Timeframe   |
-----------------------------------------------------
|  Start        Stop       |  Start      Stop       |
|  Date         Date       |  Time       Time       |
-----------------------------------------------------
|  1996/11/22   1996/11/23 |  23:30       19:30     |
-----------------------------------------------------

Current time               : 1996/11/22 16:32:05
Automated Tabaudit         : Inactive

The following tables are INCLUDED

   From table ACTPATCH (0)                  to table SSRFORM (479)

The following tables are EXCLUDED
```

```
   No tables have been excluded.

Please confirm ("YES", "Y", "NO", or "N"):

> yes
```

4 After the AUTOTABAUDIT has finished running, obtain an error REPORT to see the results of the testing.

Note: Several useful options are available with the REPORT command. Type HELP REPORT to see the options.

```
TABAUDIT:
> REPORT ERRORS
```

The REPORT ERRORS command displays tuples that are in question. To see detailed information on why a tuple has failed, position on the table tuple in question and issue a CHECK command. This will display the verify proc messages that fail.

5 After reviewing reported table data errors, *correct all the errors*. Then execute TABAUDIT on the fixed tables to verify there are no further errors.

```
TABAUDIT:
> INCLUDE <table_name>
```

```
TABAUDIT:
> EXECUTE
```

```
TABAUDIT:
> REPORT <table_name>
```

Northern Telecom recommends that data integrity checking using TABAUDIT be made a regular and ongoing part of normal maintenance procedures. By using the automatic scheduling function this can be accomplished with a minimum of effort.

Certain Northern Telecom support areas ask offices to send in a "Test Image" for a dry run ONP. Only these customers need to perform the following procedure. Please check with your Nortel customer support representative.

NOTICE
If your office has successfully been using TABAUDIT and all errors are being corrected, then Telco may choose *not* to send in the test image for a dry run ONP. If this is the case, *do not* perform the following procedure.

Procedure 2
Prepare test image

The following procedure steps show how to prepare a "test image" to be used by Nortel for a dry run ONP.

Perform this procedure at 10 days prior to the software delivery date. ***Nortel must receive a test image by 8 days prior to the software delivery date to allow enough time to perform the dry run ONP.***

1 Site/ACT ***Verify user OPERATOR is permitted as follows in order to allow Nortel access to the system image.*** Nortel will load the test image into a captive switch to perform ONP validation.

a.
```
> SHOW USERS

NAME      PRIO STACK NRDEV LANGUAGE  PRIV

OPERATOR  4    5000   --   ENGLISH   ALL
```

b. If OPERATOR is not permitted as stated above, change with the "PERMIT" command.

c. ***Important:*** Change the PASSWORD to "OPERATOR" for the image. This is only temporary and can be changed back when done.

2 Take a system image to tape (or SLM cartridge) in order to capture current office system data. (This is a snapshot, NOT a "frozen image.")

3 Verify the image tape by either (1) listing the tape to the end and watching logs for "DEVICE ERROR" or other I/O messages, or (2) loading the image into the inactive side (LDMATE) to confirm the image will initialize.

4 Label the image tape as "TEST IMAGE" with the office name.

—continued—

Procedure 2
Prepare test image (continued)

5 *Ship the image tape* to Northern Telecom via “overnight” air express. Include in the shipment the name and telephone number of the Telco translations engineer who can help in resolving any data consistency issues.

- ***Please contact Nortel Applications to get instructions and the correct mailing address for your support area.***

Procedure 3
Review table data checks results

If there are any further problems reported with table data, Telco is responsible to make needed corrections and resolve all table problems by at least 2 days before the scheduled software delivery date.

CAUTION
Telco must make corrections and resolve all table problems by 2 days before to the software delivery.

Updating loads in the Message Switch

Begin this section as soon as the new PCL IMAGE tape arrives on site (typically two to three days before the software delivery).

Procedure 1 is to copy the new CM and MS loads onto an SLM disk. This is required in order to load the MS and to enable faster loading of the mate CM from SLM disk.

Procedure 2 is used to preload the MSs with the new MS load.

Note: When scheduled for a PCL to same PCL application (for example, LECB005 to LECB005) Telco may choose to NOT preload MSs as long as the present MS load is *patched current*.

Procedure 1
Restore CM and MS loads

Restore (that is, copy) the new CM and MS loads onto a SLM disk.

1 **Site/ACT** List the SLM tape cartridge with the new PCL IMAGE files (both _MS and _CM loads).

a. Place the cartridge into the SLM tape drive on the same side as the *inactive* CPU.

b. `> DISKUT NORET`

c. `> IT <`**`tape_device`**`>`
Inserts the tape into the inactive-side SLM, for example:
IT S00T or IT S01T

d. `> LF <`**`tape_device`**`>`
for example, LF S00T or LF S01T. May take up to one hour to list.

e. Verify the MS and CM load files are the correct ones to use.

Note: Filenames for _MS and _CM loads will vary. The MS Universal load, for example, will be of the form: MUC00004 <H#>_S1.

2 Select a SLM disk volume onto which to restore the new PCL IMAGE.

- The volume selected should not be on the same SLM with active DIRP billing.
- The volume should not be the same volume normally used to take images. (This is so that AUTOIMAGE won't fail for lack of disk space.)

If there is a problem completing this step, please contact the next level of support.

—continued—

Procedure 1
Restore CM and MS loads (continued)

3 Restore both the CM load and the appropriate MS load onto the selected SLM disk volume.

a. `> DISKUT`

b. If on a Base06 or higher PCL use

`> MFRESTORE FILE` <**disk_volume**><**tape_device**><**filename_cm**><**filename_ms**>
Restores both the CM and MS loads onto the SLM, where
disk_volume *is the SLM volume name*
tape_device *is the SLM tape unit*
filename_cm *is the CM load file name*
filename_ms *is the MS load file name*

c. If on a Base05 or lower PCL use

`> RE FILE` <**disk_volume**><**tape_device**><**filename_cm**>
Restores the CM load onto the SLM, for example:
RE FILE S01DIMG0 S01T LD101015ND36_CM

`> RE FILE` <**disk_volume**><**tape_device**><**filename_ms**>
Restores the MS load onto the SLM, for example:
RE FILE S01DIMG0 S01T LD101015MS36CR_MS

d. `> ET` <**tape_device**>
Ejects the SLM tape, for example:
ET S01T

e. `> QUIT`

Procedure 2
Preload both MSs

Backward-compatibility is supported in the Message Switch (MS). This means it is possible to PRELOAD *both* MSs with the new MS load before the CM is upgraded to the new PCL.

CAUTION
Do not attempt to upgrade the Message Switch at this time unless the office is currently on BCS34 or higher.
Failure to heed this caution could result in degradation of the switch since the MS load is not backward compatible until the office is on at least BCS34.

CAUTION
Both MSs must be loaded with the MS load provided on the PCL IMAGE tape prior to starting the PCL application.
The following procedure assumes the proper MS load is already copied to the SLM disk volume.

Note: The PCL IMAGE provided is patched current. If any new patches are required, these will be downloaded to SFDEV and applied on the night of the PCL application.

1 **Site/ACT** List the SLM disk volume onto which the new PCL IMAGE files (both _MS and _CM loads) were previously restored (copied).

a.

```
> DISKUT
```

```
> LF SOOD<volume>                              {or SO1D<volume>}
```

where <volume> is the SLM disk volume with the PCL IMAGE.

b. Verify the image files (both _MS and _CM) on the SLM disk volume are correct. Be sure to use the image files provided on the new PCL IMAGE tape.
To help understand the image filenames, you may use the DISPMS <filename> command which displays the image header information. (Refer to Appx. A for more details of this command.)

---

2 At the MS level of the MAP, determine which MS contains the SLAVE clock. (Look for "slave" under the CLOCK field.)

```
> MAPCI;MTC;MS
```

—continued—

Procedure 2
Preload both MSs (continued)

3

```
> BSY <MS#>
```

{the MS with the slave clock}

4

```
> LOADMS <MS#> <filename>
```

where <filename> is the name of the _MS load file listed above in step 1.

```
> YES
```

{for confirmation}

---

5 When prompted, perform an out-of-service test on the MS just loaded.

```
> TST <MS#>
```

{on the OOS MS}

Ensure the test passes with no faults. Determine the cause for any failure, fix the fault, and repeat the test.

> **CAUTION**
> **Do not proceed unless NO faults are reported.**
> Replace cards if necessary and repeat the test. Contact site supervisor if the test fails repeatedly.

6

```
> RTS <MS#>
```

{not OOBAND!}

---

7 Wait 5 minutes to ensure the clocks are stable and to allow the hardware audit to run. Both MSs should be inservice.

Note: The MS load on the CM image tape is patched as current as possible. Copies of all MS patches that were applied to this load will be in mate SFDEV when the CM image is loaded for the PCL application. Once the CM load is made active (by the SWACT) the MATCHALL MS (Patcher) or DBAUDIT (PRSM) will function normally.

---

8 Switch MS clock mastership.

```
> SWMAST
```

---

9 Monitor MS logs for 10 minutes to ensure stability.

---

10 Repeat steps 3 through 9 to update the load in the other MS.

11

```
> QUIT MAPCI
```

Site responsibilities the day of the software delivery

Site personnel must have the following steps completed *before* the software delivery engineer contacts the site to begin the scheduled ONP application.

Procedure 1
Day zero checklist

1 **Site** Verify that the front-end processors have been in sync for the past 24 hours (except for maintenance) and the last REX test passed.

---

2 Ensure an IMAGE was taken in the last 24 hours and backed-up to tape.

Note: Store a *tape copy* of the office image in a safe place for at least 30 days following the application. (This is in case the disk image is overwritten.)

---

3 Ensure you have defined and tested your *Test Call Scripts*. The test call plan must be ready following activation of the new software load.

---

4 Verify all patches required for the new PCL have been downloaded and are present on the device where they were downloaded.

---

5 Check for a **SOC file** “<clli>_SCF” or “<clli>$SCF” in SFDEV. If a SOC file is present copy the file to a disk drive (or tape). *The file will be erased from SFDEV during the ONP*.

Note: For information on installing the RTU (right-to-use) SOC password file, refer to *NTP 297-8991-901*.

---

6 Verify SFDEV is cleared of all unnecessary Telco/site-created files. (This is to clear storefile space for ONP work.)

---

7 **LIU7 image** Assuming LIU7s have had their s/w loads upgraded, dump an image of one INSV LIU7. There should be no changes to the tables C7GTT, C7GTTYPE, C7NETSSN, or C7DCIS6 between the time this image is taken and when the ONP occurs.
Utilizing feature AQ1102, this allows faster recovery of LIU7s under certain circumstances. Please refer to the feature description for details of the possible recovery scenarios. If table C7GTT is very large, for example, data sync of the LIU7 following the CC Warm SWACT will complete faster if an LIU7 image with up-to-date data is available.

Procedure 2
Patch verification

The Site is responsible for the following patch verification step.

1 **Site/ACT** Any patches downloaded after the final office review should be copied to disk (or to tape) and *left in SFDEV*.

• From-side patches that are process-related (affecting ONP tools) will be applied by the PCL Applicator on the night of the application.

• To-side front-end and MS patches in SFDEV will be automatically applied to the new PCL on the night of the application.

2 Ensure that Table PADNDEV points to the device containing the new, unapplied CM patches (normally in SFDEV). This will help in locating the patches to be applied to the new software load.

CAUTION
DO NOT erase the *DLCHECK* file from storefile.

This file (if present) will be needed to run automatically when mate-side patches are applied.

Procedure 3
Run DATADUMP

1 **Site/ACT** Run DATADUMP to output important switch information for future reference.

a. `> LOGUTIL;STOPDEV <`**`printer`**`>`
where <printer> is an available printer to be used for recording. This makes sure the logs are stopped on the device.

```
> LEAVE
```

b. `> RECORD START ONTO <`**`printer`**`>`

c. `> BCSUPDATE;DATADUMP`

When DATADUMP is completed:

```
> QUIT
```

d. `> RECORD STOP ONTO <`**`printer`**`>`

Procedure 4
FX switch access

1 **Site** Ensure there will be uninterrupted communication with the software delivery engineer during the conversion. Provide four FX data lines and at least one FX voice line.

- *Four dialup ports* are required for the SNSE/SN conversion: two for the SNSE processor and two for the SuperNode processor. On each processor one dialup must reside on IOC 0 and one dialup on IOC 1. A *reliable voice number* also must be provided to allow direct contact with the Nortel software delivery engineer.
- It is crucial that *Foreign Exchange (FX)* directory numbers are used for the data ports and voice line(s). The cutover will totally disable call processing on the DMS. **All contact will be lost unless FX voice and data lines are used**. Nortel Installation will connect the lines provided to the SuperNode IOC. The site is requested to assist installation by providing these FX lines.

CAUTION

If possible, do not use X25 terminals to perform the ONP software delivery.

X25 ports will drop during the Cutover which will delay or prevent the user from logging back in after the SWACT.

---

2 Ensure all dialup ports provided are operational. These should have COMCLASS of ALL. Use direct-connect modems which are clearly labeled.

- It is highly recommended that access to the SNSE and SuperNode be via *direct connection modem devices that require no manual intervention* to operate. Otherwise, should the software delivery engineer lose contact with the SuperNode during cutover, a serious degradation of service could occur or be prolonged.

---

3 Make note of the IOC, CARD and CKT numbers for each dialup. Verify user names to be used during the software update have PRIVCLAS of ALL.

Procedure 5
Network management control

Note: If necessary, contact your Network Maintenance personnel for assistance with these steps.

Procedure for going to Base05 (NA004) and higher

If Network Management code blocking is active prior to an ONP, the code blocking must be restored after the ONP is complete.

1 **Site/ACT** Make a full list of all active code controls prior to the ONP to aid in the restoral of code blocking.

```
> MASSCALL LIST CGAP ACODE ALL
```

This will give a full list of CODE CONTROLS which are ACTIVE.

2 The code blocking *must be restored after the ONP is complete*. Make arrangements with Network Maintenance personnel for assistance.

Installer procedures to prepare the SuperNode

Nortel Installation must complete the following procedures *before* the software delivery engineer contacts the site to begin the Dump and Restore.

Procedure 1
Verify SuperNode datafill

1 **Inst/SN** Using the D610/620 hardware extracts and the DMOPRO files provided by Nortel Data Engineering, ensure that the “engineered” data are identical with the actual hardware configuration of the SuperNode.

2 Please contact the TAC Center immediately if any discrepancies are identified with the information provided by Data Engineering.

3 Ensure the data present in the SuperNode (Active) is correct and has been verified during Link Testing.

The Dump/Restore engineer will re-use data from the following tables and will transfer them into a new undatafilled load.

CMSHELF
MSINV
MSCDINV
MSPTINV
MSILINV
ENINV
ENCDINV
IOC
SLM
LIMINV
LIMCDINV
LIMPTINV
SUSHELF
NIUINV
LIUINV
LTCINV
TMINV
LMINV
DCMINV
TERMDEV

At this point in time, the SuperNode should contain the verified datafill that was used during Installation link testing. The SuperNode should also remain InSYNC.

—continued—

Procedure 1
Verify SuperNode datafill (continued)

4 Ensure all *PM loads* required are copied from SNSE disk volume to the SuperNode SLM volumes. Also copy all required *“X” and “I” patches*. Especially verify each *ISN load* has current image dumped to SNSE disk (patched current) being on SuperNode SLM volume. Ensure *tables PMLOADS and PADNDEV* on the SuperNode reflect the required SLM disk volume datafill.

Procedure 2
Assure remote login

1 Login on both the SNSE and SuperNode switches. For each of the assigned dial-ups, verify the datafill and device states are correct to allow remote dial-in by the Dump/Restore engineer.

---

2 Verify CIOC "READ" function from MTD.

a. Mount SCRATCH tape with write ring on the SNSE MTD that will be used for the ONP data dump.

b.
```
> ERASETAPE T<x>
```

```
> MOUNT <x> FORMAT SETEST
```

```
> DUMPTAB <tablename> EXTERNAL T<x> TOTAL PUT
<filename>
```
Where tablename and filename are, respectively:
MTD....MTD$SNSE
DDU....DDU$SNSE
MPC....MPC$SNSE
TERMDEV....TERMDEV$SNSE

```
> DEMOUNT T<x>
```

c. Mount this “SETEST” tape on the CIOC MTD.

```
> MOUNT <x>
```
Verify tape does mount successfully.

```
> LIST T<x>
```

```
Print the files from the test tape in reverse order:
```

```
> PRINT TERMDEV$SNSE;MPC$SNSE;DDU$SNSE;MTD$SNSE
```
Verify each file is printed; investigate any errors encountered.

```
> DEMOUNT T<x>
```

This concludes the preparations on the SuperNode required to begin the Dump and Restore. Leave the SuperNode in this configuration until contacted by the Dump/Restore engineer.

Do not make any further changes to the SuperNode software or table data. To do so will cause a delay in the conversion and may result in an abort.

This concludes the Site Preparation phase of this Method of Procedure. Assure that the Site Preparation Checklist is completed.

Preliminary phase procedure

This section details Dump and Restore activities on the morning of the Cutover performed by Telco SITE personnel, the Nortel Installer, and the Dump/Restore engineer (or Applicator).

Four dialup ports are required for the SNSE/SN, two on the SNSE processor and two on the SuperNode processor. For each switch one dialup must be on IOC 0 and the other on IOC 1. The Site is requested to assist Installation in providing these dialups. **If possible, do *not* use X25 terminals to perform the remote procedures**.

Telco may elect to monitor the application process by recording onto printers. Start recording by issuing the following command for printers for each of the dialup ports to be used by the Applicator: "RECORD START FROM <terminal_id> ONTO <printer>." In addition, the Dump/Restore engineer should acquire a *soft copy* of console sessions for all dialups.

> **CAUTION**
> **Ensure no hardware changes, additions or retrofits are in progress. This includes network or memory extensions and peripheral additions/deletions.**
> These activities are prohibited during the Dump and Restore. Such hardware must be made INB (installation busy), and any further software changes must cease.

Note on Installer responsibilities

Nortel Installation must have completed a series of Installation Methods (IM) to continue beyond this point. The sequence for the completion of these methods is referenced in IM 78-5145 *SuperNode SE to SuperNode Upgrade Planning* method.

On the Night of Cutover, the Installer will perform IM 78-5146 *SuperNode SE to SuperNode Night of Cutover* in concert with the remaining procedures in this MOP. The Applicator will be responsible to direct all software delivery activities.

Procedure 1
Interrupt/ABORT process for SNSE/SN

Both SITE and Applicator should be familiar with each of the following steps before continuing with the procedures in this section.

If problems develop during the software delivery, resort to one of the following actions.

1 If the TABXFR process must be halted or interrupted, use the HALT option. Refer to “Interrupt TABXFR” in *Appendix A* (page A-19).

2 It may be necessary to STOP (and reschedule) the application after PRESWACT has been implemented, but before the switch of activity. Refer to “PRESWACT Abort” in *Appendix A* (page A-30).

3 If a controlled REVERT is required after the switch of activity (SWACT) refer to “Revert to SNSE procedure” (page 3-119).

Procedure 2
Initial remote login on SNSE and SN

1 **App/SNSE and SN** Contact the control center (if required) and the site on the voice phone and connect to all dialups. For each switch verify one dialup port is on IOC0 and the other is on IOC1.

2 Login the users and, if applicable, set LOGINCONTROL (for all dial-ups).

a. <*break*>

```
?LOGIN
Enter username and password              {system response}
```

```
> <username> <password>
```

or

```
> <username>
> <password>
```

b. Obtain IOC device and user information as follows.

```
> BCSUPDATE;DEVICE
> QUIT
```

```
> QUSER
```

```
> LOGINCONTROL <device> QUERY
```

c. Verify each of the following conditions exist. If not, change it.

- *User Priority* is *4*
- *User Stack Size* is at least *7000*
- *User Privilege Class* is *ALL*
- *ComClass* is *ALL*
- *OpenForceout* is *N*. If not, note original status and enter:

```
> LOGINCONTROL <device> OPENFORCEOUT FALSE
```

- *MaxIdleTime* is *Forever*. If not, note original status and enter:

```
> LOGINCONTROL <device> MAXIDLETIME FOREVER
```

d. Repeat this entire procedure for each of the assigned dial-ups.

Procedure 3
Check logs SNSE and SN

1 **App/SNSE and SN** Check system logs to verify processor stability.

```
> BCSUPDATE;LOGCHECK
> QUIT
```

Do not continue until all logs have been explained.

---

2 Perform the following:

```
> AUTODUMP OFF            {if scheduled time conflicts with image dump}

> REXTEST SUSPEND ALL     {if scheduled time conflicts with image dump}
```

If using PRSM on 04/05:

```
> AUTOAPP STOP;AUTOAPP DELAY
```

If using PRSM on 06 and higher:

```
> AUTOPROC ALL STOP;AUTOPROC ALL DELAY
```

Procedure 4
Drop sync SNODE

1 **App/ACT SN** Type:

```
> MAPCI;MTC;CM
```

---

2 **App/ACT** Ensure the CM you want to load with the new PCL is *inactive* and the corresponding MS and SLM components are used.
For example: if the new PCL image resides on SLM disk 0, then CM 0 should be the inactive side, and the MS 0 clock should be the slave clock.

a. Determine where the new PCL image resides (normally SLM disk 0).

b. If needed to align the CM with the SLM, you may switch activity of the CM using SWACT (CM level).

c. If needed to align the MS clock with the CM, you may switch MS clock mastership using SWMAST (MS level). If you do this, wait five minutes to continue.

> **CAUTION**
> **Ensure there are no opened files (such as DIRP/AMA) on the same SLM device with the image.**
> If any opened files are found, close them before proceeding. This will prevent the file system from interfering with the loadmate process.

---

3 **Site/INACT** From the inactive RTIF enter:

```
RTIF> \JAM
RTIF> YES
```
{for confirmation)

---

4 **App/ACT**

```
> DPSYNC
```
{from CM level)

```
> YES
```
{if prompted to disable AUTO PATCHING}

```
> YES
```
{to confirm DPSYNC}

---

5 **Site/INACT** Site must tell the engineer when the inactive CM is flashing A1.

---

6 **App/ACT**

```
> QUIT MAPCI
```

Procedure 5
Loadmate SNODE

1 **App/ACT SN** List the SLM disk volume onto which the new PCL IMAGE files (both _MS and _CM load files) were previously restored (copied).

a.
```
> DISKUT
```

```
> LF SOOD<volume>                                   {or SO1D<volume>}
```
where <volume> is the SLM disk volume with the PCL IMAGE.

b. Verify the image files (both _MS and _CM) on the SLM disk volume are correct. Be sure to use the image files that were provided on the new PCL IMAGE tape.
To help understand the image filenames, you may use the DISPMS <filename> command which displays the image header information. (Refer to Appx. A for details of this command.)

CAUTION

LDMATE DIRECT (below) will fail if there are any opened files on the SLM device with the image.

"DIRECT LOADMATE OPERATION FAILED: File System operations must be halted before initiating loadmate."

2.1 **App/ACT** Loadmate directly from SLM disk as follows.

a. Ensure no DIRP files are opened on the SLM disk with the image.

```
> DIRP;QUERY <subsystem> ALL
```
where <subsystem> is AMA, OM, JF, or DLOG.

If any opened files, close the files (or rotate the information to the active side).

b. Verify the correct CM load is in the Image Table of Contents (ITOC). If not, do the following.

```
> SBF <volume><filename_CM><boot_table><entry#>
```
where <volume> is the SLM disk volume with the load IMAGE.
The <entry#> is usually a number between 5 and 15 which is not being used as an entry for a bootable image file.

Example: SBF S00D LD123456_CM CM 9

c.
```
> LDMATE DIRECT DISK <filename_CM>
```
If you get the above "open files" message, either close the opened file(s), or else loadmate using the VIAMS option as follows.

```
> LDMATE VIAMS <filename_CM>
```

—continued—

Procedure 5
Loadmate SNODE (continued)

2.2 **App/ACT** Alternate step to loadmate directly from SLM tape. (Use this, for example, if the _CM image file was *not* previously copied to the SLM disk.)

Insert the new PCL IMAGE cartridge into the SLM tape drive corresponding to the inactive CM.

```
> LDMATE DIRECT TAPE 2
```
{loads 2nd file on the tape}

---

3 **Site and App/INACT** Wait for loadmate to complete and the inactive processor to flash A1.

Procedure 6
Load Modules on SNSE

1 **Site and App/ACT SNSE** Mount the TAS Non-Res tape for the SNSE and LOAD the following Conversion process modules.

The modules can also be found at IP address 47.39.48.238 in directories: /bnr/tools/sts/data/COMMON/CORE/BCS/SNSEtoSNconv/modules. Select the correct processor type (68k or 88k) and download the modules.

RETROUI
RETROCI
RETLINES
RTSDRWRS

2 **SNSE** Verify each of the modules loaded successfully.

```
> QUERY RETROUI
```

```
> QUERY RETROCI
```

```
> QUERY RETLINES
```

```
> QUERY RTSDRWRS
```

Each query should show module information if the module is present.

Procedure 7
ACTDUMP on SN

1 **App/ACT SN** On the SuperNode “ACTDUMP” each of the following tables using the DUMPONLY setup.

```
> TABXFR
```

```
> TRACECI ME
```

```
> SETUP DUMPONLY;DUMP SFDEV ONLY <tablename>
```

Repeat this entire line for each table below.

CMSHELF
MSINV
MSCDINV
MSPTINV
MSILINV
MSFWLOAD
ENINV
ENCDINV
IOC
SLM
IMAGEDEV
LIMINV
LIMCDINV
LIMPTINV
SUSHELF
NIUINV
RLOGTAB
LTCINV
TMINV
LMINV
DCMINV

The resulting “$” files (for example, “CMSHELF$”) will be used during the data transfer (that is, TABXFR) placing the data into the new undatafilled load.

---

2 DUMPTAB tables LIUINV and XSGDEF as follows.

```
> DUMPTAB LIUINV EXTERNAL SFDEV TOTAL PUT LIUINV$
```

```
> DUMPTAB XSGDEF EXTERNAL SFDEV TOTAL PUT XSGDEF$
```

Creates two “$” files to be used during the data transfer.

---

3 Matecopy to inactive SFDEV each of the files created in in steps 1 and 2 above.

```
> MATECOPY <filename$>
```

where <filename$> is each “$” file from above.

Procedure 8
Inactive side login after loadmate

1 **App/SN** On the SuperNode after loadmate is complete, starting from the active side login to the inactive (mate) side.

a. **ACT** Type:

```
> MATELINK RTS
```

The inactive side should initialize.

b. **INACT** Wait for the inactive side to initialize (flashing A1).

c. On the terminal reserved for inactive side access, LOGOUT of the active side if logged in here.

d. **ACT** On the active side, type:

```
> MATEIO
```

```
> MATELOG <device>
```

where <device> is the name of the inactive side terminal.

e. **INACT** On the inactive side, login as follows.

```
Enter username and password
```

{system response}

```
Mate> OPERATOR OPERATOR
```

or

```
Mate> OPERATOR
Mate> OPERATOR
```

Procedure 9
Fix data after Loadmate

1 **App/INACT SN** *On the inactive side of the SuperNode* make the following changes.

In table IOC change the EQPEC to ‘1X61AB’.

In table TERMDEV set MAP to ‘EIA’ ‘VT100’ and ‘B1200’.

In table MTD assure the MTD is assigned as installation has it cabled. If not the same as IOC 0 on the SNSE, match the MTD card from the Commissioning load.

You may have to busy and offline the current MTD, delete it from table MTD, and add it in the correct circuit.

2 **ACT SN** *On the active side of the SuperNode* matecopy the following Installation test files to inactive SFDEV.

```
> MATECOPY LINK_RES
```

```
> MATECOPY SNODE_CUT
```

```
> MATECOPY SHELL1$LD
```

Procedure 10
SuperNode patching

1 **App/ACT** Apply any needed patches/PRSUs to the SuperNode load at this time. Typically these would have been downloaded to the Commissioning load SFDEV.

---

2 **ACT SNSE and INACT SN** Review the patch list with Telco and decide which patches/PRSUs (if any) need to be *activated* (turned ON) or set to NA (no audit) state.

a. To activate an ACT PRSU in PRSM type:

```
Mate> ASSIGN ACTIVE Y IN PRSUSET <prsu_id>
```

This activates the prsu.

Repeat the command for each PRSU to be activated.

b. To activate an ACT patch in PATCHER type:

```
Mate> PATCHEDIT <patch_id> ON
```

This activates the patch.

Note: Passwords, if required, will be provided on the 'APF' report for any "feature patches" in the new load. Give the new password to Telco, but do *not* activate the patch unless it is ON in the old load.

Repeat the command for each patch to be activated.

c. Determine from the patch list if any patches/PRSUs should be set to "NA" (no audit) state.

To set the state to NA in PRSM type:

```
Mate> ASSIGN ACTIVE NA IN PRSUSET <prsu_id>
```

This sets the prsu to "NA" state.

To set the state to NA in Patcher type:

```
Mate> PATCHEDIT <patch> NA
```

This sets the patch to "NA" state.

Repeat the command for each patch to be set to "NA."

Procedure 11
Swap MODEM and MAP

1 **Inst and App/ACT SN** *On the Active side of the SuperNode* (Commissioning load) change the following tuples in table TERMDEV:

a. POS MAP — Make note of the current datafill for later reference.

b. POS MAP — Change to ‘EIA’, ‘VT100’, and ‘B1200’.

c. POSITION the dialup modem device for SDE, and note the current datafill for later reference.

d. Dialup Modem — Change tuple DATA to match what was set for the MAP (‘CL’, VT100’ and ‘B1200/B2400’).

2 LOGOUT of the MAP (I&C on site) and from the DIALUP MODEM being used by the SDE.

3 MANBSY the IOC 0 CARD 2 PORT 0 (MAP) and the DIALUP MODEM device PORT for the SDE—typically IOC 0 CARD 2 PORT 3.

4 Have the I&C tech on site swap the CIOC connectors for the DIALUP MODEM device with that connector for the MAP—typically “C15” (CARD 2 PORT 3) with “C12” (MAP).

5 Return-to-service *both* the MAP terminal Port (now connected as the Dialup Modem) and the DIALUP terminal Port (now connected as the MAP terminal device).

6 LOGIN—SDE to the Dialup Modem device (“MAP” device) and I&C on site to the MAP terminal.

Do NOT proceed until both devices have been verified operational.

Procedure 12
Cold SWACT SNODE

1 **App/ACT SN** Logout of the mate side.

```
> MATELINK BSY
> YES
```
{for confirmation}

---

2 **Site and App/INACT** Ensure inactive side is *unjammed*.

---

3 **App/ACT** Perform a switch of activity (cold swact) to activate the new load.

```
> MAPCI NODISP;MTC;CM;SWACT FORCE
> YES
```
{for confirmation}

---

4 **Site/ACT** Monitor the SWACT, and tell the software delivery engineer when the active processor is again flashing A1.
At this point the CM switch of activity is over.

Procedure 13
Set date and header message on SN

1 **App/ACT SN** On the active side of the SuperNode set the current date and time.

```
> SETDATE <dd mm yyyy>                          {set today's date}
```

Note: Prior to Base07 use the two digit designation for year...<**dd mm yy**>.

Also ensure the time is correct—if not, set the time.

---

2 Set the correct office header message.

```
> SETLOGMSG '<text>'
```

*where <text> becomes the office header on the new software load. Using the old header as the model, change the **Office Order** (COEO), **office name**, **Product Code** (or PCL level), and **application date**. Ensure all symbols at the beginning and end of the header message remain the same (including spaces).*

Note: The “Order/Suborder” (Office Order) and “To Product/Version” (Product Code/PCL) can be found in the *Parmmail*.

Example:

```
95/09/12 00:41 *** H01234 OFFICE LEC00005 120995 ***
```

3

```
> TRAPINFO CLEAR
```

4

```
> TABLE DSLIMIT;POS STOREFS
```

Make sure DSMAX is at least ‘458752’.

---

5 RTS SLM 0 and SLM 1 (to access disk volume files).

---

6 Upload (or MATEBIND from SLM disk) to SFDEV the following files.

a. Parms

b. SOC file

c. any Commissioning SFDEV files or specific data files required for the TABXFR restore process, such as MTD$ (with actual MTD and temporary CIOC datafill), PMLOADS$ (for disk volume name changes), and any other data files required for SFDEV changes.

Dump and Restore procedure (data transfer)

In this section the Dump/Restore engineer begins the Dump and Restore activities. In some instances by prior agreement, certain steps in the procedure can have been completed ahead of time.

It is assumed that the SNSE front-end is inservice with no faults and is in-SYNC. The dump process is performed with the SNSE remaining in-SYNC.

Manual workaround
Restoral of OMs (DMS 250/500)

On <u>DMS 250 and DMS 500 offices</u> it is necessary to do a manual workaround for OM restoral as follows.

Mount a 9-track tape on the SNSE:

```
ERASTAPE <x>
MOUNT <x> FORMAT OMRES
SEND T<x> OMRES
OMDUMP ALL COMMANDS
SEND PREVIOUS
DEMOUNT T<x>
```

Move the tape to the SN and copy the file on it (title OMRES) to SFDEV on the SuperNode. <u>During the TABXFR, table OMACC will fail to restore. At that time, complete the following</u>:

```
LISTSF ALL
READ OMRES                              {This will run for about 55 minutes}
TABLE DART;DARTEDIT;ACCESS ALL;OVE;VER OFF
POS OMACC;CHANGE DRAREA N
QUIT;QUIT
TABXFR
```

This completes the restore of Operational Measurments.

CAUTION

Servord activity and table changes done during the ONP (Dump and Restore) will not exist on the restored side.

Inform personnel that data changes on the DMS, such as Servord and table changes, should be HALTED until the ONP is completed; or else the changes will have to be repeated after ONP.

Procedure 1
Stop journal file on SNSE

1 **Site or App/SNSE** ROTATE and STOP the Journal File recording.

a. `> MAPCI;MTC;IOD;DIRP;QUERY JF ALL`
Check which JF volume is currently active.

b. `> JF STOP`

c. `> CLOSE JF ACTIVE`

d. `> QUERY JF ALL`
Verify JF is stopped

e. `> QUIT MAPCI`

Procedure 2
Dump on the SNSE

Office data changes to the SNSE software load (such as service orders or data changes) should NOT be allowed after this point because these changes will not be carried over to the SuperNode and may adversely impact the success of the application. Notify the Nortel Software Delivery Engineer prior to making ANY data changes.

1 **Site/SNSE** On the SNSE MTD up a tape (blank, certified 2400-foot) with a write-enable ring to be used for dumping office data. This will be referred to as the "Retrofit data dump tape."

App/ACT SNSE Enter:

```
> ERASTAPE <x>
```

```
> MOUNT <x> FORMAT RETD<xx>
```

where <x> is the tape drive number, and <xx> is the from_BCS/PCL number on the SNSE.
Note: *LEC04=BCS39, 05=40, 06=41, etc.*

For DMS 250 and DMS 500 offices do the following.

```
> DUMPTAB IEXCLUDE EXTERNAL T<x> TOTAL PUT IEXCLUDE$ALL
SUPBLKS
```

2

```
> TRACECI ME
```

Response:

```
This device is selected for TRACEing
```

Note: This sends DUMP output to this terminal device (ME) for monitoring. A different device could also be used using the command: TRACECI DEVICE <terminal_id>.

3

```
> TABXFR
```

```
> SETUP DUMPONLY
```

```
> DUMP T<x>
```

where <x> is the number of the tape drive with Retrofit data dump tape.

—continued—

Procedure 2
Dump on the SNSE (continued)

4 Messages for the retrofit dump will be output. Examine these for any ERROR conditions and for updates on how the dump procedure is running. The process does not stop on errors. Therefore, fix all errors after the process is completed (see below).

5 After the process is completed, enter:

```
> LIST T<x>
```

where <x> is the number of the tape drive with the Retrofit data dump tape.

6 For each table that has a dump error, fix the problem using the table editor. Then verify the fix by entering:

```
> DUMP T<x> ONLY <table_name>
```

where <x> is the number of the tape drive with the Retrofit data dump tape, and <table_name> is the name of the table in error.

7
```
> DEMOUNT T<x>
```

where <x> is the number of the tape drive with the Retrofit data dump tape.

8
```
> TABXFR
> CANCEL
> YES
```

9 Start journal file and verify started.

a. `> JF START`

b. `> MAPCI;MTC;IOD;DIRP`

c. `> QUERY JF`

Note: QUERY JF should respond with "AVAIL." If a standby device is being used, both active and standby volumes should be marked "AVAIL."

d. `> QUIT ALL`

10 Take down the tape demounted in a previous step and label it as the "Retrofit data dump tape."

11 Save hard copy of the console session for this procedure.

Procedure 3
Restore on the SuperNode

This procedure involves restoring office data from the Retrofit data dump tape that was produced from the *Dump procedure on the SNSE*. The procedure can begin upon completion of the SuperNode loading, verification of DMOPRO files, and the dump on the SNSE.

Patch CKC09 breaks the TABXFR at table AUTOMON. The patch allows for PRSM to be turned on over the ONP. This is generally not a requirement for this product, therefore the patch should be removed prior to starting the TABXFR.

1 **Site/SN** On the SuperNode MTD put up the Retrofit data dump tape (from *Dump on the SNSE*) procedure).

Verify the the tape will mount and demount properly:

```
> MOUNT <x>
> DEMOUNT T<x>
```

---

2 **App/ACT SN** From the primary terminal (BCSDIALUP), enter:

```
> TRACECI ME
```

Response on the active side:

```
This device is selected for TRACEing
```

Note: This sends the restore output initiated from this terminal to this same terminal device (ME) for monitoring. A different device could also be used using the command: TRACECI DEVICE <terminal_id>.

---

3 From the terminal (BCSDIALUP), enter:

```
> TABXFR
> SETUP ACTREST
```

Set the data transfer threshold to stop at each error (STOPIF 1 is default) with a limit of not more than 100 total errors (Limit of 20 is recommended).

```
> LIMIT 20
> STATUS
```

Verify the defined STOP points, STOPIF, and LIMIT settings.

```
> RMOUNT T<x>
```

where T<x> is the tape drive with the Retrofit data dump tape.

—continued—

Procedure 3
Restore on the SuperNode (continued)

The RMOUNT command will mount *and* list *the Retrofit data dump tape and create a directory of the files on the tape.*

The STATUS command can be used at any time while in the TABXFR increment to display information about the set-up and status of the data transfer.

---

4 For DMS 250 and DMS 500 offices

RCOPY IEXCLUDE$ALL from the data dump tape.

---

5 Set these tables not to move via TABXFR:

ENMLASST
PMNODES
XLIUMAP *(not in UCS05 load)*
AKEYTAB
PLATAB
TAPIDTAB
VCHIDTAB
FTSPCINV
PTIDTAB
LKIDTAB
INIDTAB *(not in UCS05 load)*

Set the above tables *not to move*.

```
> DARTEDIT
```
Enters dart editing utility

```
> ACCESS ALL
```
Allows edit of table DART

```
> TABLE DART
```

To set a table *not to move* do the following:

```
> POS <table_name>
> CHANGE DRAREA N
```

Position on each table and make the change on each position.

Remember to also set tables IOC and NETWORK *not to move*.

For DMS 250 and DMS 500 offices:

```
> POS IEXCLUDE$IEXCNPAS;CHA 7 N
> POS IEXCLUDE$IEXCNPAS$IEXCNXXS;CHA 7 N
```

After all table positions have been changed, type:

```
> QUIT;QUIT
```

—continued—

Procedure 3
Restore on the SuperNode (continued)

6 Create “dummy” files in SFDEV for TABXFR restore process for NNASST and, if required, for MSILINV. Using table NNASST as an example:

```
> FILE SFDEV NNASST$                                  {if already existing}
> EDIT NNASST$
> INPUT ' '                                           {one space is inserted}
> INPUT '1'
> ERASESF NNASST$
```

7 Set stop points and start the restore process on the SuperNode as follows:

```
> STOPXFR BEFORE TERMDEV
  - added BEFORE <tablename> to STOP LIST.  {system response}

> STOPXFR BEFORE TRKMEM

> STOPXFR BEFORE LIUINV

> STOPXFR AFTER IEXCLUDE   Only required for DMS250/500.

> LISTSF
> STARTXFR
```

8 When the TABXFR stops at TERMDEV, perform the following:

```
> MAPCI NODISP;MTC;IOD;IOC 0; CARD 2
```

In table TERMDEV verify ‘EIA’, ‘TERMTYPE’, ‘BAUDRATE’ for both dialup device ports expected for CIOC CARD 2 ports

Permit 2 users.

You must change the tuple to reflect the current configuration of the SN. Failure to do so will cause connection to the SN to be lost after the upcoming restart. You may also wish to alter other tuples, typically IOC 0, circuits 8 thru 11, to reflect the current configuration. Complete this step by:

LOGIN PORT 1 (I&C on-site) and 2nd dialup (PORT 2).

LOGOFF 1st dialup (MAP) and MBSY CARD 2 PORT 0.

Swap CIOC connectors.

RTS PORT 0 and PORT 3 (1st dialup) and LOGIN.

Reset TRACECI ME (required after re-login to the dialup)

```
> STARTXFR
```

Continue to monitor the data transfer on this terminal. Observe any failures indicated in the far right column.

Warning: In particular, make sure tables IOC and “INV” h/w restore properly during the data transfer. Do not continue until any failures in these tables are corrected.

—continued—

Procedure 3
Restore on the SuperNode (continued)

9 If the error threshold is reached, TABXFR will stop. Correct any error(s) indicated.

Make needed corrections or additions to the tables using normal Table Editor commands. (Since journal file is inactive, use OVEride.)

10 When needed changes or corrections have been made, start the process again as follows.

```
> STARTXFR
```

11 When TABXFR stops at table TRKMEM, check to be sure that table VIRTGRP restored successfully. Ensure it did (see below) and continue the TABXFR:

```
> STOPXFR CLEAR BEFORE TRKMEM
  - cleared BEFORE TRKMEM from STOP LIST.    {system response}
```

```
> STARTXFR
```

Otherwise, if table VIRTGRPS did not restore completely:

```
> RCOPY VIRTGRPS
> EDIT VIRTGRPS$
> DOWN '<Virtgrp_key> USES
> CHANGE 'USES <PLUS THE REST OF TUPLE>' 'SIZE 0 NIL $'
```

Repeat the above two steps for each failure.

```
> ERASESF VIRTGRPS$
> FILE SFDEV VIRTGRPS$
> TABLE VIRTGRPS
> BOT
> OVE;VER OFF
> SEND SINK
> WHILE (DELETE)(BOTTOM);SEND PREVIOUS
> COUNT;QUIT     {Make sure count is '0'}
```

```
> TABXFR;STOPXFR CLEAR BEFORE TRKMEM
> XFRONLY VIRTGRPS
> XFRONLY VFGDATA
> XFRONLY VFGENG
> ERASESF VIRTGRPS$
> STARTXFR
```

—continued—

Procedure 3
Restore on the SuperNode (continued)

When the TABXFR stops at LIUINV:

```
> TABLE OFCSTD;OVE;VER OFF;RWOK ON
> POS DUMP_RESTORE_IN_PROGRESS;CHA 2 N
> QUIT
> DMOPRO LIUINV$
> DMOPRO XSGDEF$
> TABLE DART;DARTEDIT;ACCESS ALL;OVE;VER OFF
> POS LIUINV;CHA DRAREA N
> POS XSGDEF;CHA DRAREA N
> QUIT;QUIT
> TABLE OFCSTD;OVE;VER OFF;RWOK ON
> POS DUMP_RESTORE_IN_PROGRESS;CHA 2 Y
> QUIT
> TABXFR;STOPXFR CLEAR BEFORE LIUINV
> STARTXFR
```

When the TABXFR stops at IEXCLUDE:

```
> LISTSF ALL
> DMOPRO IEXCLUDE$ALL
> STOPXFR CLEAR AFTER IEXCLUDE
> STARTXFR
```

12 At the completion of this process, the following message will be output:

```
************************************************
    - Completed D/R of office
************************************************
```

13 After the data transfer process is completed, on the mate side retrieve D/R statistics and the data transfer report as follows.

```
Mate> DRTIME PRINT
```

Note: This printout is used for statistical information only. It should be forwarded to the appropriate Northern Telecom location listed at the end of the *Site preparation overview* section.

```
Mate> REPORT
```

Note: This creates a dump and restore delta report.

—continued—

Procedure 3
Restore on the SuperNode (continued)

14
```
> RDEMOUNT
```
This demounts the data transfer tape.

```
> SETUP STANDARD
```
This Resets setup type to standard.

```
> QUIT
```

15
```
> TRAPINFO
```

Note: If a trap occurred, do not continue until the trap is explained and action taken to correct any problem.

---

16 **App/ACT SN** Put the SuperNode SLM in service as follows.

```
> MAPCI;MTC;IOD;IOC 0
> SLM 0;BSY;RTS
> SLM 1;BSY;RTS
> QUIT MAPCI
```

---

17 Manually datafill SLM volumes in table IMAGEDEV

---

18 Perform this step to put the SuperNode inSYNC.

Site/INACT SN At the INACTIVE Reset Terminal enter:

`RTIF> \RELEASE JAM` *{verify inactive side}*

App/ACT SN SYNC the SuperNode CMs.

```
> MAPCI;MTC;CM;SYNC
```

Wait for the inactive side to flash "A1" (within 5 minutes).

---

19 If patch CKC09 was removed, re-apply this patch.

Note: A copy of the patch should be on the Patch tape shipped with the NODATA image.

20
```
> TABLE OFCENG;REP REMTERMEQP N;QUIT
```

Procedure 4
Put MS cards and networks in service

Additional MS Cards were added to in MSCDINV to support the networks. With this procedure ensure that the MS cards and the networks/IOCs are put inservice.

1 **App/ACT SN** Put the MS Cards *inservice* that were added for the networks as follows.

```
> MAPCI;MTC;MS;SHELF
```

```
> BSY 0 <X>;RTS 0 <X>
```

where <X> is offline card(s) on MS Shelf 0.
Repeat for all cards added to MSCDINV (cards 23-7)

```
> BSY 1 <X>;RTS 0 <X>
```

where <X> is offline card(s) on MS Shelf 1.
Repeat for all cards added to MSCDINV (cards 23-7)

```
> QUIT MAPCI
```

---

2 Ensure that all elements of the SuperNode ENET are put *inservice*.

```
> MAPCI;MTC;NET
```

```
> SYSTEM;BSY 0 ALL;RTS 0 ALL
> SYSTEM;BSY 1 ALL;RTS 1 ALL
```

```
> MATRIX;BSY 0 ALL;RTS 0 ALL
> MATRIX;BSY 1 ALL;RTS 1 ALL
```

Also do CARDS 1–15 AND 29–34.

CAUTION
Failure to put all elements inservice can cause an extended outage when the SuperNode is activated.

Procedure 5
Load Modules on SN

1 **Site and App/ACT SN** Mount the TAS Non-Res tape for the SN and LOAD the following Conversion process modules.

The modules can also be found at IP address 47.39.48.238 in directories: /bnr/tools/sts/data/COMMON/CORE/BCS/SNSEtoSNconv/modules. Select the correct processor type (68k or 88k) and download the modules.

RETROUI
RETROCI
RETLINES
RTSDRWRS

Also load any Customer required Non-Res modules (for example, TRKQUERY AUTO, COPYTAPE, or JFPRINT).

---

2 **SN** Verify each of the modules loaded successfully.

```
> QUERY RETROUI
```

```
> QUERY RETROCI
```

```
> QUERY RETLINES
```

```
> QUERY RTSDRWRS
```

Each query should show module information if the module is present.

Ensure the Conversion modules are now present on <u>both</u> the SNSE and SuperNode.

Procedure 6
Backup SN datafilled image

1 `> DUMP DR_IMG<`**`yymmdd`**`> <`**`slm`**`> ACTIVE <`**`option`**`> TOTAL`
where <yymmdd> refers to the year, month, and day; <slm> is the SLM disk volume cleared to receive the image; and <option> refers to UPDATE for normal image load route, otherwise use RETAIN.

---

2 Insert a blank SLM cartridge in the SLM where the image was dumped.

a. `> INSERTTAPE <`**`tape`**`> WRITELABEL FROZEN`
Where <tape> refers to the SLM tape on the same SLM device where the image was dumped.

b. `> LISTFL <`**`slm`**`>`
Where <slm> refers to the SLM volume where the image was dumped.
Retain the filenames for the next step.

c. `> BACKUP FILE <`**`slm`**`> <`**`filename`**`>_MS`

`> BACKUP FILE <`**`slm`**`> <`**`filename`**`>_CM`
Where <slm> refers to the SLM volume (example: S00DIMAGE) where the image was dumped and <filename> refers to the files listed just above.

d. `> EJECTTAPE <`**`tape`**`>`

e. Remove the SLM cartridge once the tape has stopped moving, and label the SLM cartridge as ”Datafilled SN IMAGE <PCL_name>, Date: yy/mm/dd.”

Procedure 7
Data freeze and Journal File rules

1 **Site/SNSE** Inform Telco personnel that a DATA FREEZE period begins and will remain in effect until the beginning of the conversion. ONLY LIMITED DATA CHANGES WILL BE ALLOWED. See *Journal file rules* below.

---

2 **Site/ACT** Start journal file and verify started.

a.
```
> MAPCI;MTC;IOD;DIRP;QUERY JF ALL
```
Check which JF volume is currently active.

Note: QUERY should respond with “AVAIL.” If a standby device is being used, both active and standby volumes should be marked “AVAIL.”

b.
```
> CLOSE JF ACTIVE
> CLOSE JF ACTIVE
```
JF is closed twice to ensure current timestamp on active journal file. QUERY again to verify rotation.

c.
```
> JF START
```

d.
```
> JF STATUS
```
Verify JF started.

e.
```
> QUIT MAPCI
```

Site **FOLLOW THESE RULES** through the entire data freeze period (normally up to the night of the conversion). The data freeze begins once the image is completed and the journal file is started (above). Please inform control center and craftsperson personnel of the following restrictions.

1 LIMIT SERVORD ACTIVITY and TABLE CHANGES during the data freeze.

Warning: Whenever possible use SERVORD, not table control, to make data changes.

---

2 Journal file is never to be stopped, even during journal file rotations.
If the AUTOIMAGE feature is used to take regular office images, the journal file starts and stops automatically as the image is dumped. This is the only exception to the rule.

---

3 ACTIVITIES WHICH **ARE NOT** PERMITTED

- changes to “restricted tables” (including tables C7LKSET, C7RTESET, and C7LINKSET)

  Note: A list of all the RESTRICTED TABLES can be seen by listing entries in table FREEZTAB.

—continued—

Procedure 7
Data freeze and Journal File rules (continued)

- network changes, additions, and deletions (tables NETWORK and NETJUNCT)
- PM changes, additions, and deletions (all tables ending with 'INV')
- trunk group changes, additions, and deletions (tables TRKGRP and TRKSGRP)
- trunk member changes, additions, and deletions (table TRKMEM)
- table TRKNAME changes, additions, and deletions
- IBN customer group changes, additions, and deletions
- OM and EADAS changes, additions, and deletions (tables OMACCTAB, OMCLASS, OMACCGRP, OMACCFLD, OMACCKEY, OMDELTA, and OMSET)
- DRAMREC changes, additions, and deletions (that is, ASSIGN and RECORD)
- table changes, additions, and deletions from store files, and using OVERRIDE (OVE) or VERIFY OFF (VER OFF)
- use of the RENAMECLLI command
- use of the DMOPRO command
- use of the JF STOP command
- erasing journal files from disk
- use of DIRP CLEANUP command. If file space is required then cleanup only non-journal files.

---

4 ACTIVITIES WHICH **ARE** PERMITTED

- all SERVORD commands
- table changes must be made with VERIFY ON and kept on hard copy
- emergency translation changes

---

5 CLOSE and ROTATE journal files daily (whenever the number of records exceeds 1000). KEEP THE FILENAMES IN CHRONOLOGICAL ORDER IN A JOURNAL FILE LOGBOOK.

This concludes the Dump and Restore phase of the Conversion.

Application procedures (PRESWACT)

The SNSE to SuperNode conversion continues when the software delivery engineer dials in to begin the Application phase of the Conversion.

It is assumed that the SNSE front-end is inservice with no faults and is in-SYNC. This phase is performed with the SNSE remaining in-SYNC.

Procedure 1
Remote login on SNSE and SN

1 **App/SNSE and SN** Contact the control center (if required) and the site on the voice phone and connect to all dialups.

2 Login the users.

a. *<break>*

```
?LOGIN
Enter username and password              {system response}
```

```
> <username> <password>
```

or

```
> <username>
> <password>
```

Procedure 2
Check logs SNSE and SN

1 **App/SNSE and SN** Check system logs to verify processor stability.

```
> BCSUPDATE;LOGCHECK
> QUIT
```

Do not continue until all logs have been explained.

---

2 Perform the following:

```
> AUTODUMP OFF
```

{if scheduled time conflicts with image dump}

```
> REXTEST SUSPEND ALL
```

{if scheduled time conflicts with image dump}

If using PRSM on 04/05:

```
> AUTOAPP STOP;AUTOAPP DELAY
```

If using PRSM on 06 and higher:

```
> AUTOPROC ALL STOP;AUTOPROC ALL DELAY
```

Procedure 2.1
Stop journal file

1 **Site or App/ACT** ROTATE and STOP the Journal File recording.

a.
```
> MAPCI;MTC;IOD;DIRP;QUERY JF ALL
```
Check which JF volume is currently active.

b.
```
> CLOSE JF ACTIVE
```
QUERY again to verify rotation.

c.
```
> JF STOP
```
Verify stopped.

d.
```
> QUIT MAPCI
```

Procedure 2.2
Manual journal file dump

1 **Site and App** Locate the FIRST JOURNAL FILE TAPE or JF DISK VOLUME.

> **CAUTION**
> **It is very important to start with the *first* journal file volume**
> (containing the first journal files created since the data freeze).

a. **App** If journal files are on TAPE, list the JF tape as follows.

Put up the tape *without* a write enable ring.

```
> MOUNT <x>                                        {journal file tape}
> LIST T<x>                                        {retain file names}
```

b. **App** If journal files are on DISK, list the JF disk volume as follows.

```
> DSKUT;LISTVOL <JF_disk> ALL
```

where <JF_disk> refers to the disk volume(s) containing journal files. Be extra sure to list all the volumes with JF.

---

2 Locate a scratch DISK volume for the journal file dump (or else have site install a scratch tape with a write-enable ring, and MOUNT FORMAT).

3.1 If going *to* BCS36, dump the files to the volume selected above with the following command. Otherwise, skip to the next substep.

Note: In using the JFDUMP command the applicator may also utilize a file to dump multiple journal files.

```
> JFDUMP <jfin> <jfout> <volume> <from_BCS> 36
```

where <jfin> refers to the journal file name. For <jfout> use 'JFA' thru 'JFZ', then use 'JFAA' thru 'JFZZ' (except Canada) —or use 'JF$100 thru JF$999' (Canada only). <volume> refers to the recording device volume on which to dump journal file (disk, tape, or SFDEV).

As an example: 'JFDUMP U950405000090JF JFA D000SCRATCH 34 36' dumps file U950405000090JF to file JFA onto disk volumeD000SCRATCH with formatting from BCS34 to BCS36.

Retain the output filenames for *Matebind journal files* procedure (to follow).

—continued—

Procedure 2.2
Manual journal file dump (continued)

3.2 If going *to* any PCL, dump the journal files to the volume selected above with the following command.

Note: In using the JFDUMP command the applicator may also utilize a file to dump multiple journal files.

```
> JFDUMP <jfin> <jfout> <volume>
```

where <jfin> refers to the journal file name. For <jfout> use 'JFA' thru 'JFZ', then use 'JFAA' thru 'JFZZ' (except Canada) —or use 'JF$100 thru JF$999' (Canada only). <volume> refers to the recording device volume on which to dump journal file (disk, tape, or SFDEV).

As an example: 'JFDUMP U950405000090JF JFA D000SCRATCH' dumps file U950405000090JF to file JFA onto disk volume D000SCRATCH with appropriate formatting.

Retain the output filenames for *Matebind journal files* procedure (to follow).

---

4 Repeat step 3 for each journal file listed from step 1 above.

> **CAUTION**
> **Review the time stamps of the reformatted JF to confirm the entire data freeze period is accounted for.**
> —JF active during the entire data freeze with no significant interval without journal file

---

5 `> DEMOUNT T<x>` *{only if JF was recorded on tape}*

---

6 **Site** If journal file was recorded on tape, remove the previous journal file tape and replace the write enable ring. Install the next tape without a write enable ring.

> **CAUTION**
> **Install each journal file tape in the order they were created.**

---

7 **Site and App** For each journal file tape, MOUNT and LIST the tape and repeat steps 3 through 6 above.

Procedure 2.3
Matebind journal files

1 **App/ACT** Matebind the reformatted journal files.

a. **ACT** List the device used for the journal file dump from the jf dump steps (previous procedure).

b.
```
> MATEIO
```

c.
```
> MATEBIND <jffile> <jffile>
```
where <jffile> refers to all reformatted JF filenames created in the JF dump steps.

d. Repeat MATEBIND for each filename created in the JF dump steps.

e.
```
> MATEBIND DMOLIST DMOLIST
```

Note: The site is responsible to input all DMOs from the DMOLIST. (These are DMOs which were input since datafreeze was suspended.)

f. **INACT**
```
Mate> MATEIO
```

Procedure 2.4
Restore journal files

1 **App** Restore the journal files to the inactive side.

Note: In the following steps, <jffile> refers to all the reformatted JF filenames created previously in the *journal file dump* procedure.

CAUTION
Restore all journal files in the same order they were created.

ACT

```
> TRACECI ME
```

Note: This command allows you to monitor, on the terminal from which it is entered, the results of the RESTAB command (below).

INACT

```
Mate> RESTAB <jffile> <from_BCS>
```

Correct all errors which may occur.

Repeat RESTAB for each filename created in the journal file dump.

Procedure 3
PRESWACT on the SNSE

The following procedure involves creating a Retrofit delta tape to be used for comparing office data between the SNSE and SuperNode.

1 **Site/SNSE** Put up a blank tape to be used to compare (delta) the office data between the SNSE and SuperNode loads. This will be referred to as the “Retrofit delta tape.”

App/SNSE Enter:

```
> MOUNT <x> FORMAT DELT<xx>
```

where <x> is the tape drive number, and <xx> is the from_BCS/PCL number on the SNSE.

---

2 **App/SNSE** From the RETROFIT level begin PRESWACT.

```
> RETROFIT <DUMP>
```

Note: *LEC04=BCS39, 05=40, 06=41, etc.*

Response:

```
RETROFIT>>
```

```
> PRESWACT
```

Response:

```
Please enter the tape drive to be used:
```

```
> T<x>
```

```
Enter target BCS of Supernode:
```

```
> <yy>
```

Note: *LEC04=BCS39, 05=40, 06=41, etc.*

```
Tape <x> selected, CM target = yy
:
:
```

Verify that <yy> is the RESTORE_# (to_BCS) on the SuperNode.

Note: PRESWACT runs all steps required before the conversion, and flags them as complete when they pass. If any error occurs, PRESWACT will stop and give instructions. If this is the case, follow PRESWACT instructions to correct the problem (contact technical support if necessary). Run PRESWACT again (type >PRESWACT) to continue.

Example of an error condition:

```
SET_OFFICE_TUPLES            executing
```

```
ACT - Error: Node rexcontrol value is invalid.
Correct error condition.  Enter Preswact to continue.
```

—continued—

Procedure 3
PRESWACT on the SNSE (continued)

3 When PRESWACT is complete, a listing of all PRESWACT procedures is printed showing their status and execution time.

4 After the PRESWACT process is completed, enter:

```
> DEMOUNT T<x>
```

where <x> is the number of the tape drive with the Retrofit delta tape.

```
> LEAVE
```

5 **Site** Take down the tape demounted in the previous step and label it as the “Retrofit delta tape.”

Procedure 4
PRESWACT on the SuperNode

1 **App/SN** Mount and list the Retrofit delta tape from the previous procedure.

```
> MOUNT <x>;LIST T<x>
```

where <x> is the number of the tape drive with the Retrofit delta tape.

---

2 **App/SN** From the RETROFIT level begin PRESWACT.

```
> RETROFIT <RESTORE>
```

Note: *LEC04=BCS39, 05=40, 06=41, etc.*

Response:

```
RETROFIT>>
```

```
> PRESWACT
```

Response:

```
Please enter the tape drive to be used:
```

```
> T<x>
```

```
Retrieving SNSE BCS number.

Tape x selected, SNSE BCS number = xx
:
:
```

Verify that <xx> is the DUMP_# (from_BCS) on the SNSE.

Note: PRESWACT runs all steps required before the conversion, and flags them as complete when they pass. If any error occurs, PRESWACT will stop and give instructions. If this is the case, follow PRESWACT instructions to correct the problem (contact technical support if necessary). Run PRESWACT again (type >PRESWACT) to continue.

Example of an error condition:

```
DELTA_TABLES                 executing

TABLE AMAOPTS.  36 tuples reordered.

Table AMAOPTS did not match file AMAOPTS.  See file
DELTA$ERROR
DELTA_TABLES                 not complete.

ACT - Error: Correct delta error.
Correct error condition.  Enter Preswact to continue.
```

To correct an error condition, print file DELTA$ERROR to examine the differences between the old and new data. Determine which are true errors. If necessary, change the table in error (SuperNode) using normal Table Editor commands to match the data in the corresponding file (SNSE). Note that the table format may change between the old and new loads.

—continued—

Procedure 4
PRESWACT on the SuperNode (continued)

Certain PRESWACT steps will indicate an error during a SNSE to SN Conversion. It is *normal* for the following steps to fail and cause PRESWACT to stop:

READ_DTDETECT
READ_DEVICE_STATUS
OFFL_DEVICE_CHECK

If any of the above steps fail, investigate the source of a failure as always.

— For **DMS 250 offices** “DTDETECT” data will not exist and will, therefore, cause PRESWACT to stop.

— The expected status of IOC 0 and the MS Port Cards can also cause a step to fail.

These errors are normal. The Applicator should OVERRIDE the failed step and continue PRESWACT. *For example:*

```
> OVERRIDE READ_DTDETECT
> PRESWACT
```

3 At the midpoint of PRESWACT (at procedure CM_RESTART), a restart reload is required. When prompted to do so, enter “YES” to proceed with the restart.

4 Following the restart reload and after the CM has initialized (flashing “A1”), LOGIN in on the SuperNode.

```
<break>
```

```
? LOGIN
```

```
Enter user name and password
>
```
{or enter each separately}

5 Start PRESWACT running again by repeating steps 1 and 2 above.

6 When PRESWACT is complete, a listing of all PRESWACT procedures is printed showing their status and execution time.

7 After the PRESWACT process is completed, SYNC the SuperNode as follows.

```
> TRAPINFO
```

Note: If a trap occurred, do not continue until the trap is explained and action taken to correct the error.

```
> DEMOUNT T<x>
```
where <x> is the number of the tape drive with the Retrofit delta tape.

—continued—

Procedure 4
PRESWACT on the SuperNode (continued)

```
> LEAVE
```

```
> MATELINK BSY
> YES                                          {for confirmation}
```

```
> MAPCI;MTC;CM;SYNC
```

DO NOT release the JAM according to the following response:

```
The JAM should be removed prior to SYNCing.
Do you wish to continue?
Please confirm ("YES" or "NO")
```

```
> YES
```

```
> QUIT MAPCI
```

Procedure 5
Match Signaling System 7 states

1 **App/SNSE** For all CCS7 offices, the following must be performed to ensure all CCS7 components are inservice after the conversion.

Note: In the following steps 'InSv' and 'IDL' are references for all states but 'OffL'.

a. `> TABLE C7LKSET;LIST ALL;QUIT` *{retain for later}*

b. `> TABLE C7RTESET;LIST ALL;QUIT` *{retain for later}*

c. `> TABLE C7LOCSSN;LIST ALL;QUIT` *{retain for later}*

d. `> TABLE C7NETSSN;LIST ALL;QUIT` *{retain for later}*

e. *If MSB7 is not equipped, go to* ***substep k*** *below.*

`> MAPCI;MTC;PM;POST MSB7 ALL` *{retain the InSv MSB7 number for later}*

f. `> STC;POST ALL` *{retain the InSv STC of the posted MSB7 for later}*

g. `> NEXT` *{retain the InSv STC of the posted MSB7 for later}*

h. Repeat **substep g** for each STC in the posted set.

i. `> QUIT;NEXT` *{retain the InSv MSB7 number for later}*

j. Repeat **substeps e** through **i** for each MSB7 in the posted set.

k. `> CCS;CCS7;C7LKSET`

l. `> POST C <linkset in table C7LKSET>` *{retain all InSv LK numbers for later}*

m. `> NEXT` *{retain all InSv LK numbers for later}*

n. Repeat **substep m** until 'No more links to be viewed in the linkset' message.

o. Repeat **substeps l** through **n** for each LINKSET listed from table C7LKSET.

p. `> C7RTESET`

q. `> POST C <routeset in table C7RTESET>` *{retain the InSv RTE for later}*

r. Repeat **substep q** for each ROUTESET listed from table C7RTESET.

s. `> SCCPLOC;POST ALL` *{retain each InSv SubSystem and Instance-0 to 31 for later}*

t. `> NEXT` *{retain each InSv SubSystem and Instance-0 to 31 for later}*

u. Repeat **substep t** until 'End of posted set' message.

—continued—

Procedure 5
Match Signaling System 7 states (continued)

v. `> SCCPRPC`

w. `> POST <pcname in table C7NETSSN>` *{retain the InSv POINTCODE for later}*

x. `> SCCPRSS;POST ALL` *{retain each InSv SubSystem of the posted POINTCODE}*

y. `> NEXT` *{retain each InSv SubSystem of the posted POINTCODE}*

z. Repeat **substep y** until 'End of posted set' message.

aa. `> QUIT`

bb. Repeat **substeps v** through **z** for each PCNAME listed from table C7NETSSN.

---

2 **App/SN** For all CCS7 offices, apply the data obtained earlier from the SNSE.

a. `> TABLE C7LKSET;LIST ALL;QUIT` *{verify contents are the same as the SNSE}*

b. `> TABLE C7RTESET;LIST ALL;QUIT` *{verify contents are the same as the SNSE}*

c. `> TABLE C7LOCSSN;LIST ALL;QUIT` *{verify contents are the same as the SNSE}*

d. `> TABLE C7NETSSN;LIST ALL;QUIT` *{verify contents are the same as the SNSE}*

e. `> MAPCI;MTC;PM;POST MSB7 ALL` *{verify MSB7 is CBSY for each InSv SNSE MSB7}*

f. If the MSB7 is CBSY for each InSv SNSE MSB7, go to **substep i**; otherwise, continue.

g. `> BSY PM;RTS PM`

h. `> YES` *{for confirmation}*

i. `> STC`

j. `> POST <InSv SNSE STC>;BSY;RTS`

k. Repeat **substep j** for each InSv SNSE STC.

l. `> QUIT;NEXT`

m. Repeat **substeps g** through **l** for each InSv SNSE MSB7 in the posted set.

—continued—

Procedure 5
Match Signaling System 7 states (continued)

n. `> CCS;CCS7;C7LKSET`

o. `> POST C <InSv SNSE LINKSET>`

p. `> BSY <InSv SNSE LK number>`

q. `> ACT <InSv SNSE LK number>` *{disregard message 'Link #: Failed, unable to seize a transmission link'}*

r. `> RTS <InSv SNSE LK number>`

s. Repeat **substeps o** through **r** for each InSv SNSE LK in the posted set.

t. `> NEXT`

u. Repeat **substeps p** through **t** until 'No more links to be viewed in the linkset' message.

v. Repeat **substeps o** through **s** for each InSv SNSE LK.

w. `> C7RTESET`

x. `> POST C <InSv SNSE RTE>;BSY;RTS`

y. Repeat **substep x** for each InSv SNSE RTE.

z. `> SCCPLOC;POST ALL`

Note: **Substeps aa** and **bb** are shown on two lines connected by an "↵", but will be able to be input on a single command line.

aa.

```
> BSY <InSv SNSE SubSystem> ↵
<SNSE InSv Instance-0 to 31>
```

bb.

```
> RTS <InSv SNSE SubSystem> ↵
<SNSE InSv Instance-0 to 31>
```

cc. Repeat **substeps aa** and **bb** for each InSv SNSE SubSystem and Instance.

dd. `> NEXT`

ee. Repeat **substeps aa** through **dd** until 'End of posted set' message.

—continued—

Procedure 5
Match Signaling System 7 states (continued)

ff. `> SCCPRPC`

gg. `> POST <InSv SNSE POINTCODE>;BSY;RTS`

hh. `> SCCPRSS;POST ALL`

ii. `> BSY <InSv SNSE SubSystem>`

jj. `> RTS <InSv SNSE SubSystem>`

kk. Repeat **substeps ii** and **jj** for each InSv SNSE SubSystem.

ll. `> NEXT`

mm Repeat **substeps ii** through **ll** until 'End of posted set' message.

nn. `> QUIT`

oo. Repeat **substeps gg** through **mm** for each InSv SNSE POINTCODE.

Procedure 6
Match ISDN DCH facility states

1 **App/SNSE** For all ISDN offices with table DCHINV datafilled, the following must be performed to ensure all ISDN-LTC/LGC and DCH components are inservice after the conversion.

Note 1: In the following steps 'InSv' is a reference for all states but 'OffL'.

Note 2: PMI' is used as a reference for the ISDN LTC or LGC.

Capture the following information from the SNSE for later application to the SNSE

a. `> TABLE DCHINV;LIST ALL;QUIT`

From the table DCHINV listing note all LTCs/LGCs associated with D-Channel Handlers. Post each LTC/LGC and it's associated DCH's:

b. `> MAPCI;MTC;PM`

c. `> POST LTC <X>` (or `LGC <X>`)
where <X> is the LTC or LGC number from table DCHINV

d. `> DCH;POST ALL` *{retain the InSv DCH of the posted PMI for later}*

e. `> NEXT` *{retain the InSv DCH of the posted PMI for later}*

f. Repeat **substep d** for each DCH in the posted set until 'End of Post Set' message.

g. `> ISG;POST ALL` *{retain all InSv ISG ports of the posted PMI for later}*

h. `> NEXT` *{retain all InSv ISG ports of the posted PMI for later}*

i. Repeat **substep g** for each ISG in the posted set until 'End of postset' message.

j. `> QUIT`

k. Repeat **substeps d** through **i** for each LTC/LGC in the DCHINV table listing.

l. `> QUIT MAPCI`

—continued—

Procedure 6
Match ISDN DCH facility states (continued)

2 **App/SN** For all ISDN-LTCI/LGCI offices, apply the data obtained earlier from the SNSE.

a. **App/SN** `> MAPCI;MTC;PM`

b. `> POST LGCI ALL LTCI ALL`

c. If 'No PM posted' message, continue; otherwise, go to **substep e**.

d. `> NEXT`

e. If the PMI is CBSY for each InSv SNSE PMI, go to **substep h**; otherwise, continue.

f. `> BSY PM;RTS PM`

g. `> YES` *{for confirmation}*

h. `> DCH`

i. `> POST <InSv SNSE DCH>;BSY;RTS`

j. Repeat **substep i** for each InSv SNSE DCH of the posted PMI.

k. `> ISG`

l. `> POST <InSv SNSE ISG>`

m. `> BSY <InSv SNSE CHNL>`

n. Repeat **substep m** for each InSv SNSE CHNL of the posted ISG.

o. Repeat **substeps l** through **n** for each InSv SNSE ISG of the posted PMI.

p. `> QUIT;NEXT`

q. Repeat **substeps c** through **p** for each InSv SNSE PMI in the posted set.

r. `> QUIT MAPCI`

Procedure 7
Match system clock configuration

1 **App/SN** Update TABLE SYNCLK if SLAVE configuration is used.

SNSE and SN `> TABLE SYNCLK`

SNSE and SN `> LIST`

SN `> CHANGE` *{all information will be the same except 'MASTERINT'}*

Note: Get the external timing link information from the SNSE load (that is, LK0PTYP, LK0PNUM, LK0CCT, LK0REG, LK1PTYP, LK1PNUM, LK1CCT, and LK1REG).

Examples:

```
> CHANGE 2 STRAT3 SLAVE DTC 0 0 0 DTC 2 2 1
```

or

```
> CHANGE 2 STRAT2 NT3X16AA IOE 2 4 7 A 8 ↵
SLAVE DTC 0 0 0 DTC 2 0 1
```

Procedure 8
PMLOAD tape precaution

1 **Site/SN** If office is *not* equipped with DDUs, place the new PM load tape on a tape drive at load point and online.

Procedure 9: Pre-Cutover DS30 LPP Link Move

All commands in this section will be typed by the software applicator. The software applicator will request the installer to perform cabling and switchbox related tasks at various points in the procedure.

This procedure involves busying and migrating LIM <lim_no> UNIT 0 from the SNSE to the SuperNode:
where lim_no = number of the LIM UNIT 0 being busied and migrated to the SuperNode.

All LMS->MS DS30 links must be integrated into the LPP DS30 Switchboxes for both the SNSE MS and the SuperNodeMS (as per IM 78-5144) prior to performing this procedure.

The LPP must be datafilled and offlined on the SuperNode prior to performing this procedure.

Commands will be performed at both the SNSE and the SNode during this procedure. Pay careful attention as to what switch a particular command should be executed on.

Procedure 9 – Pre-Cutover LPP DS30 Link Move (LIM <lim_no> UNIT 0)

Step	Action	Observation
1	Get the Installer to ensure that the switchboxes are in local mode.	The local LEDs will be illuminated at the switchboxes.
2	Get the installer to ensure that the switchbox toggles are set to SNSE MS.	The SNSE MS LEDs will be illuminated.
3	Note some commands will be typed at the SuperNode and some commands will be typed at the SNSE during this procedure.	
-------SUPERNODE --- SuperNode Commands ---SUPERNODE-------		
4	At the SuperNode, enter the PM level and post the LIM to be tested: **>MAPCI;MTC;PM** **>POST LIM <lim_no>** where lim_no = number of the LIM having its links tested Verify that the LIM is in an OFFL state.	
5	At the SuperNode, enter the MS level and busy the MS ports connected to LIM <lim_no> UNIT 0: **>MS;SHELF 0** **>CARD <card_no_A>** **>BSY 0 PORT <port_no_i>** **>BSY 1 PORT <port_no_i>** **>CARD <card_no_B>** **>BSY 0 <port_no_ii>** **>BSY 1 <port_no_ii>** where card_no_A and card_no_B = the numbers of the MS cards that have links to the LMS and port_no_i and port_no_ii are the numbers of the ports connected to LIM <lim_no> UNIT 0. These cards and ports can be determined from LIMPTINV. These ports will be P busy prior to executing the BSY command.	

– continued –

Procedure 9 – Pre-Cutover LPP DS30 Link Move (LIM <lim_no> UNIT 0) (Cont'd)

Step	Action	Observation
---SNSE-------SNSE --- SNSE Commands---SNSE-------SNSE---		
6	At the SNSE, enter the PM level at the MAP and post the LIM that is to be tested: **>PM** **>POST LIM <lim_no>** where lim_no = is the LIM Number	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: INSV . . UNIT 1: INSV . .
7	At the SNSE, translate the LMS to MS link assignments: **>TRNSL 1** **>TRNSL 0** Ensure that the status of all equipped links are open.	Open links display as Open or Open_pla_op.

– continued –

Procedure 9 – Pre-Cutover LPP DS30 Link Move (LIM <lim_no> UNIT 0) (Cont'd)

Step	Action	Observation
---SNSE-------SNSE --- SNSE Commands---SNSE-------SNSE---		
8	At the SNSE, perform an inservice test on UNIT 0 of the LIM: **>TST UNIT 0**	"LIM x UNIT 0 Test Initiated." "LIM x UNIT 0 Test Passed."
9	At the SNSE, perform an inservice test on UNIT 1 of the LIM: **>TST UNIT 1**	"LIM x UNIT 1 Test Initiated." "LIM x UNIT 1 Test Passed."
10	At the SNSE, busy FBUS 0: **>FBUS** **>BSY FBUS 0** **>YES** **>QUIT**	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: INSV . X UNIT 1: INSV . . where X = # of FBUS taps on FBUS 0
11	At the SNSE, busy UNIT 0 of the LIM: **>BSY UNIT 0**	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: MANB 2 X UNIT 1: ISTB 2 . where X = # of FBUS taps on FBUS 0
12	At the SNSE, determine, from the output to the TRNSL 1 command, which links are LIM cross links (usually Links 2 and 6). Busy these links. **>BSY LINK 1 <link_no_A>** **>BSY LINK 1 <link_no_B>** where link_no_A and link_no_B = the link numbers of the LIM cross-links.	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: MANB 2 X UNIT 1: ISTB 2 . where X = # of FBUS taps on FBUS 0

– continued –

Procedure 9 – Pre-Cutover LPP DS30 Link Move (LIM <lim_no> UNIT 0) (Cont'd)

Step	Action	Observation
---SNSE-------SNSE --- SNSE Commands---SNSE-------SNSE---		
13	At the SNSE, enter the MS level and busy the MS ports connected to LIM <lim_no> UNIT 0: **>MS;SHELF 0** **>CARD <card_no_A>** **>BSY 0 PORT <port_no_i>** **>BSY 1 PORT <port_no_i>** **>CARD <card_no_B>** **>BSY 0 <port_no_ii>** **>BSY 1 <port_no_ii>** where card_no_A and card_no_B = the numbers of the MS cards that have links to the LMS and port_no_i and port_no_ii are the numbers of the ports connected to LIM <lim_no> UNIT 0. These cards and ports can be determined from LIMPTINV. These ports will be P busy prior to executing the BSY command.	
14	At the SNSE, translate the LMS 1 to MS link assignments: **>PM;POST LIM <lim_no>** **>TRNSL 1** Ensure that the status of all equipped links to MS 0 and MS 1 are open.	Open links display as Open or Open_pla_op.
15	At the LPP, get the installer to remove LIM Cross Links from the LIM UNIT 0 (shelf 39, rear right hand side) bulkhead as follows: Remove cable LIM 1 from connector A2. Remove cable LIM 2 from connector B2.	
16	At the LPP switchbox LIM <lim_no> UNIT 0, get the installer to move the MS 0 (upper half) switch from the SNSE MS to the SNode MS position.	LED for SNSE MS with extinguish and LED for SNode MS will illuminate.

17	At the LPP switchbox LIM <lim_no> UNIT 0, get the installer move the MS 1 (bottom half) switch from the SNSE MS to the SNode MS position.	LED for SNSE MS with extinguish and LED for SNode MS will illuminate.

– continued –

Procedure 9 – Pre-Cutover LPP DS30 Link Move (LIM <lim_no> UNIT 0) (Cont'd)

Step	Action	Observation
	-------SUPERNODE --- SuperNode Commands ---SUPERNODE-------	
18	At the SuperNode, enter the MS level and RTS the MS ports connected to LIM <lim_no> UNIT 0: **>MS;SHELF 0** **>CARD <card_no_A>** **>RTS 0 PORT <port_no_i>** **>RTS 1 PORT <port_no_i>** **>CARD <card_no_B>** **>RTS 0 <port_no_ii>** **>RTS 1 <port_no_ii>** where card_no_A and card_no_B = the numbers of the MS cards that have links to the LMS and port_no_i and port_no_ii are the numbers of the ports connected to LIM <lim_no> UNIT 0. These cards and ports can be determined from LIMPTINV	
19	At the SuperNode, post the LIM at the PM level: **>PM** **>POST LIM <lim_no>** where lim_no = is the LIM Number	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: OFFL 6 X UNIT 1: OFFL 6 X
20	At the SuperNode, busy the LIM: **>BSY PM**	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: MANB 2 X UNIT 1: MANB 2 X
21	At the SuperNode, busy FBUS 0: **>FBUS** **>BSY FBUS 0** **>YES** **>QUIT**	
22	At the SuperNode, load LIM UNIT 0: **>LOADPM UNIT 0**	"LIM x UNIT 0 Load Initiated." "LIM x UNIT 0 Load Passed."

Procedure 9 – Pre-Cutover LPP DS30 Link Move (LIM <lim_no> UNIT 0) (Cont'd)

Step	Action	Observation
	-------SUPERNODE --- SuperNode Commands ---SUPERNODE-------	
23	At the SuperNode, test LIM UNIT 0: **>TST UNIT 0**	"LIM x UNIT 0 Test Initiated." "LIM x UNIT 0 Test Passed."
24	At the SuperNode, return LIM UNIT 0 to service: **>RTS UNIT 0**	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: ISTB 2 X UNIT 1: MANB 2 X
25	At the SuperNode, perform an inservice test on UNIT 0 of the LIM: **>TST UNIT 0**	"LIM x UNIT 0 Test Initiated." "LIM x UNIT 0 Test Passed."
26	At the SuperNode, busy LIM UNIT 0: **>BSY UNIT 0** **>YES**	"LIM x UNIT 0 Busy Initiated." "LIM x UNIT 0 Busy Passed."
27	At the SuperNode, offline the LIM: **>OFFL**	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: OFFL 6 X UNIT 1: OFFL 6 X
28	Repeat steps 1 to 27 for all remaining LIMs to be cutover.	

Procedure 10
Switch IOCs to SuperNode

1 **Site/SNSE** Remove parallel tape devices.

- **a.** `> MAPCI;MTC;IOD;DIRP`
- **b.** `> CLOSE <`**`subsystem`**`> PARALEL`
- **c.** Repeat **substep b** for each parallel tape device.
- **d.** `> QUIT MAPCI`
- **e.** `> TABLE DIRPSSYS`
- **f.** `> POS <`**`subsystem`**`>`
- **g.** `> CHANGE PARVOL $`
- **h.** `> Y` *{for confirmation}*
- **i.** Repeat **substeps f** through **h** for each parallel tape device.
- **j.** `> QUIT`

2 **Site and Inst/SNSE** Close all files going to disk (DDU) that were not started by DIRP. This includes LOGUTIL, EDIT, SEND, COPY, etc. Be sure to get a hard copy for use later.

- **a.** `> MAPCI NODISP;MTC;IOD;LISTDEV DDU`
- **b.** `> IOC <`**a**`>` *{IOC number from “LISTDEV”}*
- **c.** `> CARD <`**b**`>` *{CARD number from “LISTDEV”}*
- **d.** `> STATUS`
- **e.** `> ALLOC` *{only if STATUS shows “ready, online”}*

  Note: All non-DIRP files must have a ZERO in the FILES_OPEN field.
- **f.** Repeat **substeps b** through **e** for each equipped DDU.
- **g.** `> QUIT MAPCI`

3 **Site/SNSE** For each billing subsystem in DIRP, assign in table DIRPSSYS standby volumes to IOC 0 (if not already assigned) to allow a transfer.

—continued—

Procedure 10
Switch IOCs to SuperNode (continued)

4 **Site/SNSE** For each active volume on IOC 1 and higher, rotate to IOC 0. To accomplish this, perform either **substep 4a** or **4b** below depending on the DIRP subsystem configuration.

Note: In the following substeps <subsystem> refers to the DIRP subsystem name identified in table DIRPSSYS (such as AMA SMDR OCC CDR).

a. If DIRP subsystem is on disk and IOC 1 and higher, rotate the disk files and, if required by Telco, copy the billing files.

`> MAPCI;MTC;IOD;DIRP`

`> ROTATE <`**`subsystem`**`>`

`> YES` *{for confirmation}*

Repeat steps for each subsystem on a disk drive on IOC 1 and higher.

Copy any unprocessed AMA or other billing files on disk to a magnetic tape.

Put up a tape on MTD 0. Go to DSKUT and do LISTVOL on the volume with the files you want to copy. Either enter

`> DIRPAUTO <`**`subsystem`**`>` *{steps you through the process}*

or MOUNT the tape and enter

`> DIRPCOPY <`**`subsystem`**`> <`**`filename`**`> T0`
where <filename> refers to the unprocessed DIRP files. These start with the letter “U” (for example, U890327133614AMA).

Repeat DIRPAUTO or DIRPCOPY for each billing subsystem on IOC 1 and higher.

If applicable, verify that table DIRPHOLD is empty.

b. **Site/SNSE** If DIRP system is on tape (MTD) or BMC and on IOC 1 and higher, close the active file and , if BMC, poll it.

`> MAPCI;MTC;IOD;DIRP`

`> ROTATE <`**`subsystem`**`>`

`> YES` *{for confirmation}*

If BMC, call downstream processing to poll the billing information.

—continued—

Procedure 10
Switch IOCs to SuperNode (continued)

DPP instructions

5 **Site/SNSE** If DIRP system is on DPP and IOC 1 and higher collecting AMA, retrieve the DPP settings and close the active file.

Note 1: Be sure you get a hard copy of the polling information.

Note 2: The DPP loadname is found in the *Peripheral Software Release Document*.

```
> MAPCI;MTC;IOD;DPP AMA
```

`> COLLPSW` *{display the COLLECTOR passwords}*

`> AMATPSW` *{display AMAT passwords}*

`> AMAHRS` *{display hour boundaries for receiving data blocks}*

`> VALPARM INVALID` *{display the invalid block threshold}*

`> LSTDIR ALL` *{display the version, record this for later}*

`> ERRMAP ACT` *{display alarm assignments}*

`> VS ACT`
Verify the loadname is correct for the to_PCL load and firmware is at least DOS 21.08.

`> VS STDBY`
Verify the loadname is correct for the to_PCL load and firmware is at least DOS 21.08.

Warning: Do not continue if the DPP load name or firmware version is incorrect.

Close the active DPP file if on IOC 1 and higher.

```
> MAPCI;MTC;IOD;DIRP
```

`> ROTATE <`**`subsystem`**`>`

`> YES` *{for confirmation}*

```
> QUI;DPP AMA
```

```
> IDXMAINT CREATE FILE AMA
```

`> YES` *{for confirmation}*

—continued—

Procedure 10
Switch IOCs to SuperNode (continued)

6 Move IOC 1 and higher to SuperNode.

a. **Site/SNSE** DMNT all DIRP subsystems that are currently assigned to IOC 1 and higher.

```
> MAPCI;MTC;IOD;DIRP
```

`> QUERY <`**`subsystem`**`>` *{Retain for steps 8 and 9 below, subsystem names are listed in table DIRPSSYS}*

`> DMNT <`**`subsystem`**`>`

Repeat DMNT for all subsystems on IOC 1 and higher.

b. **Site and App/SNSE** Stop all log devices on IOC 1 and higher.

```
> LOGUTIL;LISTDEVS
```

`> STOPDEV <`**`device`**`>`

Repeat for each log device on IOC 1 and higher.

```
> QUIT
```

c. **Site and App/SNSE** BSY and OFFL all devices on IOC 1 and higher. On IOC 1 and higher, BSY and OFFL the IOC.

```
> MAPCI;MTC;IOD
```

```
> IOC 1
```

`> CARD <`**`n`**`>`
where <n> is the IOC card to be offlined.

`> BSY <`**`port`**`>;OFFL <`**`port`**`>`
where <port> is each assigned port on the card

Repeat for each device on the designated CARD.

```
> QUIT;BSY IOC;OFFL IOC
```

Repeat this substep for each IOC higher than IOC 1.

```
> QUIT MAPCI
```

d. **Inst/SNSE** Switch IOC 1 and higher by throwing the switch (backplane of IOC) from CMC to MS. Alternatively, remove the connectors “C00” and “C01”and replace them with the SuperNode cables.

Note: Cable C00 must connect to MS1 bulk-head and cable C01 must connect to MS0 bulk-head.

—continued—

Procedure 10
Switch IOCs to SuperNode (continued)

7 **App/SN** Return IOC 1 to service

```
> MAPCI;MTC;IOD;IOC 1;BSY IOC;RTS IOC
```

Following the return to service, LOGIN in on the SuperNode on a dialup on IOC 1.

```
<break>

? LOGIN

Enter user name and password   {or enter each separately}
>
```

8 **App/SN** On the SuperNode, verify the cables to IOC 1 and higher were properly connected.

```
> MAPCI;MTC;MS
```

Identify the MS with the SLAVE clock.

```
> BSY <slave_MS> CARD <x>
```

where <x> is the card number with IOC 1 assigned.

Verify that IOC 1 remains in-service. If not, reverse the cables.

```
> RTS <slave_MS> CARD <x>
```

9 **App/SN** Recover each MS card as follows.

```
> MAPCI;MTC;MS;SHELF 0

> BSY 0 <CARD#>
> RTS 0 <CARD#>
```

Repeat BSY/RTS 0 for each card assigned on MS 0.

```
> BSY 1 <CARD#>
> RTS 1 <CARD#>
```

Repeat BSY/RTS 1 for each card assigned on MS 1.

```
> QUIT MAPCI
```

Procedure 11 Mount DIRP to disk devices

1 **Site/SN** For DIRP subsystems on disk, re-assign DIRP subsystems on IOC 1 and higher.

```
> MAPCI;MTC;IOD;DIRP
> MNT <subsystem> <disk>
```

Repeat MNT for all DIRP subsystems on disk.

---

2 **Site/SN** For DIRP subsystems on tape, DPP or BMC, DO NOT re-assign DIRP subsystems on IOC 1 and higher unless table DIRPPOOL is datafilled with a DEVTYPE of DPP for the subsystem. If this is the case, proceed.

```
> MOUNT <x> FORMAT <volume_name>
> DEMOUNT T<x>
> MAPCI;MTC;IOD;DIRP
> MNT <subsystem> T<x>
```

Repeat this step for all DIRP subsystems that meet requirements.

---

3 **App/SN** Verify file DIRP_INAC is correct for DIRP billing subsystems after the conversion to the SuperNode.

```
> PRINT DIRP_INAC
```

If needed EDIT file DIRP_INAC to make corrections. The DEVTYPE field of DIRP_INAC may need to be altered to reflect the use of a DPP or BMC device.

---

4 **App/SNSE** Verify that on IOC 1 and higher, all devices are in-service, and logs are stopped. (Devices on other IOCs need to be OFFL.)

Conversion to the SuperNode (SWACT and POSTSWACT)

In the *Convert to SuperNode and Post-conversion* phase the CPUs are switched to make the SuperNode the controlling CPU. When the POSTSWACT step is started, it allocates billing devices (AMA) and performs miscellaneous hardware verifications. At that time, the SITE begins the prearranged call verification testing while the software delivery engineer performs additional recovery and verification steps.

Procedure warnings

1 Verify all essential services or high profile customers (police and emergency bureaus, hospitals, and radio stations) are not in emergency call processing mode by contacting the appropriate customer.

2 In *Procedure 1: Convert to SuperNode* (below) ensure the activity switch is not performed during periodic testing. Work quickly to minimize downtime once **step 2** is performed—Call processing ceases until that step is complete.

3 No activity is to be performed on the SuperNode CM, MS, or CLOCK until it is cleared by the software delivery engineer.
Failure to comply may result in a system restart.

4 Confirm FX voice line numbers with Site and establish voice and data contact on same if this is not already the case.

5 All personnel must read all of *Procedure 1* before proceeding.

6 Should the SuperNode develop hardware problems after conversion, the following options should be considered:

A If the fault can be cleared by system testing, do so.

B If the fault cannot be cleared by system testing, it is imperative that **absolutely no hardware in the NT9X01 frame is to be replaced until an image of the SuperNode is taken.**

C If the customer representative concludes that there is not enough time in the maintenance window to take an image prior to replacing the hardware, there are two options remaining.

At the discretion of the customer representative:

1 The fault may remain until their next maintenance window, or

2 The office shall revert to the SNSE processor.

The Customer Representative is encouraged to discuss all options with Nortel Emergency Recovery and Software Delivery to satisfactorily resolve the issue. This warning is critical because no bootable image exists on the

SuperNode at this point. The image taken by the Nortel Dump/Restore Engineer has all networks and peripherals set to an offline condition. An extended outage will occur should the SuperNode have to boot from that image.

Procedure 1
Convert to SuperNode

1 ***This is where critical activity starts.***
Make some final preparations before switching activity to the SuperNode.

a. **App/SNSE** Post the CMC level of the MAP to observe and verify that all links go “S” (SYSB) once the master switches are thrown below (this may take some time depending on the office size).

b. **Inst/MSBX** Verify the master switchboxes are powered up, the “CMC” LED is “ON”, and the “DISC” LED is “OFF”.

c. **Inst/RSBX** Verify the “LOC” LED is “OFF” at all the remote switchboxes and confirm information to the APP.

Note: If any “LOC” LEDs are “ON”, the remote switchbox may be defective and must be replaced.

d. **Inst/RSBX** Once all the “LOC” LEDs are confirmed “OFF” at all remote switchboxes, the Installer throws the switch from the “CMC” position to the “MS” position.

---

2 Switch of activity from the SNSE to the SuperNode

a. **App/SN** ***<u>SuperNode LPP</u> steps just prior to Cutover***

Repeat the following for each LPP in the office.

On the SuperNode:

```
> PM;POST LIM <lim_no>
```

```
> BSY PM
```

```
> FBUS;BSY FBUS 0;QUIT
```

```
> RTS UNIT 0
```

b. **App/SN** Perform a Reload Restart on the SuperNode.

<u>hint</u>: For this step use this verbal countdown:
• Type RESTART RELOAD ACTIVE
*• on **3** — hit <cr> and type YES*
*• on **2** — hit <cr>*
*• on **1** — Installer throws MASTER switches*

```
> RESTART RELOAD
```

Warning: As soon as the restart is confirmed (by entering “YES” (below), IMMEDIATELY GO TO **substep c**.

`> YES` *{Do not press “enter” until the installer is ready}*

—continued—

Procedure 1
Convert to SuperNode (continued)

c. **Inst and App/MSBX** The Applicator instructs the Installer to throw the master switches from the “CMC” position to the “MS” position**.**

d. **Inst and App/MSBX** The Installer confirms to the Applicator that the “CMC” LED is “OFF” and the “MS” LED is “ON” for all the master switchboxes.

e. **App/SNSE** <u>SNSE LPP</u> steps immediately after Cutover

Repeat the following for each LPP in the office.

On the SNSE:

```
> PM;POST LIM <lim_no>
```

```
> FBUS;BSY FBUS 1 FORCE
> Y
> QUIT
```

```
> BSY UNIT 1 FORCE
> Y
```

```
> OFFL
```

f. **Inst/SN** Verify the SuperNode is flashing “A1” (within 5 minutes).

Warning: Do not attempt manual system recovery at this point. Wait until the SRC (System Resource Controller) has had a chance to work (5 minutes) before beginning any system maintenance.

Procedure 2
LOGIN & Begin POSTSWACT

1 **Site and App/SN** Login on the SuperNode, confirm the date and time, and initiate POSTSWACT.

Warning: Work quickly to complete this step.

```
<break>
```

```
? LOGIN
```

```
Enter user name and password   {or enter each separately}
>
```

```
> DATE
```

Warning: If the date and time are not correct, alert the update engineer immediately so the date and time can be corrected before DIRP subsystems are allocated. If the date is incorrect, change it using: SETDATE <dd mm yy>. If the time is incorrect, change it using: SETTIME <hour> <minute>.

---

2 **App** `> BCSUPDATE`

---

3 **App** `> POSTSWACT`

Note 1: POSTSWACT runs all steps required after the restart, and flags them as complete when they pass. If any error occurs, POSTSWACT will stop and give instructions. If this is the case, follow POSTSWACT instructions to correct the problem, and run POSTSWACT again (type >POSTSWACT) to continue.

Note 2: If no problems are encountered, POSTSWACT stops after BEGIN_TESTING and waits until the site accepts the new software load. At that point, do not start POSTSWACT again until the processors are in sync. (See *SYNC clock and CPU* below.)

---

4 **Site and App** POSTSWACT recovers primary billing subsystems (such as AMA SMDR OCC CDR). Confirm that all affected DIRP subsystems were properly activated.

```
> MAPCI;MTC;IOD;DIRP
```

```
> QUERY AMA ALL
```

If applicable, allocate DIRP subsystems for standby AMA.

```
> QUIT MAPCI
```

—continued—

Procedure 2
LOGIN & Begin POSTSWACT (continued)

5 **App On the SuperNode:**

```
> PM;POST LIM <lim_no>

> BSY UNIT 1 FORCE                    {from a SYSB state}
> Y

> FBUS;BSY FBUS 1 FORCE               {from a SYSB state}
> Y

> QUIT
```

6 **Site** Perform all call tests identified in *Appendix C: Post conversion test scripts*. The times in which each test is to be performed is outlined in the Organization heading.

Note: Verify AMAB logs in conjunction with each test call made.

Warning: If an abort becomes necessary due to critical test failures, go back to the old load using *Revert to SNSE procedure* (below)

Procedure 3
Collect DPP settings

1 **App** If DIRP subsystem is DPP, perform this step while the site performs their tests. Display the current DPP settings to hard copy, and change if different from the data obtained during procedure *Switch IOCs to SuperNode* (above).

a.
```
> MAPCI NODISP;MTC;IOD;DPP AMA
```

b.
```
> COLLPSW
```
*If different, perform **substeps c** and **d**; otherwise, go to **substep e**.*

c.
```
> COLLPSW 1 <4_digits> <6_digits>
```

d.
```
> COLLPSW 2 <4_digits> <6_digits>
```

e.
```
> AMATPSW
```
*If different, perform **substep f**; otherwise, go to **substep g**.*

f.
```
> AMATPSW <4_digits> <6_digits>
```

g.
```
> AMAHRS
```
*If different, perform **substep h**; otherwise, go to **substep i**.*

h.
```
> AMAHRS <start_hour> <end_hour>
```

i.
```
> VALPARM INVALID
```
*If different, perform **substep j**; otherwise, go to **substep k**.*

j.
```
> VALPARM INVALID <threshold>
```

k.
```
> ERRMAP ACT
```
*If different, perform **substeps l** and **m**.*

l.
```
> ERRMAP <alarm_no> <type> <level>
```
Repeat for each alarm that is different.

n.
```
> VS STDBY
```
Verify the loadname is correct for the to_PCL load and firmware is at least DOS 21.08.

Warning: Do not continue if the DPP load name or firmware version is incorrect.

Procedure 4
SYNC clock and CPU

1 **App** Sync the CMs and CLOCK, and load the MS corresponding to the inactive CPU.

```
> MAPCI;MTC;MS;CLOCK;SYNC
> CM;SYNC
> QUIT MAPCI
```

Procedure 5
Site accepts load

1 **Site and App** BCSUPDATE stops (after BEGIN_TESTING) and waits until the Telco/Carrier accepts the new software load. Continue only after the software delivery engineer and site personnel agree that all call tests have passed satisfactorily.

```
> POSTSWACT
```

Note: At this point the BCSUPDATE will run all required POSTSWACT steps and flag them as complete when they pass. If any failures occur, follow BCSUPDATE instructions and enter POSTSWACT once the problem is corrected.

Procedure 6
Demount DIRP on IOC 0

1 **Site/SNSE** Close, dirpcopy, and demount all DIRP subsystems on IOC 0. It is important to follow accepted operating company practices while demounting DIRP subsystems. The operating company may require that some subsystems be copied prior to demounting.

```
> MAPCI;MTC;IOD;DIRP
```

```
> CLOSE <subsystem> ACTIVE
```

Repeat for DIRP subsystems.

---

2 Put up a tape on MTD 0. Go to DSKUT and do LISTVOL on the volume with the unprocessed files you want to copy.

Enter either

`> DIRPAUTO <subsystem>` *{steps you through the process}*

or MOUNT the tape and enter

```
> DIRPCOPY <subsystem> <filename> T0
```

where <filename> refers to the unprocessed DIRP files. These start with the letter "U" (for example, U890327133614AMA).

Repeat for all unprocessed files.

---

3 `> DMNT <subsystem>` *{retain for later}*

Repeat for all DIRP subsystems.

Procedure 7: Post-Cutover DS30 LPP Link Move

This procedure provides instructions to cutover remaining LIM (lim_no) UNIT 1 to the now in-service SuperNode:
where lim_no = number of the LIM UNIT 1 being migrated to the SuperNode.

Commands will be performed at both the SNSE and the SNode during this procedure. Pay careful attention as to what switch a particular command should be executed on.

Procedure 7 – Post-Cutover DS30 LPP Link Move (LIM <lim_no> UNIT 1)

Step	Action	Observation
1	Get the installer to ensure that the switchboxes are in local mode.	The local LEDs will be illuminated at the switchboxes.
2	Get the installer to ensure that the switchbox toggles on the LIM <lim_no> UNIT 1 Switchbox are set to SNSE MS.	The SNSE MS LEDs will be illuminated on the LIM <lim_no> UNIT 1 switchbox.
	Ensure that the switchbox toggles on the LMS 0 Switchbox are set to SNode MS.	The SNode MS LEDs will be illuminated on the LMS 0 switchbox.
3	Note some commands will be typed at the SuperNode and some commands will be typed at the SNSE during this procedure.	
	-------SUPERNODE --- SuperNode Commands ---SUPERNODE-------	
4	At the SuperNode, enter the PM level and post the LIM to be tested: **>MAPCI;MTC;PM** **>POST LIM <lim_no>** where lim_no = number of the LIM having its links tested Verify LIM <lim_no> UNIT 0 is ISTB and UNIT 1 is either MANB (RU) or SYSB.	The following will be displayed at the MAP: Links_OOS Taps_OOS UNIT 0: ISTB 4 y UNIT 1: xxxx 4 y where xxxx = MANB (RU) or SYSB

5	At the SuperNode, enter the MS level and busy the MS ports connected to LIM <lim_no> UNIT 1: **>MS;SHELF 0** **>CARD <card_no_A>** **>BSY 0 PORT <port_no_i>** **>BSY 1 PORT <port_no_i>** **>CARD <card_no_B>** **>BSY 0 <port_no_ii>** **>BSY 1 <port_no_ii>** where card_no_A and card_no_B = the numbers of the MS cards that have links to the LMS and port_no_i and port_no_ii are the numbers of the ports connected to LMS 0. These cards and ports can be determined from LIMPTINV. These ports will be P busy prior to executing the BSY command.	At the MAP alarm banner, "Man B" will appear under MS and "MC Tbl" will appear under CM.
---SNSE-------SNSE --- SNSE Commands---SNSE-------SNSE---		
6	At the SNSE, enter the MS level and busy the MS ports connected to LIM <lim_no> UNIT 1: **>MS;SHELF 0** **>CARD <card_no_A>** **>BSY 0 PORT <port_no_i>** **>BSY 1 PORT <port_no_i>** **>CARD <card_no_B>** **>BSY 0 <port_no_ii>** **>BSY 1 <port_no_ii>** where card_no_A and card_no_B = the numbers of the MS cards that have links to the LMS and port_no_i and port_no_ii are the numbers of the ports connected to LMS 0. These cards and ports can be determined from LIMPTINV. These ports will be P busy prior to executing the BSY command.	
7	At the LPP switchbox LIM <lim_no> UNIT 1, get the installer to move the MS 0 (upper half) switch from the SNSE MS to the SNode MS position.	LED for SNSE MS with extinguish and LED for SNode MS will illuminate.

8 At the LPP, get the installer to replace LIM Cross Links at the LIM UNIT 0 (shelf 39, rear right hand side) bulkhead as follows:

Connect cable LIM 1 to connector A2.

Connect cable LIM 2 to connector B2.

9 At the LPP switchbox LIM <lim_no> UNIT 1, get the installer to move the MS 1 (bottom half) switch from the SNSE MS to the SNode MS position.

LED for SNSE MS with extinguish and LED for SNode MS will illuminate.

-------SUPERNODE --- SuperNode Commands ---SUPERNODE-------

10 At the SuperNode, enter the MS level and RTS the MS ports connected to LIM <lim_no> UNIT 1:

```
>MS;SHELF 0
>CARD <card_no_A>
>RTS 0 PORT <port_no_i>
>RTS 1 PORT <port_no_i>

>CARD <card_no_B>
>RTS 0 <port_no_ii>
>RTS 1 <port_no_ii>
```

where card_no_A and card_no_B = the numbers of the MS cards that have links to the LMS and port_no_i and port_no_ii are the numbers of the ports connected to LIM <lim_no> UNIT 1. These cards and ports can be determined from LIMPTINV

11 At the SuperNode, post the LIM at the PM level:

```
>PM
>POST LIM <lim_no>
```

where lim_no = is the LIM Number

The following will be displayed at the MAP:

```
               Links_OOS   Taps_OOS
UNIT 0: ISTB       2           .
UNIT 1: MANB       2           X
```

12 At the SuperNode, busy the LIM UNIT 1:

```
>BSY UNIT 1
```

The following will be displayed at the MAP:

```
               Links_OOS   Taps_OOS
UNIT 0: ISTB       2           .
UNIT 1: MANB       2           X
```

13 At the SuperNode, busy FBUS 1:

```
>FBUS
>BSY FBUS 1
>YES
>QUIT
```

<table>
<tr><td>14</td><td>At the SuperNode, load LIM UNIT 1:
<b>&gt;LOADPM UNIT 1</b></td><td>"LIM x UNIT 1 Load Initiated."
"LIM x UNIT 1 Load Passed."</td></tr>
<tr><td>15</td><td>At the SuperNode, test LIM UNIT 1:
<b>&gt;TST UNIT 1</b></td><td>"LIM x UNIT 1 Test Initiated."
"LIM x UNIT 1 Test Passed."</td></tr>
<tr><td colspan="3"><b>-------SUPERNODE --- SuperNode Commands ---SUPERNODE-------</b></td></tr>
<tr><td>16</td><td>At the SuperNode, return LIM UNIT 1 to service:
<b>&gt;RTS UNIT 1</b></td><td>The following will be displayed at the MAP:
                    Links_OOS   Taps_OOS
UNIT 0: INSV   .   .
UNIT 1: ISTB   .   .</td></tr>
<tr><td>17</td><td>At the SuperNode, perform an inservice test on UNIT 1 of the LIM:
<b>&gt;TST UNIT 1</b></td><td>"LIM x UNIT 1 Test Initiated."
"LIM x UNIT 1 Test Passed."</td></tr>
<tr><td>18</td><td>Repeat steps 1 to 17 for all remaining LIMs to be cutover.</td><td></td></tr>
</table>

– End –

Procedure 8
Switch CIOC with IOC 0 & mount DIRP

1 **Site and App/SNSE** Except for CARD 1 (MTD 0) and CARD 2 (terminal controllers), BSY and OFFL any unneeded devices on IOC 0.

Inst Switch the CIOC with IOC 0.

Site and App RTS any device not INSV; then mount subsystems.

2 **Site** If DIRP subsystem is on disk.

```
> MAPCI;MTC;IOD;DIRP
```

`> MNT <`**`subsystem`**`> <`**`disk`**`>` *{subsystems from previous procedure}*

Repeat MNT for all DIRP subsystems on disk.

3 **Site** If DIRP subsystem is on tape or BMC.

`> MOUNT <`**`x`**`> FORMAT <`**`volume_name`**`>`

`> DEMOUNT T<`**`x`**`>`

```
> MAPCI;MTC;IOD;DIRP
```

`> MNT <subsystem> T<`**`x`**`>`

Repeat MNT for all DIRP subsystems.

4 **Site and App** Close all DIRP standby subsystems to clear the “**I**” alarm under the IOD banner of the MAP.

`> CLOSE <`**`subsystem`**`> STDBY1`

Repeat for all DIRP subsystems.

```
> QUIT MAPCI
```

Procedure 9
Connect E2A telemetry cables

1 **Inst** Connect E2A to SuperNode (reference *IM925, section 5935*) by disconnecting the XS cable from the XT cable and connecting it to the XU cable.

2 **Inst** Transfer interbay alarm cables from the SNSE to SuperNode.

Procedure 10
Check MPC loads

1 **App** Offices equipped with MPC boards and feature NTX273AA.

a. List the device which contains the MPC peripheral load.

b. Access table MPC and replace all tuples in order to bind the load name to the FID.

```
> TABLE MPC;OVE
```

c. Replace all tuples in table MPC in order to bind the load name to the FID.

```
> CHANGE
```

```
> <cr>
```

Repeat the “<cr>“ until “Tuple to be changed” message.

`> Y` *{for confirmation}*

```
> DOWN
```

Repeat this substep (**c**) until “Bottom” message.

Procedure 11
AMA verification

1 **Site**

a. Mount the AMA tape that was demounted above as a regular tape, and use the AMADUMP to verify the AMA entries on site (reference *NTP 297-1001-119N*). For contingency action, use ADUMP from the TAS NON-RES tape.

b. For CDR verification, use command,

```
> CDRDUMP <filename> <format>
```

Format is HEX, EBCDIC, or ASCII.

Procedure 12
Mount parallel DIRP subsystems

1 **Site** Bring up parallel devices (as required) using the preformatted volumes.

a. In table DIRPSSYS position on a DIRP subsystem requiring a parallel volume. Note the PARLPOOL name for the DIRP subsystem selected.
Example:

```
TABLE DIRPSSYS;POS AMA
```

b. In table DIRPPOOL position on the parallel pool number associated with the PARLPOOL from table DIRPSSYS. Then activate the parallel volume by datafilling the volume name.
Examples:

`TABLE DIRPPOOL;POS 62` *(pool for AMAPOOL)*
`CHA VOLUME23 T4`
or `CHA VOLUME23 D010PAMA`

c. Repeat **substeps a** and **b** for each parallel volume to be activated.

d. `> QUIT MAPCI`

---

2 Ensure all regular and parallel devices are working for all available billing subsystems in DIRP.

Procedure 13
Erase application files

1 **App** Clean up SFDEV by erasing any application-related files (for example: DRNOW, FEATDATA, and all patches).

2 **App** Clean up disks by erasing any application-related files (for example: JFA, CFWA . . .).

Procedure 14
Return SLMs to service

1 **App** Recover SLM components as follows.

```
> MAPCI;MTC;IOD
> SLM 0;BSY;RTS
> SLM 1;BSY;RTS
> QUIT MAPCI
```

2 **Site** Restore any needed Site files to SFDEV.

Procedure 15
Start Journal File & LOGOUT

1 **App** Start journal file and verify started.

```
> JF START
```

```
> MAPCI;MTC;IOD;DIRP
```

```
> QUERY JF
```

Note: QUERY JF should respond with “AVAIL.” If a standby device is being used, both active and standby volumes should be marked “AVAIL.”

```
> QUIT ALL
```

```
> LOGOUT
```

Repeat this step for all journal file devices.

Procedure 16
Take images

1 **Site** Dump office images for backup—one SuperNode image to SLM disk 1 and backup the image file to SLM tape cartridge.

2 **Site** Send logs to tape for 24 hours and retain on site for 2 weeks.

3 **Site** Update the software change control log.

Revert to SNSE procedure

Perform this procedure if a controlled REVERT is required after the conversion to the SuperNode has already occurred.

Warnings

Before beginning, confer with online support for any possible changes to the procedure to minimize the system outage.

Using the LTP and TTP levels of the MAP, and TOPS operator(s), verify all essential services or high profile customers (that is, police and emergency bureaus, hospitals, and radio stations) are not in emergency call processing mode.

In *Procedure 1: Revert to SNSE* (below) work quickly to minimize downtime once **step 6** is performed—Call processing ceases until that step is complete.

The Installer throws the master switchboxes only when the software delivery engineer (App) indicates to do so.

CRITICAL WARNING: No activity is to be performed on the SuperNode CM, MS, or CLOCK until it is cleared by the software delivery engineer. **Failure to comply may result in a system restart.**

Confirm FX voice line numbers with Site and establish voice contact on same if this is not already the case.

Procedure 1
Revert to SNSE

1 **Site/SN** Perform manual DIRP subsystem transfers, and remove subsystem tape or close the active file on disk.

```
> MAPCI;MTC;IOD;DIRP
> ROTATE <subsystem>
> YES    {for confirmation}
```

Repeat for each DIRP subsystem.

Put up a tape on MTD 0. Go to DSKUT and do LISTVOL on the volume with the unprocessed files you want to copy.

Enter either

```
> DIRPAUTO <subsystem>    {steps you through the process}
```

or MOUNT the tape and enter

```
> DIRPCOPY <subsystem> <filename> T0
```

where <filename> refers to the unprocessed DIRP files. These start with the letter “U” (for example, U890327133614AMA).

Repeat for each DIRP subsystem.

2

```
> RFMT TAB DIRPPOOL ALL PUT FILE DIRP_INAC SFDEV
```

---

3 **App/SNSE** Perform a RESTART RELOAD on the SNSE.

```
> RESTART RELOAD
> YES
```

---

4 **App/SNSE** LOGIN on the SNSE, and DMNT and MNT any dead subsystems at the DIRP level of the MAP.

```
<break>
? LOGIN
Enter user name and password   {or enter each separately}
>
> MAPCI;MTC;IOD;DIRP
> QUERY AMA ALL   {Recover DIRP subsystems as required}
```

—continued—

Procedure 1
Revert to SNSE (continued)

5 **SN and SNSE/All** Revert to the SNSE.

a. **App/SN** Post the MS SHELF 0 level of the MAP to observe and verify that all PORTS go “S” (SYSB) once the master switches are thrown (this may take some time depending on the office size).

b. **Inst/MSBX** Verify the “MS” LED is “ON” and the “DISC” LED is “OFF” at the master switchboxes.

c. **Inst/RSBX** Verify the “LOC” LED is “OFF” at all the remote switchboxes, and confirm to the APP.

Note: If any “LOC” LEDs are “ON”, the remote switchbox may be defective.

d. **Inst/RSBX** Once all the “LOC” LEDs are confirmed “OFF” at all remote switchboxes, the Installer throws the switch from the “MS” position to the “CMC” position.

---

6 Switch activity from the SuperNode to the SNSE.

a. **App/SNSE** Perform a RELOAD restart.

```
> RESTART RELOAD
```

Warning: As soon as the restart is confirmed (by entering “YES” below), IMMEDIATELY GO TO **substep b** BELOW.

```
> YES
```

b. **Inst and App/MSBX** Applicator instructs the Installer to throw the master switchboxes from the “MS” position to the “CMC” position.

c. **Inst and App/MSBX** The Installer confirms to the Applicator that the “MS” LED goes “OFF” and the “CMC” LED is “ON” all the master switchboxes.

d. **Inst/SNSE** Verify the SNSE is flashing “A1” (within 5 minutes).

Procedure 2
LOGIN on SNSE & POSTSWACT

1 **Site and App/SNSE** LOGIN on the SNSE.

Warning: Work quickly to complete this step.

```
<break>
```

```
? LOGIN
```

```
Enter user name and password
>
```
{or enter each separately}

Confirm the date and time.

```
> DATE
```

Warning: If the date and time are not correct, alert the update engineer immediately so the date and time can be corrected before DIRP subsystems are allocated. If the date is incorrect, change it using: SETDATE <dd mm yy>. If the time is incorrect, change it using: SETTIME <hour> <minute>.

```
> DMOPRO DIRP_INAC
```
{if this file is present}

```
> TRACECI CLOSE
```

---

2 **App/SNSE** Confirm all DIRP subsystems are incrementing, and audit to remove the EMG indicators.

```
> MAPCI;MTC;IOD;DIRP;QUERY <subsystem>
```

```
> AUDIT <subsystem>
```

```
> YES
```
{for confirmation}

Repeat for all DIRP subsystems.

Note: If any DIRP subsystems are not available, use MNT command.

---

3 **Site** If the DATA DUMP field in the table CRSFMT is set to Y for SMDR, an additional rotate must be performed in order for the identifier translator tables to be dumped to the SMDR records.

Procedure 3
Assure peripheral recovery

1 **App/SNSE** Go to the NET level of the MAP and verify that all networks are in service.

```
> NET
> BSY <plane> <pair>;RTS <plane> <pair>
```

{for network planes not INSV}

Repeat for each network plane not INSV.

---

2 **App/SNSE** List the device containing the new PCL PM loads.

```
> DISKUT
> LISTVOL <pmload_volume> ALL
> QUIT
```

---

3 **App/SNSE** At the PM level of the MAP recover peripherals as needed.

```
> PM
```

a. For all SYSB PMs not under 'Maintenance':

```
> POST SYSB
```

```
> RTS UNIT <nn>
```

or

```
> BSY UNIT <nn>;RTS UNIT <nn> FORCE NOWAIT
```

where <nn> refers to the ACTIVE unit

```
> NEXT
```

Repeat as required.

b. For all MANB PMs:

```
> POST MANB
```

```
> LOADPM UNIT <nn> CC
```

or

```
> LOADPM UNIT <nn> CC NOWAIT
```

where <nn> refers to the ACTIVE unit

```
> RTS UNIT <nn>
```

or

```
> RTS UNIT <nn> FORCE NOWAIT
```

where <nn> refers to the ACTIVE unit

```
> NEXT
```

Repeat as required.

Procedure 4
Assure carrier & trunk recovery

1 **App/SNSE** Go to the CARRIER level and confirm circuits are in proper states according to SNSE data.

---

2 **App/SNSE** Verify log systems have started or start them manually.

---

3 **App/SNSE** Correct trunk states.

```
> MAPCI;MTC;TRKS;TTP
> POST A INI
> SEND SINK
> REPEAT 100 (BSY;RTS;NEXT)
```

Repeat until no more trunks are posted.

```
> SEND PREVIOUS
> QUIT MAPCI
```

Put up the Retrofit delta tape.

```
> MOUNT <x>;LIST T<x>
> COPY MB$TRKS SFDEV
> COPY RES$TRKS SFDEV
> DEMOUNT T<x>
> LISTSF ALL
> MAPCI NODISP;MTC;TRKS;TTP
> READ MB$TRKS
> READ RES$TRKS
```

Procedure 5
Assure attendant console recovery

1 **Site and App/SNSE** Ensure attendant consoles (ATTCONS) are in service. If any are not in service

```
> MAPCI NODISP;MTC;LNS;LTP;LEVEL IBNCON
```

```
> SELECT C <n>;BSY;RTS
```

where <n> is console number

Repeat until all consoles are in service.

Procedure 6
Check for dialtone

1 **Site/SNSE** If not already done,

Check for dial tone on all LMs, RLMs, LCMs, or RLCMs.

Execute critical call processing tests.

---

2 **Site/SNSE** Start journal file per NTP 297-1001-127.

Procedure 7
Stabilize front end

1 **App/SNSE**

At the CMC level of MAP, ensure both CMCs show status of “.” (dot).

Go to SYNCLK level and sync the clocks.

```
> MAPCI;MTC;CMC;SYNCLK;SYNC 0;SYNC 1
> QUIT MAPCI
```

2 **App/SNSE**

Sync the CC if a RESTART RELOAD was used in Procedure 1, step **6**.

```
> MAPCI;MTC;CC;SYNC
> QUIT MAPCI
```

Procedure 8
Restore IOC 1 & DIRP

1 **SNSE and App/SN** Move IOC 1 and higher to the SNSE.

SN Set MINFILES to 0 (zero) for all subsystems.

Turn ON access to RWOK using TASTOOLS.

```
> TABLE OFCSTD;VER OFF;OVE;VER OFF
> REP DIRPKILL_IN_EFFECT Y
```

Turn off access to RWOK.

```
> TABLE DIRPSSYS;OVE;VER OFF
> CHANGE MINFILES 0;DOWN
```

Repeat until “Bottom” message.

```
> QUIT
```

2 **Site/SN** Close, copy, and demount all subsystems.

```
> MAPCI;MTC;IOD;DIRP
> CLOSE <subsystem> ACTIVE
```

Repeat CLOSE for all DIRP subsystems.

Put up a tape on MTD 0. Go to DSKUT and do LISTVOL on the volume with the unprocessed files you want to copy.

Enter either

> DIRPAUTO <**subsystem**> *{steps you through the process}*

or MOUNT the tape and enter

> DIRPCOPY <**subsystem**> <**filename**> T0
where <filename> refers to the unprocessed DIRP files. These start with the letter “U” (for example, U890327133614AMA).

Repeat for all unprocessed files.

> DMNT <**subsystem**> *{retain for later}*

Repeat for all DIRP subsystems.

Stop all log devices on IOC 1 and higher. On IOC 1 and higher: BSY and OFFL all devices, BSY IOC, OFFL IOC.

Switch IOC 1 and higher by removing the SuperNode cables and replacing them with “C00” and “C01” connectors.

Note: Cable C00 must connect to CMC 1 and cable C01 must connect to CMC 0.

—continued—

Procedure 8
Restore IOC 1 & DIRP (continued)

3 **Site and App/SNSE** RTS IOC and all devices on IOC 1 and higher.

Site Assign DIRPSSYSs on IOC 1 and higher. If on disk, verify no volume “IN ERROR” with QUERY <subsystem> ALL.

---

4 **Site/SNSE** If DIRP subsystem is on disk.

```
> MAPCI;MTC;IOD;DIRP
> MNT <subsystem> <disk>
```

{subsystems from previous procedure}

Repeat MNT for all DIRP subsystems on disk.

---

5 **Site/SNSE** If DIRP subsystem is on tape or BMC.

```
> MOUNT <x> FORMAT <volume_name>
> DEMOUNT T<x>
> MAPCI;MTC;IOD;DIRP
> MNT <subsystem> T<x>
```

{subsystems from previous procedure}

Repeat this step for all DIRP subsystems.

---

6 **Site/SNSE** Restore parallel subsystems that were assigned.

```
> TABLE DIRPSSYS;POS <ssys_name>
> CHANGE PARVOL <vol_name>
> QUIT ALL
```

Procedure 9
Collect DPP settings

1 **App** If DIRP subsystem is DPP, perform this step while the site performs their tests. Display the current DPP settings to hard copy, and change if different from the data obtained during procedure *Switch IOCs to SuperNode* (above).

a. `> MAPCI NODISP;MTC;IOD;DPP AMA`

b. `> COLLPSW`
*If different, perform **substeps c** and **d**; otherwise, go to **substep e**.*

c. `> COLLPSW 1 <4_digits> <6_digits>`

d. `> COLLPSW 2 <4_digits> <6_digits>`

e. `> AMATPSW`
*If different, perform **substep f**; otherwise, go to **substep g**.*

f. `> AMATPSW <4_digits> <6_digits>`

g. `> AMAHRS`
*If different, perform **substep h**; otherwise, go to **substep i**.*

h. `> AMAHRS <start_hour> <end_hour>`

i. `> VALPARM INVALID`
*If different, perform **substep j**; otherwise, go to **substep k**.*

j. `> VALPARM INVALID <threshold>`

k. `> ERRMAP ACT`
*If different, perform **substeps l** and **m**.*

l. `> ERRMAP <alarm_no> <type> <level>`
Repeat for each alarm that is different.

n. `> VS STDBY`
Verify the loadname is correct for the to_PCL load and firmware is at least DOS 21.08.

Warning: Do not continue if the DPP load name or firmware version is incorrect.

Procedure 10
Clear DIRP alarms

1 **Site and App/SNSE** Close all DIRP standby subsystems to clear the “I” alarm under the IOD banner of the MAP.

```
> CLOSE <subsystem> STDBY1
```

Repeat CLOSE for all DIRP subsystems.

```
> QUIT MAPCI
```

Procedure 11
Restore MPC loads

1 **App/SNSE** For offices equipped with MPC boards and feature NTX273AA.

a. List the device which contains the MPC peripheral load.

b. Access table MPC.

```
> TABLE MPC;OVE
```

c. Replace all tuples in table MPC in order to bind the load name to the FID.

```
> CHANGE
```

```
> <cr>
```

Repeat the “<cr>“ until “Tuple to be changed” message.

> Y *{for confirmation}*

```
> DOWN
```

Repeat this substep (**c**) until “Bottom” message.

Procedure 12
Restore office parameters

1 **App/SNSE**

a. If NTX077AA is present, verify table DSLIMIT, tuple STOREFS, is set to 458752; otherwise, skip to **substep 1b**.

Turn ON access to RWOK using TASTOOLS.

```
> TABLE DSLIMIT;VER OFF
> POS STOREFS
> CHA DSMAX 458752
```

Turn off access to RWOK.

b. If NTX077AA is *not* present, verify table DSLIMIT, tuple STOREFS, is set to 65536.

Turn ON access to RWOK using TASTOOLS.

```
> TABLE DSLIMIT;VER OFF
> POS STOREFS
> CHA DSMAX 65536;QUIT
```

Turn off access to RWOK.

---

2 **App/SNSE** Change tuple NODEREXCONTROL in table OFCVAR back to the original value.

Procedure 13
Cleanup and LOGOUT

1 **App/SNSE** Erase all dump and restore, and conversion files from SFDEV.

2 **SNSE and App/SN** LOGOUT

Appendix A: Command Summaries

Appendix A table of contents

Using TABAUDIT and AUTOTABAUDIT

Checking tables for data inconsistencies prior to a software delivery is necessary to ensure that all tables are free of datafill errors and each table has valid data. This is accomplished by running TABAUDIT. This tool is provided to thoroughly verify the integrity of table data on a DMS switch. TABAUDIT will be run with the DMS switch in sync. TABAUDIT will only allow one session to be active at any one time.

CAUTION

Review Software Delivery Bulletins and all current Warning Bulletins concerning TABAUDIT before using it.

How TABAUDIT works

TABAUDIT will verify DMS tables by performing a series of checks on the tables and on the data in the tables. Following is a list of the types of checks that are performed on each table:

- Generic table checks,
- Syntax checks,
- Table specific checks, including routing checks

The generic table integrity checks consist of verifying that each table has a top and a bottom and that no tuples exist beyond these boundaries. Tables are also checked to ensure that no infinite tuple loops exist.

Syntax checks are done on a per tuple basis. Each field in a tuple is checked against its data dictionary definition in order to ensure data validity. These checks are modified to use the logical tuple definition and not the custflds representation.

The routing checks are only performed on tuples that reference routing tables. This type of check verifies the tuple being referenced in the routing tables.

Table-specific checks are done on a per-tuple basis. This type of check verifies the tuple's data for valid data options. (This is implemented using verify procedures in table control software.)

TABAUDIT example

```
> TABAUDIT ALL D000SCRATCH
Starting DMS Data Verification...
DATA VERIFICATION COMPLETED.
```

```
> LIST D000SCRATCH
SUMMARY$FILE
CCTR$FILE  (NOTE: This table failed the data check)

> PRINT SUMMARY$FILE
Tbl DART     : tuples checked  859,  passed 859,  failed  0.
                                 •
                                 •
Tbl DDU      : tuples checked    5,  passed   5,  failed  0.
                                 •
                                 •
Tbl CCTR     : tuples checked    15,  passed   14,  failed  1.

> PRINT CCTR$FILE
TABLE CCTR: New Table Control.
DATA IN ASSOCIATED ROUTING TABLE NOT PRESENT
---ERROR: Data does not verify.
    POSITION:  55
Completed tuple checking.
SUMMARY: Tbl CCTR: tuples checked 15, passed 14, failed 1.
```

TABAUDIT help

```
> HELP TABAUDIT
TABAUDIT Command
----------------
The TABAUDIT options are:
   ONLY  - Checks a single table. The output can be
           redirected to a specific device and filename.
   ALL   - Checks all tables listed in table DART. The
           output can be redirected to a specific device.
   FROM  - Checks all tables following and including a
           given table listed in table DART. The output
           can be redirected to a specific device.
           The FROM command can also be followed by a
           TO option to specify where TABAUDIT should
           stop checking. The table following the TO
           option will be included in the check.

The default output device is the screen and the default
filename is the table name concatenated with the string
$FILE. If the ALL or FROM option is chosen, TABAUDIT
creates a SUMMARY$FILE that lists the results of the
tables it has checked.

The results can be sent to SFDEV, disk, or tape. The
summary file will always be sent first to SFDEV and then
```

```
later copied to the user specified device and not
erased from SFDEV.

      eg1: TABAUDIT ONLY ofcstd
      eg2: TABAUDIT ONLY ofcstd sfdev temp
      eg3: TABAUDIT ALL sfdev
      eg4: TABAUDIT FROM ofcstd d000scratch
      eg5: TABAUDIT FROM custab TO ofcstd sfdev

WARNING: Only one TABAUDIT should be running with the
         ALL or FROM option.

Parms: <Function> {ONLY <Table Name> STRING
                        [<Device Name> DEVICE name]
                        [<File Name> STRING],
                   ALL [<Device Name> DEVICE name],
                   FROM <Start table Name> {(otherwise)
                        [{TO <End Table Name> STRING}]}
                        [<Device Name> DEVICE name]}
```

TABAUDIT enhancements (BCS36 and higher)

In BCS36 features were introduced to facilitate TABAUDIT’s usage by:

- automating TABAUDIT based on a scheduled time table.
- implementing a report facility that will
  - report tables that have not been checked.
  - report the time and date of the last check performed on a table.
  - report table specific data errors including routing errors.
  - report syntax errors.
  - report generic table integrity errors such as false tops, bottoms and holes in tables.
- linking TABAUDIT and TABXFR in order to identify data issues before data move is started.
- adding a PRECHECK step that verifies that all tables in DART have been verified.

TABAUDIT AUTO help

```
TABAUDIT:
> HELP
The TABAUDIT increment is used to setup a standard
session of TABAUDIT.
```

```
The increment consists of the following subcommands:

INCLUDE EXCLUDE STATUS REPORT CLEAR EXECUTE
AUTO QUIT HELP INFO

From within the TABAUDIT increment type:

  HELP <subcommand>
for further help on subcommand.

Note: The AUTO subcommand is used to enter the
      AUTOTABAUDIT increment.

> HELP AUTO
AUTO command
Command to enter the AUTOTABAUDIT level of TABAUDIT.
Note: Only one user may occupy this level at a time.

       eg1:  AUTO

All AUTOTABAUDIT commands entered from the AUTOTABAUDIT level.
Use the TIMEFRAME command to schedule the automated
verification of tables.
```

Automated TABAUDIT

This BCS36 feature provides an automated TABAUDIT process. Once the initial parameters are entered, automated TABAUDIT has the ability to check data integrity without external guidance. The results of the data checks are maintained, and can later be displayed via a report utility.

Automated TABAUDIT does not generate reports for each table as the table is being verified the way TABAUDIT does. Instead, all report data may be accessed via the report utility. TABAUDIT, in addition to generating reports as tables are verified, is altered so that such data may be accessed in the same manner as automated TABAUDIT’s data, via the report utility.

The parameters required to initiate the automated tabaudit are entered via the CI increment TABAUDIT. The original TABAUDIT command is moved into the TABAUDIT command increment. A second level containing the automated TABAUDIT commands is accessible from within the TABAUDIT increment.

The automation of TABAUDIT via a schedule provides the user with the ability to set a time frame within which TABAUDIT should be running. The schedule that governs TABAUDIT’s execution is based on a 24 hour clock cycle. TABAUDIT activates daily to verify tables’ data integrity during a specified time frame until all data is verified. The user may also specify the date on which the TABAUDIT is to be started and stopped by the scheduler.

Automated TABAUDIT is setup from within the CI increment, TABAUDIT, which allows the user to specify tables to be verified, when the verification is to take place, the type of data integrity report[1], and the report's destination. The following lists the various functions added by this feature via the TABAUDIT increment to facilitate the use of TABAUDIT:

- Schedule the verification of a single table in DART.
- Schedule the verification of a range of tables in DART.
- Schedule the verification of all tables in DART.
- Generate a data integrity report for all tables in DART.
- Generate a data integrity report for all verified tables.
- Generate a list of tables that have not been checked by TABAUDIT.
- Generate a data integrity report for a single table.
- Display the status of the scheduler.
- Activate or deactivate the scheduler.
- Clear scheduling information.
- Have the ability to specify select tables to be included in a table range
- Have the ability to specify select tables not to be checked.

Automated TABAUDIT uses table DART for the order in which to verify tables' data integrity. Table DART, if not previously sorted, is sorted using the dump and restore ordering in advance of any data integrity checks.

Automated TABAUDIT may be in one of two states:

- Active: The scheduler is executing TABAUDIT during specified time frames.
- Inactive: The scheduler is not executing TABAUDIT, and execution of TABAUDIT is not scheduled.

For standard tabaudit the STATUS command displays the range of tables included and excluded. The ALL option can be used to display all tables included and excluded.

The STATUS command displays the current parameters being used by the automated TABAUDIT process. The parameters that are displayed consist

[1] A data integrity report consists of the results of the three types of checks mentioned under "How TABAUDIT works" above.

of: all included table ranges and their indices, the start time, the start date, the stop time, the stop date, whether or not automated TABAUDIT is currently active, and current table. If the ALL option is used, the parameters that are displayed consist of: the all included and excluded tables and their indices, the start time, the start date, the stop time, the stop date, whether or not automated TABAUDIT is currently active, and current table.

When activating TABAUDIT, the current information the TABAUDIT session is working with is displayed. The user is asked to confirm the data. If data is missing, the user is told what data is missing. If for some reason a session of tabaudit can not obtain the resources it requires, the user is notified that the tabaudit session was terminated, and for what reason.

When QUITing out of a standard tabaudit session, the user loses that session’s parameters. When QUITing out of an automated tabaudit session, the user is placed back in the standard tabaudit increment, and as long as the automated session is active, the session parameters are preserved.

All data that automated TABAUDIT captures is stored in protected store. This includes table statistics (time, date and error counts), and the keys of tuples containing errors.

Automated TABAUDIT has a couple of restrictions. Sessions of automated TABAUDIT and TABAUDIT can be running at the same time, however; only one session can be verifying a specific table at a time. Two sessions of automated TABAUDIT can not be both running at any given instant. Furthermore, once an automated tabaudit session is active, no changes can be made to the session’s parameters without first terminating the session.

Examples:

The following commands will enter the TABAUDIT increment at the AUTO level, set the time frame within which TABAUDIT is to execute, and specify the range of tables to verify the number of verification cycles to perform.

```
> TABAUDIT
TABAUDIT:

> AUTO
AUTOTABAUDIT:

This level of the Tabaudit CI is used to setup a
scheduled session of Tabaudit. The TIMEFRAME
command is used to specify the time frame within
which the verification of tables is to be performed.
```

```
              AUTOTABAUDIT:

              > TIMEFRAME 1:00 30:06:92 5:00 05:07:92
              Is the following schedule correct?

              Automated Tabaudit is to execute from 1:00
              to 5:00 between the following dates:

              Start date: 1992/06/30
              Stop date: 1992/07/05

              Please confirm ("YES", "Y", "NO", or "N"):
              > Y

              > INCLUDE FROM  LTCINV TO KSETFEAT

              > EXECUTE
+--------------------------------------------------------------------------+
|                                                                          |
|                        AUTOMATED TABAUDIT STATUS                         |
|                                                                          |
|--------------------------------------------------------------------------|
|  Active Timeframe                  |  Executing Timeframe                |
|--------------------------------------------------------------------------|
|    Start                 Stop      |    Start                 Stop       |
|    Date                  Date      |    Time                  Time       |
|--------------------------------------------------------------------------|
|  1994/06/30           1994/07/05   |    1:00                  5:00       |
+--------------------------------------------------------------------------+

              Current time           : 1994/06/23  16:32:05
              Automated Tabaudit     : Inactive

              The following tables are INCLUDED

                  From table LTCINV (52)           to table KSETFEAT (420)

              The following tables are EXCLUDED

                  From table DART (0)              to table XLIUMAP (51)
                  From table DNINV (421)           to table CNMDBTST (1115)

              Please confirm ("YES", "Y", "NO" or "N"):
              > Y

              Automated Tabaudit has been activated.
```

Reporting utility

The report utility is initiated by the CI command, REPORT command in the TABAUDIT directory. This command has five options:

- CHECKED
- NOTCHECKED
- ALL
- INCLUDED
- ERRORS
- <table_name>

The report utility’s CHECKED function generates a list of all tables that have been checked, the time and date of the last check, as well as the number of each type of error encountered. At the end of the report, the total number of errors encountered and the amount of time TABAUDIT required to perform the checks are displayed.

The report utility’s NOTCHECKED function generates a list of all tables that have not been verified.

If the ALL function is specified, a full report of all tables in table DART is generated indicating whether the table has been checked, the table has been checked and passed verification, or the table has been checked, but an error has been encountered. The number of errors, the error types, and time and date of data verification are indicated. For every table, each failed tuple’s key is listed. The output may be directed to a specific device. In addition, at the end of the report, the total number of errors and the total amount of time that TABAUDIT was executing are displayed.

The INCLUDED function generates a report of all tables in a specified includes list. The report details whether or not a table has been checked, and whether or not it passed all checks. If a check failed, the number of each type of error, time and date of the check as well as the failed tuples key are displayed. In addition, at the end of the report the total number of errors encountered in the tables in the includes list, as well as the total amount of time TABAUDIT was executing are displayed.

The ERRORS function is identical to the ALL function except reports are generated only for those tables known to have errors.

If a table name is specified, a report of that table’s data integrity is generated. The number of errors, the error types, and time and date of data verification will be indicated. The key of each tuple that failed is displayed.

TABXFR linking

If all data issues have not been resolved and rechecked by TABAUDIT before a data move is attempted, a warning is displayed. The types of warnings are:

- WARNING - ## tables have not been verified by TABAUDIT.
- WARNING - ## errors encountered by TABAUDIT.
- WARNING - ## tables not checked by TABAUDIT within the last 30 days.

Note: ## represents an integer.

Refer to the report utility to determine which tables cause the warnings above.

PRECHECK step

A precheck step is used to ensure that any tables that have not been checked, or any errors that have not been fixed are brought to the Applicators attention. The number of all tables that did not pass TABAUDIT's checks is displayed via the following warning messages:

- WARNING - ## tables not checked by TABAUDIT.
- WARNING - ## errors encountered by TABAUDIT.
- WARNING - ## tables not checked by TABAUDIT within the last 30 days.

Note: ## represents an integer.

Refer to the reporting utility to determine which tables cause the above warnings.

TABAUDIT enhancements (CSP07 PCLs and higher)

In CSP07 enhancements to TABAUDIT provide users with a further automated TABAUDIT process, enhanced scheduling capabilities and timeframe specification, and a flexible human-machine interface.

The enhancements to TABAUDIT can be broken down as follows:

Enhancement to AUTOTABAUDIT scheduling capabilities

- Allow iterative execution of the automated TABAUDIT during the specified timeframes without human interaction.
- Allow TABAUDIT to start in the middle of a timeframe.
- Make TABAUDIT contention free with image dump and Tabxfer, etc.

Enhancements to Human-Machine Interface (HMI)

- Option to exclude tables which are not transferred during the One Night Process (ONP).
- Option to include all previously failed tables.
- Option to include all tables which have been modified since the last audit.
- Capability to specify timeframe with daily, weekly and monthly options
- Capability to modify timeframes through table control.
- Allow users to clear failure information for those tables which will never pass the check.
- Allow all users to enter AUTOTABAUDIT level and check the execution status, but only one user can issue the execute command and terminate an active AUTOTABAUDIT session.

Enhancements to Reporting Capabilities

- Correct the report command to show corrupted tables (i.e. no top, no bottom, holes etc.).
- Enhance the report command to display all tables modified since the last audit.
- Eliminate the generation of a TABAUDIT output file on sfdev when a non-sfdev volume is specified in the **EXECUTE** ci command.

Man machine interface (MM)

The following CI commands of the TABAUDIT CI have been modified. The changes are shown with change bars.

The INCLUDE command

The include command is used to specify the range of tables to be verified by TABAUDIT. The modified command will allow users to specify all failed / changed tables. The syntax will be:

```
INCLUDE   ALL

          | FROM <table1> [TO <table2>]

          | FAILED

          | CHANGED

          | <table>
```

Examples:

```
> INCLUDE FAILED              % Include all the failed tables from the last audit

> INCLUDE FROM termdev TO ltcinv      % Include table termdev to ltcinv

> INCLUDE CHANGED             % Include tables changed since last audit
```

The EXCLUDE command

The exclude command provides users the ability to specify a table that is not to be checked by TABAUDIT. The enhanced command will also provide users the option to disallow all non D/R tables (excluding the dynamic tables) to be checked by TABAUDIT. The syntax of the command will be enhanced as follows:

```
EXCLUDE   <table>

          | NODR
```

Examples:

```
> EXCLUDE linestat            % Exclude table linestat

> EXCLUDE NODR                % Exclude all non dump and restore tables
```

The CLEAR command

The clear command is used to clear the included table list and the scheduled timeframes. The new modified command will also allow users to clear table failure information. The syntax will be:

```
Manual TABAUDIT:   CLEAR    INCLUDED
                   | FAILED <table>
Auto TABAUDIT:     CLEAR    INCLUDED
                   | SCHEDULE {ALL | ONE}
                   | ALL
                   | FAILED <table>
```

Examples:

```
> CLEAR SCHEDULE ALL    % Clear all specified timeframes
> CLEAR FAILED linestat % Clear error information for table linestat
```

The TIMEFRAME command

The timeframe command is used to specify the start time and stop time for automated TABAUDIT processing activities. The modified command will provide users the option to specify the weekday in the timeframes. The syntax will be changed accordingly to:

```
TIMEFRAME   SINGLE <start time> [<start date>] <stop time> [<stop date>]
            | DAILY <start time> <stop time>
            | WEEKLY <start time> {MON|TUE|WED|THU|FRI|SAT|SUN}
            | MONTHLY <start time> <day of month> <stop time>
              <day of month>
```

Note: The duration of the timeframe can not exceed six hours. For monthly option, the 31st day of each month is not allowed.

Examples:

```
>TIMEFRAME WEEKLY 08:00 MON 10:00 MON
    % Run every Monday 8:00 - 10:00 am
>TIMEFRAME WEEKLY 23:00 FRI 02:00 SAT
    % Run every week between 23:00 Friday to 2:00 Saturday
>TIMEFRAME DAILY 08:00 10:00
    % Run everyday 8:00 - 10:00 am
>TIMEFRAME SINGLE 11:20 12:30
```

% Start at 11:20 and stop at 12:30 today

```
>TIMEFRAME MONTHLY 07:00 07 09:00 07
```

% Run 7th of every month from 7:00 to 9:00

The REPORT command

The report command is used to generate audit result reports. Users can generate reports for all included tables, all tables with errors, all tables checked, all tables not checked, or all tables in table DART. The new report command will allow users to view the tables changed since the last audit. The modified syntax of the command is:

```
REPORT    NOTCHECKED [<device>] [<filename>]
          | CHECKED [<device>] [<filename>]
          | INCLUDED [<device>]
          | ERRORS [<device>]
          | ALL [<device>]
          | CHANGED [<device>] [<filename>]
          | <table> [<device>] [<filename>]
```

Examples:

```
> REPORT CHANGED              % Report all tables changed since last audit
> REPORT ofcvar vol2 errfile  % Generate errfile for table ofcvar on vol2
> REPORT INCLUDED vol1        % Report on all included tables on vol1
> REPORT ERRORS               % Report on all tables with errors
> REPORT NOTCHECKED           % Report on all tables not checked
> REPORT CHECKED              % Report on all tables checked
```

Some examples of the report are shown as follows:

The report format for the options checked, included, all, errors, and <table>:

```
DART Table Name      Table Control | Checked    Pass   Fail   Start Time                  Elapsed Time
--------------------  ---------------+----------  -----  -----  ------------                --------------
0:VERSIONS           New |     124       124      0      1996/07/06 23:31:22.55:           11.567

Total Number of Tables Reported for this report option:     1
Total Number of Passed Tuples for this report option:      124
Total Number of Failed Tuples for this report option:      0
Total Number of Verified Tuples for this report option:     124
Total Elapsed Time to verify the above tables:            :11.567
```

The report format for the option notchecked:

```
DART Table Name     Table Control | Checked     Pass   Fail   Start Time                  Elapsed Time
--------------------  ----------------+---------   -----  -----  ------------                ---------------

   1:ACTPATCH        New  : Table has not been checked.
   2:VRINV           New  : Table has not been checked.
   3:SOCVAR          New  : Table has not been checked.
   4:SOCFEAT         New  : Table has not been checked.
   5:SOCOPT          New  : Table has not been checked.
--------------------  ----------------+---------   -----  -----  ------------                ---------------

Total Number of Tables Reported for this report option:         5
Total Number of Passed Tuples for this report option:      0
Total Number of Failed Tuples for this report option:      0
Total Number of Verified Tuples for this report option:     0
Total Elapsed Time to verify the above tables :            :00.000
```

The report format for the option changed:

```
DART Table Name            Table Control          Changed
--------------------       ---------------        ------------------------
0:VERSIONS                 New                    Table has been changed
3:SOCVAR                   New                    Table has been changed
5:SOCOPT                   New                    Table has been changed
--------------------       ---------------        ------------------------
Total Number of Tables Reported for this report option:3
```

The AUTO command

The auto command is used to place users in the automated TABAUDIT level. Currently, only one user can be in the AUTOTABAUDIT level.

The syntax of this command is:

```
AUTO
```

The modified command will allow any users to enter the automated TABAUDIT level, display the status of execution and print the report, even if there is a user already at the automated TABAUDIT level. The restriction of allowing only one user to invoke the **EXECUTE** ci command is still in effect. The syntax of the command will not be changed.

Restrictions/limitations

The new CI command Report changed is used to show all the tables that changed since the last audit. However, tables changed due to side effects may not be reported by this command.

The enhanced timeframe command can specify the monthly execution of AUTOTABAUDIT. However, it does not allow to specify the 31st day of a month by the monthly option.

The duration of the timeframe is restricted to maximum six hours.

DARTEDIT syntax

DARTEDIT — Enters the DARTEDIT increment to enable the following commands.

PRINTDART <mode>

Where <mode> can be:

SHORT — Prints a compact listing of tables in DART. (default)

LONG — Lists tables in DART in a single column format.

Examples:

```
> PRINTDART SHORT
Table DART:
 000 N DART 001 N OKPARMS 002 E DATASIZE 003 N SYSDATA 004 E OCCNAME
 005 E OCCINFO            006 E HNPACONT 007 N OFCSTD  008 N OFCOPT
 009 OFCENG 010 E CRSFMT  011 E CLLI     012 E NNASST  013 E SITE
 014 E CMSHELF...

> PRINTDART LONG
Table DART:
 000 N DART
 001 N OKPARMS
 002 E DATA SIZE
 003 N SYSDATA
 004 ED OCCNAME
 005 E OCCINFO...
```

SORTDART <mode>

Where <mode> can be:

SHORT — Prints a compact listing of tables in DART alphabetically. (default)

LONG — Lists tables in DART alphabetically in a single column format.

FIND <table> — Finds the table(s) starting with...

Example:

```
> FIND TERMDEV
 041 E TERMDEV
```

DELTA syntax

This command may be used to display the differences (delta) between a given table on the active and inactive sides.

DELTA <table> <delta level> <file option>

Where <table> is the name of the table to be checked. For a sub-table use for the table name:

[<table> SUB <subtable>]

<delta level> can be:

COUNT — Checks the number of tuples in both tables.

KEYS — Does a checksum on the keys of the tuples.

CHECKSUM — Does a checksum on the tuples (default).

And <file option> can be:

FILE — Sends results to a file on SFDEV called DELTA$REPORT (default).

NOFILE — Displays the terminal without creating a file.

Example:

```
> DELTA OFCENG NOFILE
Table OFCENG
   Tuples do not match.  Old Count = 209,  New Count = 206.
**OLD  AMA_EBCDIC_CONVERT N

**OLD  CUSTOMER_GROUP_IBNGRP_OM_COUNT 288
**NEW  CUSTOMER_GROUP_IBNGRP_OM_COUNT 32
```

TABXFR summary

In BCS35 and higher the TABXFR increment is used to dump and restore tables. This is also known as the “data transfer.”

Interrupt TABXFR procedure

Procedure A-1
Interrupt TABXFR

During the data transfer the user can stop the TABXFR process two different ways: HALT is to be used most of the time; whereas, HALT NOW can be used to stop the data transfer of a long table.

Note: For the ACTREST platform the following commands are the same, except on the *active* side.

1 **App/INACT** To *halt* TABXFR, on the inactive terminal enter:

```
Mate > HALT
```
{must be in TABXFR increment}
TABXFR process will halt after the current table is completely moved. This will not interrupt tables in the process of being moved.

---

2 To *stop* TABXFR during the data transfer of a long table, on the inactive terminal type:

```
Mate> HALT NOW
```
{in TABXFR}
TABXFR process will halt immediately, even in the middle of the table.

---

3 To *restart* the data transfer use the STARTXFR command. This starts from the last table successfully completed.

To restart TABXFR, on the inactive terminal enter:

```
Mate> STARTXFR
```
{still in TABXFR}

---

4 **ACT** You may, instead, ABORT (stop and reschedule) the application by typing on the ACTIVE side:

```
> BCSUPDATE;ABORT_PRESWACT
```
Resets DUMP_RESTORE_IN_PROGRESS bool in OFCSTD to “N”

```
> TABXFR;CANCEL
```
Enables Patcher/PRSM and turns on AUTODUMP and AUTOPATCH

```
> QUIT ALL
```

TABXFR syntax

TABXFR — Enters the TABXFR increment to enable the following commands.

TABXFR subcommands are as follows:

SETUP <options> — Set up and initialize the type of platform used to perform the data transfer.

Where <options> can be:

STANDARD — Standard split switch application.

DUMPONLY — Setup to perform the dump part of the dump and restore. This is for dumping data to tape.

ACTREST — Perform the restore part of the dump and restore (data transfer). This restores data from tape to the active side (i.e. restore to split cm mode).

THIRDPRO — Third processor-assisted data transfer.

STOPIF — Sets the number of failures allowed for a table. If this threshold is reached then the data transfer stops at the end of the table. (Range is 0 to 4294967296.)

LIMIT — Sets the number of failures allowed for a table. If this threshold is reached then the data transfer stops (whether the table has completed or not). (Range is 0 to 4294967296.)

STATUS — Displays information about the setup and status of the data transfer.

STARTXFR — Starts the data transfer process. It starts after the last completed table or at the beginning if no tables have been completed.

XFRONLY <table> — Performs a data transfer on the specified table only.

XFRFROM <table_name> — Starts a data transfer from the specified table.

CAUTION

XFRFROM places TABXFR control into a more manual state.

Use of the command may cause automatic processes to be bypassed, effecting transfer accuracy.

Using XFRFROM

- If XFRFROM is to be used, for whatever reason, it must be used through the entire TABXFR process.

  Transfer errors and logs can result if STARTXFR and XFRFROM are used interchangeably.

- XFRFROM must be used with the data found in Table DART—after TABXFR has returned the process message:

  “Table DART is now sorted”

  Table DART, after sorting, has each table associated with a number. <DART_number> <status> <table_name>

  For example: <u>0023</u> E OFCENG

  These numbers represent the order TABXFR will transfer all the tables. When using XFRFROM, the Applicator is responsible to follow this order.

  ****
  Note:
  XFRFROM must be used *only with non-recursive tables!* XFRFROM will skip tables if used on recursive tables.

  Recursive tables are identified by the following message text:

  “Table is recursive. Verify that any <table_name> data failures apply when table is run again later.”

  ****

- An errored non-recursive table may call many recursive tables before STOPIF can halt the transfer process.

- The non-recursive table logged as attempting transfer immediately before the recursive table(s) is the table which called them.

- As non-recursive tables reach STOPIF and are repaired, note the DART number of the table to be repaired. The table with the next higher DART_number is the table to be used with XFRFROM, once the errored table has been acceptably transferred.

DUMP — Performs just the dump portion of the data transfer. (Only available after issuing the SETUP DUMPONLY command.)

RMOUNT — Mounts the device to be used for the active restore.

RDEMOUNT — Demounts the device being used for the active restore.

RCOPY — Copies a file from the restore device to SFDEV.

DATASYNC — Manipulates the Data Synchronization. (Only available after issuing the SETUP THIRDPRO command.)

HALT — Stops the data transfer after the current table is completed.

HALT NOW — Causes the data transfer to halt immediately, after the current tuple.

CLEAR — Clears the specified table. This only works with tables that have a transfer type of PHYSICAL in table DART.

STOPXFR <stop_options>

Where <stop_options> can be:

BEFORE <table> — Stops TABXFR before given table.

AFTER <table> — Stops TABXFR after given table.

CLEAR BEFORE <table> — Clears the stop before given table.

CLEAR AFTER <table> — Clear the stop after given table.

QUERY — Lists all STOPs.

CANCEL — Cancels the data transfer. Entered on active side after any type of ABORT. Turns on AUTODUMP and AUTOPATCH.

REPORT — Generates a final data transfer report. This will include a table exception report and (with BCS36 and lower) an NTX package delta.

XREPORT — Creates a table exception report only.

NTXDELTA — Performs a delta of the NTX packages on the old and new loads (with BCS36 and lower).

QUIT — EXIT the TABXFR increment.

BCSUPDATE summary

This section describes the commands and program steps available within BCSUPDATE. The PRECHECK steps are listed first. Next is a “PRESWACT Abort” procedure followed by combined descriptions of the PRESWACT and POSTSWACT steps. (SWACT commands are described in the final section of this appendix, *CC WarmSWACT summary*.)

The BCSUPDATE increment enables the commands for the PCL application. BCSUPDATE level commands do the actual work of applying the new PCL software.

BCSUPDATE increments

The primary increments of BCSUPDATE that assist in switching activity from one BCS/PCL to another and in recovering from the activity switch are:

• PRECHECK	(all PCLs)
• PRECHECK FIRST	(BCS33 through BCS36)
• PRECHECK FINAL	(BCS33 through BCS36)
• PRESWACT	(all)
• SWACTCI	(BCS33 and higher)
• SWCT	(BCS31 and BCS32)
• POSTSWACT	(all)

The following processes or increments are available within BCSUPDATE:

PRECHECK — A series of pre-application checks (prechecks) used to ensure an office is ready for the software upgrade.

PRESWACT — Perform the PCL application by invoking the application driver. The command may be used repeatedly.

STATUS — Display the current status of the PCL application. It displays what critical steps have been completed (and the execution time), those still needed, and whether or not the system is ready to perform the switch of activity.

RESET — Reset all completed procedures to their initial state of NEEDED so BCSUPDATE can be re-executed.

SWCT — Enable CC WarmSWACT commands. In BCS31 this command became available only as an increment of BCSUPDATE.

SWACTCI — Same as SWCT. This changed to SWACTCI in BCS33.

Note: Commands for switching activity are available as increments of the SWACTCI/SWCT level. Refer to the next section, *CC WarmSWACT summary*, for details of SWACT commands.

POSTSWACT — Recovery functions following the SWACT.

OVERRIDE — Set a failed PRESWACT or POSTSWACT procedure to COMPLETED. This should be used with caution. Only those steps that have been investigated and pose no threat to the current application should be set completed by this command.

DATADUMP — Display office information (implemented in BCS31).

RUNSTEP — Execute individual PRESWACT steps. Starting in BCS34 this also works for POSTSWACT steps. CAUTION! Also see next command.

ABORT_PRESWACT — (BCS35 and higher) Used to abort PRESWACT or to recover after a RUNSTEP is used to run one of the preswact steps. Both PRESWACT and RUNSTEP will set the DUMP_RESTORE_IN_PROGRESS bool in OFCSTD to “Y.” After a RUNSTEP (if done out-of-process) the Applicator must run ABORT_PRESWACT to reset the bool to “N”.

QUIT — EXIT the BCSUPDATE level.

PRECHECK steps

To ensure that an office is ready for a software upgrade, a series of pre-application checks (or prechecks) must be completed. These checks are typically done at two points prior to the application date. The checks can also be performed by Nortel through the execution of separate checklists. For BCS33 and higher many of these checks that were previously manually verified can be done in advance by the site personnel using the PRECHECK command.

The PRECHECK command is valid for all PCLs. If coming from BCS36 (or lower) use the commands PRECHECK FIRST and PRECHECK FINAL.

The PRECHECK command, within the BCSUPDATE increment, will execute the steps described below. This is not a complete list, since the steps will vary from one software load to another.

```
Step Name: C7LINK_CHECK
-----------------------
Implementation: C7LSMTC1
Procedure Name: C7LINK_PRECHECK_PROC
Description: This step checks table C7LINK to ensure that the field LINKNAME
             is datafilled with all MSB7 or LIU7 type peripherals but not a
             mixture of either type peripheral.
Failure Paths: Field LINKNAME in table C7LINK has mixed peripheral types.


Step Name: CHECK_CRSFMT_AND_CRSMAP
----------------------------------
Implementation: APPLCHEK
Procedure Name: CHECK_CRSFMT_AND_CRSMAP
Description: This step checks table CRSMAP and CRSFMT in order to verify if
             Station Message Detail Recording (SMDR) data is being sent to
             BCFMT AMA stream. Since this data arrangement should not be used
             except for those sites which handle their data in a special way,
             a warning message will be displayed if this data arrangement is
             present.
Failure Paths: - Unable to read field STREAM from SMDR tuple in table CRSMAP.
               - Unable to read field FRMTAREA from AMA tuple in table CRSFMT.
               - SMDR data is being sent to BCFMT AMA stream.


Step Name: CHECK_LCMINV
-----------------------
Implementation: APPLLCM
Procedure Name: CHECK_LCMINV
Description: Step verifies memory on LCM(s) match MEMSIZE field entry
             in table LCMINV. Also verifies entries in table have a
             corresponding LCM which is in-service.
Failure Paths: - Unable to access table LCMINV.
               - Found memory mismatch between LCM and entry in table LCMINV
                 or unable to validate memory because of out-of-service unit.


Step Name: CHECK_LOGS
---------------------
Implementation: APPLYC1
```

```
Procedure Name: CHECK_LOGS
Description: Checks traps and critical logs to ensure front-end stability.
             The log output is in the following order: TRAP, INIT, SWER, CM,
             MS, SLM, MM, and CMSM. Each log is grouped by log class and
             appear in the latest to oldest order.
Failure Paths: Not Available.


Step Name: CHECK_LTCINV
-----------------------
Implementation: APPLLTC
Procedure Name: CHECK_LTCINV
Description: Checks DTCs, in table LTCINV, datafilled for CCS7
             (i.e. field OPTATTR equals "CCS7") to ensure XPM load name
             and processor compatibility.
Failure Paths: - Unable to access table LTCINV.
               - If unable to process entire table LTCINV.
               - If any incompatibility was found which generated a warning
                 message.


Step Name: CHECK_MEMORY
-----------------------
Implementation: APPLCMM
Procedure Name: CHECK_MEMORY
Description: Determines amount of physical, logical, spare, and available
             memory in the site. Used along with C-MAP reports and MEM-CALC
             tools. The amounts are displayed in one meg equivalents.
Failure Paths: Not Available


Step Name: DEVICE_CHECK
-----------------------
Implementation: STATCHK1
Procedure Name: ACTIVE_DEVICE_CHECK
Description: Verifies that all devices on the active CPU are in an OK,
             OFFLINE, or UNEQUIPPED status. These are the acceptable states
             for SWACT. Any devices found in any other state, will be
             displayed.
Failure Paths: - Fails if any nodes' status in the office are neither OK or
                 OFFLINE.
               - Fails if the status of any device is neither OK or OFFLINE.


Step Name: DISPLAY_DEVICE_AND_USER
----------------------------------
Implementation: APPLUSER
```

```
Procedure Name: DISPLAY_DEVICE_AND_USER
Description: Displays information of the IOC (or IOM) device and user
             information currently executing this step. Information includes:
             IOC card and port, COMCLASS, PRIORITY, STACK size, PRIVCLASS,
             etc.
Failure Paths: - Failed to open table TERMDEV.
               - Failed to read tuple from table TERMDEV.
               - Cannot access table LGINCTRL.
               - Failed to read tuple from table LGINCTRL.
               - Return False if any one of the following conditions exist:
                    - ComClass not equal "ALL"
                    - MaxIdleTime not equal "FOREVER"
                    - OpenForceout equals "Y"
                    - User Priority not equal 4
                    - User Stack Size less than 7000
                    - User Privilege Class not equal "ALL"


Step Name: DISPLAY_DNC_USERS
----------------------------
Implementation: APPLNOP
Procedure Name: DISPLAY_DNC_USERS
Description: Displays all DNC/MPC users and their status.
Failure Paths: Not Available.


Step Name: DISPLAY_PEC_INFO
---------------------------
Implementation: APPLYC1
Procedure Name: DISPLAY_PEC_INFO
Description: Displays PEC and release information on the CM and MS.
             Information is provided to verify the hardware is at the
             correct level (i.e. Baseline).
Failure Paths: Failure to interpret any CI commands.


Step Name: DISPLAY_PERIPHERAL_LOAD_NAMES
----------------------------------------
Implementation: APPLPMIM
Procedure Name: DISPLAY_PERIPHERAL_LOAD_NAMES
Description: Displays information regarding peripherals. Information
             includes: peripheral type, node no., and status of units
             0 and 1. Step also executes procedure variables which display
             load names for each equipped MPC, STC, DCH and DPP.
Failure Paths: - Unable to display XPM load names when node type is one of
                 the following: LTC, MSB, RCC, IAC, TAC, CSC and ESA.
               - Unable to display the following peripheral load names
```

```
                 when the node type matches the peripheral: LCM, DLM, XPE,
                 TPC, LIM, LIU, and NIU.
               - Unable to display RP load names when node type is
                 either AP or FP.
               - If unable to read peripheral information.
               - If any called procedure variables return false.


Step Name: E911SRDB_CHECK
-------------------------
Implementation: SRDBPDMC
Procedure Name: E911SRDB_CHECK
Description: Verify there are no unused values in data dictionary of type
             'SERVING_NUMBERING_PLAN_AREA'. This check is to ensure
             that table E911SRDB can be restored PHYSICALLY.
Failure Paths: - Unable to find type 'SERVING_NUMBERING_PLAN_AREA'.
               - Unused value in data dictionary of type
                 SERVING_NUMBERING_PLAN_AREA.


Step Name: LIUINV_CHECK
-----------------------
Implementation: LIUTC0
Procedure Name: LIUTCI_PRECHECK_PROC
Description: Ensures the largest number of a specific LIU type defined in
             table LIUINV does not exceed the maximum LIU type size which
             is 512. The LIU types are: LIU7, FRIU, EIU, XLIU, APUX, GSMP,
             and VPU.
Failure Paths: Total size of a specific LIU type defined in table LIUINV
               exceeds the maximum size of 512.


Step Name: TABAUDIT_VERIFY_TABLES
---------------------------------
Implementation: AUTOTBIM
Procedure Name: HAS_TABAUDIT_VERIFIED_TABLES
Description: Verifies that TABAUDIT has been executed and verified tables
             to be dump and restored in the last thirty days.
Failure Paths: Not Available.
```

PRESWACT Abort procedure

Procedure A-2
PRESWACT Abort

App It may be necessary to STOP (and reschedule) the application after PRESWACT (or TABXFR) has been implemented, but before the switch of activity. If this is the case perform the following steps to restore the active side to its original configuration.

1 **ACT** For BCS35 and higher (all PCLs) type only this command:

```
> BCSUPDATE;ABORT_PRESWACT
```

PRESWACT and POSTSWACT steps

The following steps are executed to set up the environment for the swact and to clean up after the swact. This is not a complete list of processes used by Preswact and Postswact, since steps will vary depending on the PCL software level and on certain features being present in the office.

```
Step Name: CHECK_ISN_PMS
------------------------
Implementation: APPLISN
Procedure Name: CHECK_ISN_PMS
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This procedure messages all sos based ISN pms to retrieve their
             current software level and version on the active side. The step
             passes if all ISN pms contain the same or a higher software
             level as the inactive CM.
Failure Paths: will occur when this procedure is unable to:
             - allocate a FTS resource
             - allocate a MTA resource
             - bind the process as a receiver
             - determine the pm name
             - match the mta of the ISN
             - send the message within 10 seconds
             - receive the message within 10 seconds
             - determine the mta of the ISN


Step Name: TRANSFER_CM_DATA
---------------------------
Implementation: CMPSIM1
Procedure Name: TRANSFER_CM_DATA
Procedure Type: ACTIVE_PROC
Description: Transfer the carry forward set data and the day of week for the
             full CM Routine Exercise (REX) set data from the active to the
             inactive side and update the inactive to reflect the active sets.
Failure Paths: - unable to transfer the CM REX data to the inactive
               - unable to transfer the cleanup data to the inactive
               - There are no real failure paths on the inactive, but swerrs
                 may be generated when the wrong action or data versions are
                 received.

Step Name : VERIFY_PROCESSOR_OPTIONALITY
----------------------------------------
Implementation: CMPSWACT
Procedure Name: PRESWACT_PROC_OPT_VERIFICATION
Procedure Type: PRESWACT_INACTIVE_PROC
Description   : Step verifies the processor optionality office parameter on
                the inactive CPU.  This involves reading the value of the
                CM_PROCESSOR_OPTION tuple of table OFCOPT, and verifying
                that the CPU and memory hardware actually equipped on that
                plane is compatible with the datafill.
Failure Paths  : - Equipped hardware is incompatible with datafilled value.

Step Name: CMIC_LINKHITS_CHECK
------------------------------
Implementation: CMWSWCT
Procedure Name: CMIC_LINKHITS_CHECK
Procedure Type: ACTIVE_PROC
Description: Check the integrity of the cmic links on the active side.
Failure Paths: This step fails if any mc has more that three link hits in the
               past twenty-four hours.


Step Name: DISABLE_AUTOIMAGE
----------------------------
Implementation: ADUMPCTL
Procedure Name: DISABLE_AUTOIMAGE
Procedure Type: ACTIVE_AND_INACTIVE_PROC
Description: This procedure is used to disable the auto image dump process.
             It stops a currently executing image dump as well as any
             scheduled image dumps.  It will be re-enabled later by step
             ENABLE_AUTOIMAGE.
Failure Paths: - unable to allocate mta resources
               - unable to search for the dump controller
               - unable to send the message to the dump process node
               - unable to stop the dump process
               - the wrong message type is received
               - when the receive message is not ok or timeout

Step Name: SAVE_DSLIMIT
-----------------------
Implementation: APPLUTIL
Procedure Name: SAVE_DSLIMIT
Procedure Type: ACTIVE
Description: This procedure saves the current dsmax value in table dslimit
             to 'STOREFS' file on sfdev. This file is sent to the mate side
             and will this value will be restored during POSTSWACT by step
             'CLEAN_UP_SFDEV'.
Failure Paths: - unable to create 'STOREFS' file.
               - unable to position on the STOREFS tuple in table dslimit.
               - unable to write to file 'STOREFS'.
               - unable to sent file 'STOREFS' to mate side.

Step Name: VERIFY_DSLIMIT
-------------------------
```

```
Implementation: APPLUTIL
Procedure Name: VERIFY_DSLIMIT
Procedure Type: ACTIVE_AND_INACTIVE_PROC
Description: Will ensure that there is at least 100k of free sfdev on both
             active and inactive processors. If less than 100k, then 100k
             will be added to the current dsmax value in table dslimit.
Failure Paths: - unable to access the storefs tuple
               - if more than 96% of the total data store is unavailable
               - unable to update field dsmax


Step Name: TRACE_ON
-------------------
Implementation: APPLUTIL
Procedure Name: TRACE_ON
Procedure Type: ACTIVE_PROC
Description: This procedure attempts to open a trace file on the specified
             device.  Tracing will be stopped later by steps TRACE_OFF and
             MATE_TRACE_OFF.
Failure Paths: This procedure will fail if information of the specified
               device cannot be found.

Step Name: HALT_ACTIVE_CTT
--------------------------
Implementation: CTTAPPL
Procedure Name: CTT_HALT
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This procedure attempts to halt CTT (Cell-site Terminal Testing)
             on the active side. It will be restarted by step RESUME_CTT.
Failure Paths: None

Step Name: HALT_CTT
-------------------
Implementation: CTTAPPL
Procedure Name: CTT_HALT
Procedure Type: PRESWACT_INACTIVE_PROC
Description: This procedure attempts to hal CTT (Cell-site Terminal Testing)
             on the inactive side. It will be restarted by step RESUME_CTT.
Failure Paths: None

Step Name: DISABLE_PATCH_AUDIT_ACT
----------------------------------
Implementation: PALARMIM
Procedure Name: DISABLE_PATCH_AUDIT
Procedure Type: ACTIVE_PROC
Description: This procedure will disable the patch audit on the active.
             See related steps DISABLE_PATCH_AUDIT_INACT and
             ENABLE_PATCH_AUDIT_POSTSWACT.
Failure Paths: unable to perform a cond_unprotectds


Step Name: SET_OFFICE_TUPLES
----------------------------
Implementation: APPLUTIL
Procedure Name: SET_OFFICE_TUPLES
Procedure Type: ACTIVE_PROC
Description: This procedure will save the current state of office tuples
             noderexcontrol, lcdrex_control and
             guaranteed_terminal_cpu_share. Then it will attempt to set
             office tuple noderexcontrol to 'N  9 30  9 31', lcdrex_control
             to 'N  1 0   3 0  2' and guaranteed_terminal_cpu_share to the
             maximum value.
             See related steps SET_MATE_TUPLES and RESET_OFFICE_TUPLES.
Failure Paths: - unable to create rex$file
               - unable to find tuple noderexcontrol or lcdrex_control in table
                 ofcvar or guaranteed_terminal_cpu_share in table ofceng
               - if procedure cond_unprotectds fails for office tuple
                 noderexcontrol or lcdrex_control
               - unable to write any current office tuple value to rex$file
               - unable to update tuple noderexcontrol to 'N  9 30  9 31',
                 lcdrex_control to 'N  1 0   3 0  2' or
                 guaranteed_terminal_cpu_share to the maximum value
               - unable to send rex$file to the inactive


Step Name: SET_MATE_TUPLES
--------------------------
Implementation: APPLUTIL
Procedure Name: SET_MATE_OFFICE_TUPLES
Procedure Type: INACTIVE_PROC
Description: This procedure will attempt to find rex$file on the inactive
             side and write the contents into protected store. If the file is
             not found, then save the values of the following office tuples:
             noderexcontrol, lcdrex_control and
             guaranteed_terminal_cpu_share. Then it will attempt to set
             office tuples dump_restore_in_progress to y, noderexcontrol to
             'N  9 30  9 31', lcdrex_control to 'N  1 0   3 0  2' and
             guaranteed_terminal_cpu_share to the maximum value.
             See related steps SET_OFFICE_TUPLES and RESET_OFFICE_TUPLES.
Failure Paths: The following paths will occur if the rex$file is found on the
               inactive side:
                 - unable to open rex$file
                 - unable to read records from rex$file
                 - if the guaranteed_terminal_cpu_share value contained within
```

```
                    rex$file is not a valid gterm value
                  - if procedure cond_unprotectds fails
                  - if anything other than the three office tuples are found
                    within rex$file
                The following paths will occur if the rex$file is not found:
                  - unable to find tuple noderexcontrol or lcdrex_control in
                    table ofcvar or guaranteed_terminal_cpu_share in table
                    ofceng
                  - if procedure cond_unprotectds fails for office tuple
                    noderexcontrol or lcdrex_control
                  - unable to update tuple noderexcontrol to 'N  9 30  9 31',
                    lcdrex_control to 'N  1 0   3 0  2' or
                    guaranteed_terminal_cpu_share to the maximum value


Step Name: DTDETECT_TRANSFER
----------------------------
Implementation: APPLDIGT
Procedure Name: SEND_DTDETECT
Procedure Type: ACTIVE_PROC
Description: This procedure sends the DTDETECT status from the active to the
             inactive.  DTDETECT is DigiTone DETECTion.  Step RESTORE_DTDETECT
             restores this value on the new load.  The retrofit version of
             this step is implemented by steps WRITE_DTDETECT and READ_DTDETECT.
Failure Paths: - unable to reset dtdetect_in_use on the active
               - unable to reset dtdetect_in_use on the inactive

               The following failure paths apply only if dtdetect is on
               - unable to set dtdetect_in_use on the active
               - unable to set dtdetect_in_use on the inactive


Step Name: WRITE_DTDETECT
-------------------------
Implementation: SWITCH1
Procedure Name: WRITE_DTDETECT
Procedure Type: Retrofit Dump Side
Description: This procedure will write the digitonedet status to tape file
             detect$data. A 'Y' is written if dtdetect is on or a 'N' is
             written during NT40 to SuperNode conversions.  Step READ_DTDETECT
             restores this value on the new load.  Steps DTDETECT_TRANSFER
             and RESTORE_DTDETECT implement this functionality for standard
             ONPs.
Failure Paths: unable to write the record to the detect$data file
```

```
Step Name: READ_DTDETECT
------------------------
Implementation: RETUTIL
Procedure Name: READ_DTDETECT
Procedure Type: Retrofit Restore Side
Description: This procedure reads the detect$data file and sets
             dtdetect_in_use if a 'Y' or resets it if a 'N' for nt40 to
             SuperNode conversions.  Step WRITE_DTDETECT created this file.
             Steps DTDETECT_TRANSFER and RESTORE_DTDETECT implement this
             functionality for standard ONPs.
Failure Paths: unable to read the record from detect$data


Step Name: SEND_PATCHES
-----------------------
Implementation: APPLUTIL
Procedure Name: SEND_PATCHES
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This procedure will message the inactive to build and send a
             file containing a list of applied patches, find the processor
             type of the inactive, build a patch list containing the
             appropriate patches to be applied on the inactive, and send all
             the appropriate patches to the inactive.
             This step is needed for step APPLY_PATCHES.
Failure Paths: - unable to build the applied patch/prsu or the suffix files on
                 the inactive
               - unable to receive the applied patch/prsu or the suffix files
                 from the inactive
               - unable to build the release and suffix file
               - unable to retrieve patches/prsus from the file
               - unable to retrieve the inactive processor type
               - unable to retrieve the fids for the patches/prsus to send to
                 the inactive
               - unable to send the patches/prsus to the inactive


Step Name: APPLY_PATCHES
------------------------
Implementation: APPLUTIL
Procedure Name: APPLY_PATCHES
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This procedure sends a message to the inactive starting the
             patch/prsm process which sends a message back to the active when
             the patch/prsm process completes.  This step is executed after
             SEND_PATCHES.
Failure Paths: PRSM only:
               - unable to create the suffix file when non-existent in the
```

```
                 inactive sfdev
               - unable to get the release and suffix information
               - unable to get the prsus to apply


Step Name: MATE_RESTART_COLD
----------------------------
Implementation: APPLUTIL
Procedure Name: MATE_RESTART_COLD
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This procedure sends a message to the inactive initiating a cold
             restart. The user may stop this process if any users are logged
             on the inactive. The inactive sends back a message to the active
             indicating if successful or not.
Failure Paths: - unable to query the inactive for logged on users
               - unable to message the inactive with the restart cold message
               - unable to claim the matecom
               - unable to complete the restart cold
               - unable to send a message to the inactive as a check


Step Name: MATE_RESTART_WARM
----------------------------
Implementation: APPLUTIL
Procedure Name: MATE_RESTART_WARM
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This procedure sends a message to the inactive initiating a warm
             restart. The user may stop this process if any users are logged
             on the inactive. The inactive sends back a message to the active
             indicating if it was successful or not.
Failure Paths: - unable to query the inactive for users logged on
               - unable to message the inactive with the restart warm message
               - unable to claim the matecom
               - unable to complete the restart warm
               - unable to send a message to the inactive as a check


Step Name: VERIFY_DUMP_RESTORE
------------------------------
Implementation: APPLUTIL
Procedure Name: VERIFY_DUMP_RESTORE
Procedure Type: PRESWACT_INACTIVE_PROC
Description: This procedure verifies that all tables have completed (not in
             the in_progress or not_started states) before the activity switch
             can occur.  If all tables are not in the completed state, a file
             containing the table states is generated and user is informed to
             print the tabstates file on the inactive.
```

```
Failure Paths: - unable to create the tabstates file
               - unable to write a record to the tabstates file
               - unable to close the file


Step Name: SWACT_MODULE_CHECK
-----------------------------
Implementation: MCKAC
Procedure Name: MODCHK_DRIVER
Procedure Type: ACTIVE_PROC
Description: This procedure will report any missing application modules that
             are necessary to perform the activity switch to the user. If any
             application modules are missing, the user will be asked if the
             process should continue.
Failure Paths: - if the facility is already in use
               - unable to allocate the override table


Step Name: FRAME_RELAY_BILLING_GENERATION
-----------------------------------------
Implementation: FRBSWACT
Procedure Name: FR_RELAY_BILLING_BCSUPDATE
Procedure Type: ACTIVE_PROC
Description: This procedure creates a process that will aggregate all of the
             frame relay billing data into ama records and ship them off to
             be formatted and to close off billing before the activity switch
             occurs.
Failure Paths: - if the aspect is nil
               - unable to allocate a mailbox
               - a timeout occurs, the mailbox is not ok, wrong message type is
                 received or the message result is not create_ok during
                 request_child_creation


Step Name: DISABLE_PATCH_AUDIT_INACT
------------------------------------
Implementation: PALARMIM
Procedure Name: DISABLE_PATCH_AUDIT
Procedure Type: INACTIVE_PROC
Description: This procedure will disable the patch audit on the inactive.
             The audit will be re-enabled by step ENABLE_PATCH_AUDIT_POSTSWACT.
Failure Paths: unable to save the current patch audit state and reset the
               state (cond_unprotectds)


Step Name: HALT_ACTIVE_ALT
--------------------------
```

```
Implementation: ALTTCUTL
Procedure Name: HALT_ACTIVE_ALT
Procedure Type: ACTIVE_PROC
Description: This procedure attempts to halt ALT (Automatic Line Test) on the
             active.
Failure Paths: Unable to add the times to get the override times


Step Name: HALT_ACTIVE_ATT
--------------------------
Implementation: APPLATT
Procedure Name: ATT_HALTATT
Procedure Type: ACTIVE_PROC
Description: This procedure attempts to halt ATT (Automatic Trunk Test) on the
             active.  It will be restarted by step RESUME_ATT.
Failure Paths: None


Step Name: HALT_ATT
-------------------
Implementation: APPLATT
Procedure Name: ATT_HALTATT
Procedure Type: INACTIVE_PROC
Description: This procedure attempts to halt ATT (Automatic Trunk Test) on the
             inactive.  It will be restarted by step RESUME_ATT.
Failure Paths: None

Step Name: CSC_LINK_CHECK
-------------------------
Implementation: CSCCHECK
Procedure Name: CSC_CHECK
Procedure Type: ACTIVE_PROC
Description: This procedure checks that the CSCs (Cell Site Controllers) are
             not in a pooled state (must be in the dedicated state) prior to
             the activity switch.
Failure Paths: A csc is in the pooled state


Step Name: STATUSUPDATE
-----------------------
Implementation: STAUPDUI
Procedure Name: STATUSUPDATE_PROC
Procedure Type: ACTIVE_PROC
Description: This procedure first verifies that all the hardware on the
             active is ok (insv, istb or offline). Then these node states are
             transferred and updated on the inactive side.  In-service devices
             will be set man-busy on the inactive side, and the next restart
             will attempt to put them in-service.  Steps WRITE_DEVICE_STATUS
             and READ_DEVICE_STATUS perform for retrofits what this step
             performs for standard ONPs.
Failure Paths: - prechecks (insync, on inactive, or swact in progress) fail
               - no communication with the mate
               - modcheck command has not been run
               - the active device check fails
               - the mextract procedure fails


Step Name: WRITE_DEVICE_STATUS
------------------------------
Implementation: RETSTAT
Procedure Name: WRITE_DEVICE_STATUS
Procedure Type: Retrofit Dump Side
Description: This procedure first verifies that all the hardware (nodes) on
             the active is ok (insv, istb or offline). Then these node states
             are written to the tape file device$stat.  Step READ_DEVICE_STATUS
             uses this file to restore the device states.
Failure Paths: - the device check on the active fails
               - unable to create the device$stat file
               - unable to translate the device status index type
               - unable to write to the device$stat file


Step Name: READ_DEVICE_STATUS
-----------------------------
Implementation: RETSTAT
Procedure Name: READ_DEVICE_STATUS
Procedure Type: Retrofit Restore Side
Description: This step opens and reads from file DEVICE$STAT created by step
             WRITE_DEVICE_STATUS.  It loops over each record in the file,
             looking for a matching device.  If a matching device is found,
             and it is in an offline state, the process attempts to set the
             device to a man-busy state.  The upcoming restart will attempt
             to set these devices in-service.
             WRITE_DEVICE_STATUS and READ_DEVICE_STATUS perform for
             retrofits what STATUSUPDATE performs for standard ONPs.
Failure Paths: Failure paths are abundant.  The majority deal with file errors.
               Other failures occur if the file is not found, if some devices
               are not found, or if some devices are not set man-busy.


Step Name: IXPM_STATUSUPDATE
----------------------------
Implementation: INTSWCT1
Procedure Name: IXPMSTATUSUPDATE_PROC
Procedure Type: ACTIVE_PROC
Description: This is a special statusupdate procedure for international
             offices. First it verifies that all IXPMs on the active is ok
             (insv, istb or offl). Then these node states are transferred and
             updated on the inactive side.  In-service devices will be set
             man-busy on the inactive side, and the next restart will attempt
             to put them in-service.
Failure Paths: Failure occurs when the mextract procedure fails.


Step Name: FOCUSED_MAINT_XFER
-----------------------------
Implementation: APPLFM
Procedure Name: FOCUSED_MAINT_XFER
Procedure Type: ACTIVE_PROC
Description: This step extracts all troubles suppressed and resumed in the
             LNSTRBL and TRKTRBL MAPCI levels found under the MTC;LNS level
             and MTC;TRKS level respectively. The troubles are transferred
             to the inactive side via PRESWACT_TRANSFER.  This step will
             ensure all troubles seen with lines and trunks on the active
             side are transferred to the inactive side.  The retrofit
             version of this step is comprised of steps FM_DUMP and FM_RESTORE.
Failure Paths: This step fails if the transfer of the trunks trouble data
               failed, or if the transfer of the lines trouble data failed.


Step Name: FM_DUMP
------------------
Implementation: RETROFM
Procedure Name: FOCUSED_MAINT_DUMP
Procedure Type: Retrofit Dump Side
Description: This step is executed during retrofits.  It writes the
             lines and trunks trouble information discussed in step
             FOCUSED_MAINT_XFER to files LNTRBL$DATA and TKTRBL$DATA
             respectively.  This step and step FM_RESTORE compose the
             retrofit version of step FOCUSED_MAINT_XFER.
Failure Paths: Step fails if creation of either file fails.


Step Name: FM_RESTORE
---------------------
Implementation: RETROFM
Procedure Name: FOCUSED_MAINT_REST
Procedure Type: Retrofit Restore Side
Description: This step is executed during retrofits.  It reads the
             lines and trunks trouble information discussed in step
             FOCUSED_MAINT_XFER from files LNTRBL$DATA and TKTRBL$DATA
             respectively.  The data is written into the protected memory
             for this information.  Step FM_DUMP and this step compose the
             retrofit version of step FOCUSED_MAINT_XFER.
Failure Paths: Steps fails if either file cannot be read, or if unable
               to unprotect memory to store data.


Step Name: DUMP_TOPSMP_STATES
-----------------------------
Implementation: YMPMTCH1
Procedure Name: DUMP_TOPSMP_STATES
Procedure Type: ACTIVE_PROC
Description: The first action taken by this step is to ensure no TOPS positions
             are in the training state.  If any are, they are displayed and the
             step is halted.  If none are in the training state, the step
             continues on to create file TOPSMP$INB containing a list of all
             TOPS positions in the INB state, and file TOPSMP$MB containing a
             list of all TOPS positions in the MB state.  These files are then
             sent to the inactive if this is a standard ONP.
             INB positions are restored by step RESTORE_INB_TOPSMP.
             MB positions are restore by step RESTORE_MB_TOPSMP.
Failure Paths: This step fails if there are TOPS positions in the training
               state, if file creation fails, or if file transfer fails.


Step Name: RESTORE_INB_TOPSMP
-----------------------------
Implementation: YMPMTCH1
Procedure Name: RESTORE_INB_TOPSMP
Procedure Type: INACTIVE_PROC
Description: This step first RTSs all TOPS positions and devices.  It then
             sets all positions listed in files TOPSMP$INB and TOPSMP$MB to
             an INB state.  Positions that were originally MB are set to INB
             because a restart would set these positions idle.  They will be
             set to MB during POSTSWACT.  These files were created by step
             DUMP_TOPSMP_STATES.
Failure Paths: Fails if the step cannot set all positions listed in the file
               to an INB state.


Step Name: CORRECT_DRWR_STATES
------------------------------
Implementation: DRWRMTIM
Procedure Name: CORRECT_DRWR_STATES
Procedure Type: ACTIVE_PROC
Description: This procedure messages the inactive side to execute the
             drawer init procedures and then calls the dump_offl_drwrs
             to display and correct the drawer states.
Failure Paths: This procedure will return false if any of the following
               occur:
               - line drawer found in an invalid state (not OK or OFFL)
               - unable to allocate store
               - unable to message the inactive to initialize drawer states
               - unable to dump drawer states


Step Name: RTS_ALL_DRWRS
-----------------------
Implementation: DRWRMTIM
Procedure Name: INIT_DRWR_STATES
Procedure Type: Retrofit Restore Side
Description: This procedure runs on the inactive side only during
             BCSUPDATE PRESWACT. It runs on the active side of the
             SuperNode during the RETROFIT PRESWACT. It is bound
             into the RETROFIT gate by the nonres module RTSDRWRS.
Failure Paths: None


Step Name: TRANSFER_TRUNK_STATES
-------------------------------
Implementation: TRKMTCH1
Procedure Name: DUMP_TRUNKS
Procedure Type: ACTIVE_PROC
Description: This procedure dumps the trunk states on the active side to
             files: inb$trks, inb$trks2, mb$trks, mb$trks2, res$trks and
             res$trks2. Then the files are transferred to the inactive side.
             These files will be used in steps RESTORE_INB and TRUNK_RESTORE.
Failure Paths: - unable to adjust dslimit to 100k free
               - unable to dump trunk state to the appropriate file
               - unable to send the files to inactive for onp or copy the files
                 to the preswact device for a conversion


Step Name: DUMP_TRUNK_STATES
---------------------------
Implementation: TRKMTCH1
Procedure Name: DUMP_TRUNKS
Procedure Type: Retrofit Dump Side
Description: This procedure dumps the trunk states on the active side to
             files: inb$trks, inb$trks2, mb$trks, mb$trks2, res$trks and
             res$trks2. Then the files are copied to the preswact device.
             This step and RESTORE_TRUNKS are the retrofit equivalent of
             steps TRANSFER_TRUNK_STATES, RESTORE_INB, and TRUNK_RESTORE.
Failure Paths: - unable to adjust dslimit to 100k free
```

```
               - unable to dump trunk state to the appropriate file
               - unable to send the files to inactive for onp or copy the files
                 to the preswact device for a conversion


Step Name: RESTORE_INB
----------------------
Implementation: TRKMTCH1
Procedure Name: RESTORE_INB
Procedure Type: INACTIVE_PROC
Description: This procedure restores the trunks contained within the inb$trks
             inb$trks2 files on the inactive.  This is accomplished by first
             setting all INB trunks to MB, then setting trunks listed in the
             files back to INB.  The files were created by step
             TRANSFER_TRUNK_STATES.
Failure Paths: unable to restore the trunk state to cnd_offline


Step Name: RESTORE_TRUNKS
-------------------------
Implementation: RETTRKS
Procedure Name: RETROFIT_RESTORE_TRUNKS
Procedure Type: Retrofit Restore Side
Description: This procedure restores the trunks contained within the inb$trks
             and inb$trks2 files and copies the mb$trks, mb$trks2, res$trks
             and res$trks2 files to sfdev.  Setting of the trunks to INB
             status is accomplished by first setting all INB trunks to MB,
             then setting trunks listed in the files back to INB.
             DUMP_TRUNK_STATES and this step are the retrofit equivalent of
             steps TRANSFER_TRUNK_STATES, RESTORE_INB, and TRUNK_RESTORE.
Failure Paths: - unable to restore the trunks contained within the inb$trks and
                 inb$trks2 to the cnd_offline state
               - unable to copy mb$trks, mb$trks2, res$trks and res$trks2 files
                 to sfdev


Step Name: TRANSFER_STABLE_PRI_DCHS
-----------------------------------
Implementation: PRIDONP1
Procedure Name: DUMP_STABLE_PRI_DCHS
Procedure Type: ACTIVE_PROC
Description: Step writes all installation busy (INB) and man busy (MB)
             D-Channel Handler (DCH - ISDN only) type peripherals to files
             INB$DCH and MB$DCH on SFDEV.  If a retrofit is being performed,
             these files are written to tape.  Otherwise the files are sent
             to the inactive side.  The files will be used by step
             RESTORE_STABLE_PRI_DCHS.
```

```
Failure Paths: - Not enough room in SFDEV to write files.
               - Failed to write DCHs in INB state to file.
               - Failed to write DCHs in MS state to file.
               - Failed to send INB$DCH or MB$DCH files to SFDEV if not
                 a retrofit.
               - Failed to copy INB$DCH or MB$DCH files to tape during
                 a retrofit.


Step Name: RESTORE_STABLE_PRI_DCHS
----------------------------------
Implementation: PRIDONP1
Procedure Name: RESTORE_PRI_DCHS
Procedure Type: INACTIVE_PROC
Description: Step restores D-Channel Handlers (DCH) to the states indicated
             in INB$DCH and MB$DCH files. These files were created by step
             TRANSFER_STABLE_PRI_DCHS.
Failure Paths: - Failed to open file MB$DCH on SFDEV.
               - Failed to restore DCHs in state MB per MB$DCH file.
               - Failed to open file INB$DCH on SFDEV.
               - Failed to restore DCHs in state INB per INB$DCH file.


Step Name: TRANSFER_PDTC_HG
---------------------------
Implementation: HGPDTCIM
Procedure Name: PDTC_HGSTAT_DUMP
Procedure Type: ACTIVE_PROC
Description: This procedure will dump the Handler Group (HG) information
             for PDTCs to file PDTC$HGSTATE. The file is created for
             use by step RESTORE_PDTC_HG.
Failure Paths: The step will fail if any of the following fails to occur:
              - create file PDTC$HGSTATE
              - write HG information to the file
              - close file PDTC$HGSTATE


Step Name: RESTORE_PDTC_HG
--------------------------
Implementation: HGPDTCIM
Procedure Name: PDTC_HGSTAT_RESTORE
Procedure Type: INACTIVE_PROC
Description: This procedure will retrieve file PDTC$HGSTATE from the active
             side and restore the Handler Group (HG) information into
             the new load.  The file was created by step TRANSFER_PDTC_HG.
Failure Paths: The step will fail if any of the following fails to occur:
              - matecopy file PDTC$HGSTATE
              - find file PDTC$HGSTATE, after it has been matecopied
              - open file PDTC$HGSTATE
              - read each record from PDTC$HGSTATE
              - close file PDTC$HGSTATE


Step Name: DUMP_LINE_STATES
---------------------------
Implementation: LNMTCH1
Procedure Name: DUMP_LINE_STATES
Procedure Type: ACTIVE_PROC
Description: This procedure dumps the line states on the active side to
             files: inb$lns, inb$lns2, cut$lns, cut$lns2, mb$lns, mb$lns2,
             haz$lns and haz$lns2. Then the files are copied to tape for
             retrofits, or transferred to the inactive side for standard ONPs.
             These files will be used by steps RESTORE_LINE_STATES and
             RESTORE_MB_LINES.
Failure Paths: - unable to allocate buffers
               - unable to dump the line state to the appropriate file
               - unable to copy the line file to tape for a conversion
               - unable to send the files to the inactive for an ONP


Step Name: SYSTEM_DATE_XFR
--------------------------
Implementation: INTSWCT1
Procedure Name: SYSTEM_DATE_XFR
Procedure Type: ACTIVE_PROC
Description: Transfer the system date to the inactive.
Failure Paths: Failure occurs when the transfer fails.


Step Name: RESTORE_LINE_STATES
------------------------------
Implementation: LNMTCH1
Procedure Name: RESTORE_LINE_STATES
Procedure Type: INACTIVE_PROC
Description: This procedure attempts to restore the lines states contained
             within the files: inb$lns, inb$lns2, cut$lns, cut$lns2, mb$lns,
             mb$lns2, haz$lns and haz$lns2. Since mb lines are set to idl
             during a restart, the mb lines are first set to the inb state
             and then corrected to mb following the activity switch by step
             RESTORE_MB_LINES.  These files were created by step
             DUMP_LINE_STATES.
Failure Paths: - unable to allocate buffers
               - unable to restore the lines contained within the inb$lns or
                 inb$lns2 files to line_offline
               - unable to restore the lines contained within the cut$lns or
                 cut$lns2 files to line_cutoff
               - unable to restore the lines contained within the haz$lns or
                 haz$lns2 files to line_hazard
               - unable to restore the lines contained within the mb$lns or
                 mb$lns2 files to line_offline


Step Name: ATTCONS_MATCH
------------------------
Implementation: APPLIBN
Procedure Name: ATTCONS_MATCH
Procedure Type: ACTIVE_PROC
Description: This step ensures that field INSV of table ATTCONS matches
             the current status of all attendant consoles. This routine
             loops over all possible attendant consoles, and if the field
             INSV of table ATTCONS for that attendant console does not
             match the status of the actual attendant console, the field
             is updated.
Failure Paths: This step fails if the INSV field is not updated correctly.


Step Name: OVERLAP_CHECK
------------------------
Implementation: APPLOCC
Procedure Name: OVERLAP_CHECK
Procedure Type: INACTIVE_PROC
Description: Checks to see if the market of this office is North America.
             If it is, it checks to see if this is an equal access office.
             If it is, the global OVERLAP_BOOL is set to false.
Failure Paths: None.


Step Name: SEND_TOPS_DB
-----------------------
Implementation: YMEMEDT1
Procedure Name: SEND_TOPS_DB_PROC
Procedure Type: ACTIVE_PROC
Description: This steps deactivates the TOPS booked call database and transfers
             the database from the active to the inactive CPU.
Failure Paths: Step fails if database is currently being transferred by another
               user, if the database does not exist, the database is locked or
               in use and cannot be transferred, or if there is a failure with
               the transfer itself.
```

```
Step Name: TABLE_DELTA
-----------------------
Implementation: DELTLST1
Procedure Name: DELTA_TABLES
Procedure Type: PRESWACT_ACTIVE_PROC
Description: Loop over two hard-coded lists of tables, performing a delta on
             each table.  This compares the datafill of the table on the active
             side with the datafill of the table on the inactive side.  Tuple
             differences are displayed.  Tables will not be deltaed if they
             have been successfully transferred by the ONP process within the
             last 24 hours.  This avoids deltaing unnecessary tables, and
             provides significant time savings during the ONP.
Failure Paths: Step fails if an individual table delta shows differences, or
               if the delta process itself fails for some reason.


Step Name: DUMP_DELTA_TABS
--------------------------
Implementation: RETDUMP
Procedure Name: DUMP_DELTA_TABS
Procedure Type: Retrofit Dump Side
Description: This step is executed during retrofits.  It loops over two hard-
             coded lists of tables, dumping the table data to files.  These
             files are used during step DELTA_TABLES to compare table datafill
             from the old load to table datafill on the new load.  This step
             and step DELTA_TABLES compose the retrofit version of TABLE_DELTA.
Failure Paths: Fails if the dumping of a table fails.


Step Name: DELTA_TABLES
-----------------------
Implementation: RETDELT
Procedure Name: DELTA_TABLES
Procedure Type: Retrofit Restore Side
Description: This step is executed during retrofits. It loops over two hard-
             coded lists of tables, and calls a procedure to compare table
             datafill to table datafill in files created by step
             DUMP_DELTA_TABS.  DUMP_DELTA_TABS and this step compose the
             retrofit version of TABLE_DELTA.
Failure Paths: Fails if the delta of the table data versus file data fails
               for some reason.


Step Name: SEND_ITOPS_DB
------------------------
Implementation: IYMEMED1
Procedure Name: SEND_ITOPS_DB_PROC
Procedure Type: ACTIVE_PROC
Description: This steps deactivates the international TOPS booked call database
             and transfers the database from the active to the inactive CPU.
Failure Paths: Step fails if database is currently being transferred by another
               user, if the database does not exist, the database is locked or
               in use and cannot be transferred, or if there is a failure with
               the transfer itself.


Step Name: SET_SWCT_AMA
-----------------------
Implementation: SWCTAMA
Procedure Name: SET_SWCTAMA
Procedure Type: ACTIVE_PROC
Description: Step enables premature billing during a warm swact.
Failure Paths: Failed to conditional unprotect data store.


Step Name: RESET_METERS
-----------------------
Implementation: GLOBSWT1
Procedure Name: RESETMETERS_PROC
Procedure Type: INACTIVE_PROC
Description: Step resets the line meters and trunk meters to zero on
             the inactive side to ensure that these meters are re-
             allocated on the next reload restart. This step is only
             available in international offices.
Failure Paths: Not Available.


Step Name: MATE_RESTART_RELOAD
------------------------------
Implementation: APPLUTIL
Procedure Name: MATE_RESTART_RELOAD
Procedure Type: ACTIVE_PROC
Description: This procedure sends a message to the inactive initiating a
             reload restart. The user may stop this process if any users are
             logged on the inactive. The inactive sends back a message to the
             active indicating if it was successful or not.
Failure Paths: - unable to query the inactive for users logged on
               - unable to message the inactive with the restart reload message
               - unable to claim the matecom
               - unable to complete the restart reload
               - unable to send a message to the inactive as a check
```

```
Step Name: CM_RESTART
---------------------
Implementation: RETUTIL
Procedure Name: CM_RESTART
Procedure Type: Retrofit Restore Side
Description: Performs a reload restart on the active side. This step is
             executed during retrofits only.
Failure Paths: Not Available.


Step Name: DISABLE_DCT
----------------------
Implementation: DCTPRENT
Procedure Name: DCT_PROCESS_DEACTIVATION
Procedure Type: PRESWACT_ACTIVE_PROC
Description: Step disables all Data Call Tester (DCT) tests currently
             active.  It will be reactivated by step REACTIVATE_DCT.
Failure Paths: Not Available.


Step Name: STATUSCHECK
----------------------
Implementation: STATCHK1
Procedure Name: STATUSCHECK_PROC
Procedure Type: ACTIVE_PROC
Description: Performs a status hardware check on the active side to
             ensure device are one of the following states: OK,
             OFFLINE, or UNEQUIPPED. Step also compares hardware
             status of the two sides.
Failure Paths: - Fails if the any of the following conditions exist:
                 site is in sync, step executing on inactive side, or
                 CC warm SWACT in progress.
               - Fails if unable to communicate with the mate side.
               - Fails if MODCHECK tables do not match Device Status
                 application tables and procedure was not called by
                 PRESWACT.
               - Fails if MODCHECK tables do not match Device Status
                 application tables and PRESWACT is in use.
               - Fails if the status of any devices on active side are
                 not OK, OFFLINE, or UNEQUIPPED.
               - Fails if status of hardware mismatches are found between
                 the two sides.


Step Name: OFFL_DEVICE_CHECK
----------------------------
Implementation: RETSTAT
```

```
Procedure Name: CHECK_DEVICE_STATUS
Procedure Type: Retrofit Restore Side
Description: This step opens and reads from file DEVICE$STAT created by step
             WRITE_DEVICE_STATUS.  It loops over each record in the file,
             looking for a matching device.  If a matching device is found,
             and it is in an offline state, the device description is displayed
             and the step fails.  This is the retrofit equivalent of standard
             ONP step STATUSCHECK.
Failure Paths: Failure paths are abundant.  The majority deal with file errors.
               Other failures occur if the file is not found, if some devices
               are not found, or if some devices are offline.


Step Name: VERIFY_STORE
-----------------------
Implementation: APPLUTIL
Procedure Name: VERIFY_STORE
Procedure Type: INACTIVE_PROC
Description: Ensures that no more than 96 percent of data store is used in
             the inactive load.
Failure Paths: When more than 96 percent of total data store (DS) is
              is used, an error will be displayed to the user indicating
              the percentage used.


Step Name: PRELOAD_EXECS
------------------------
Implementation: PRLDIMP
Procedure Name: PRE_LOAD_EXECS
Procedure Type: ACTIVE_PROC
Description: This is an active side procedure which will loop over the table
             of required execs and perform the following steps:
             - Message the mate to build the current exec lineup.
             - Message the mate to transfer the exec lineup.
             - Preload the lineup to all XPMs that require it.
Failure Paths: Messages user and generates a SWCT log when it's unable to
               determine the version of the software load on the inactive side.
               Return false if one of the following occurs:
                 - unable to get mate execs.
                 - unable to build exec lineup.
                 - unable to send the execs to the PMs.


Step Name: UNMASK_CUSTFLDS
--------------------------
Implementation: APPLMASK
Procedure Name: UNMASK_CUSTFLDS
Procedure Type: ACTIVE_PROC
Description: This step will unmask the following fields from tables
             CFW and CFX by deleting the field entries from table CUSTFLDS:
             - CFW fields 1, 2 & 3
             - CFX fields 1 & 2
             The fields will be re-masked by step MASK_CUSTFLDS.
Failure Paths: An error message will be displayed to the user and a swer
              will be generated if the procedure is unable to delete the
              entries from table CUSTFLDS.
Note: This step should be removed.


Step Name: REACTIVATE_DCT
-------------------------
Implementation: DCTPRENT
Procedure Name: DCT_PROCESS_ACTIVATION
Procedure Type: ABORT_PROC
Description: This step will reactivate the Data Call Tester (DCT) tool.
             The DCT tools was deactivated by step DISABLE_DCT earlier
             in the PRESWACT process.
Failure Paths: The procedure will message the user and return false if the
               DCT tool cannot be reactivated.


Step Name: MATE_UNMASK_CUSTFLDS
-------------------------------
Implementation: APPLMASK
Procedure Name: UNMASK_CUSTFLDS
Procedure Type: INACTIVE_PROC
Description: This step will unmask the following fields from tables
             CFW and CFX by deleting the field entries from table CUSTFLDS:
             - CFW fields 1, 2 & 3
             - CFX fields 1 & 2
             This is the same as step UNMASK_CUSTFLDS except it is performed
             on the inactive side.  The fields will be re-masked by step
             MATE_MASK_CUSTFLDS.
Failure Paths: An error message will be displayed to the user and a swer
               will be generated if the procedure is unable to delete the
               entries from table CUSTFLDS.
Note: This step should be removed.


Step Name: TRANSFER_CFW
-----------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_CFW
Procedure Type: ACTIVE_PROC
```

```
Description: This procedure will dump and restore table CFW from the active
             side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_CFW
--------------------
Implementation: RETDTAB
Procedure Name: WRITE_CFW
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table CFW to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step 'RESTORE_CFW'
             for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_CFW
----------------------
Implementation: RETDTAB
Procedure Name: RESTORE_CFW
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table CFW from file CFW$,
             which was created by procedure WRITE_CFW.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the CFW$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process CFW$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_CFX
-----------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_CFX
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table CFX from the active
             side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.
```

```
Step Name: WRITE_CFX
--------------------
Implementation: RETDTAB
Procedure Name: WRITE_CFX
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table CFX to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step 'RESTORE_CFX'
             for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_CFX
----------------------
Implementation: RETDTAB
Procedure Name: RESTORE_CFX
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table CFX from file CFX$,
             which was created by procedure WRITE_CFX.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the CFX$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process CFX$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_SCALLTAB
----------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_SCALLTAB
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table SCALLTAB from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_SCALLTAB
-------------------------
Implementation: RETDTAB
Procedure Name: WRITE_SCALLTAB
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
```

```
             procedure dumps the datafill in table SCALLTAB to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_SCALLTAB' for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_SCALLTAB
---------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_SCALLTAB
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table SCALLTAB from file
             SCALLTAB$, which was created by procedure WRITE_SCALLTAB.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the SCALLTAB$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process SCALLTAB$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_IBNSC
-------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_IBNSC
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table IBNSC from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_IBNSC
----------------------
Implementation: RETDTAB
Procedure Name: WRITE_IBNSC
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table IBNSC to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_IBNSC' for restoral information.
Failure Paths: If unable to dump the table.
```

```
Step Name: RESTORE_IBNSC
------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_IBNSC
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table IBNSC from file IBNSC$,
             which was created by procedure WRITE_IBNSC.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the IBNSC$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process IBNSC$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_CELLFEAT
----------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_CELLFEAT
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table CELLFEAT from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_CELLFEAT
-------------------------
Implementation: RETDTAB
Procedure Name: WRITE_CELLFEAT
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table CELLFEAT to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_CELLFEAT' for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_CELLFEAT
---------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_CELLFEAT
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table CELLFEAT from file
             CELLFEAT$, which was created by procedure WRITE_CELLFEAT.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the CELLFEAT$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process CELLFEAT$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_ACSCALL
---------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_ACSCALL
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table ACSCALL from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_ACSCALL
------------------------
Implementation: RETDTAB
Procedure Name: WRITE_ACSCALL
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table ACSCALL to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_ACSCALL' for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_ACSCALL
--------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_ACSCALL
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table ACSCALL from file
             ACSCALL$, which was created by procedure WRITE_ACSCALL.
Failure Paths: This procedure will return false when one of the following
               occur:
```

```
               - failure to open the ACSCALL$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process ACSCALL$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_RCFCLI
--------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_RCFCLI
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table RCFCLI from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_RCFCLI
-----------------------
Implementation: RETDTAB
Procedure Name: WRITE_RCFCLI
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table RCFCLI to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_RCFCLI' for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_RCFCLI
-------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_RCFCLI
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table RCFCLI from file
             RCFCLI$, which was created by procedure WRITE_RCFCLI.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the RCFCLI$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process RCFCLI$ file
               - failure to restore the entire table datafill
```

```
Step Name: TRANSFER_ATTCONS
---------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_ATTCONS
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table ATTCONS from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_ATTCONS
------------------------
Implementation: RETDTAB
Procedure Name: WRITE_ATTCONS
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table ATTCONS to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_ATTCONS' for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_ATTCONS
--------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_ATTCONS
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table ATTCONS from file
             ATTCONS$, which was created by procedure WRITE_ATTCONS.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the ATTCONS$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process ATTCONS$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_SLELIST
---------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_SLELIST
Procedure Type: ACTIVE_PROC
```

```
Description: This procedure will dump and restore table SLELIST from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.


Step Name: WRITE_SLELIST
------------------------
Implementation: RETDTAB
Procedure Name: WRITE_SLELIST
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table SLELIST to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_SLELIST' for restoral information.
Failure Paths: If unable to dump the table.


Step Name: RESTORE_SLELIST
--------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_SLELIST
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table SLELIST from file
             SLELIST$, which was created by procedure WRITE_SLELIST.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the SLELIST$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process SLELIST$ file
               - failure to restore the entire table datafill


Step Name: TRANSFER_KSETQCK
---------------------------
Implementation: APPLDTAB
Procedure Name: TRANSFER_KSETQCK
Procedure Type: ACTIVE_PROC
Description: This procedure will dump and restore table KSETQCK from the
             active side to the inactive side via the TABXFR facilities.
Failure Paths: If the entire table is not restored.
```

```
Step Name: WRITE_KSETQCK
------------------------
Implementation: RETDTAB
Procedure Name: WRITE_KSETQCK
Procedure Type: Retrofit Dump Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure dumps the datafill in table KSETQCK to the device
             designated during the start of the retrofit process, usually
             a tape drive. The tape is then physically transferred from the
             NT40 to the SuperNode for restoral. Refer to step
             'RESTORE_KSETQCK' for restoral information.
Failure Paths: If unable to dump the table.
```

```
Step Name: RESTORE_KSETQCK
--------------------------
Implementation: RETDTAB
Procedure Name: RESTORE_KSETQCK
Procedure Type: Retrofit Restore Side
Description: This step is used during a NT40 to SuperNode conversion. This
             procedure restores the datafill in table KSETQCK from file
             KSETQCK$, which was created by procedure WRITE_KSETQCK.
Failure Paths: This procedure will return false when one of the following
               occur:
               - failure to open the KSETQCK$ file
               - failure to initialize the reformat resources
               - failure to setup resources for reformatting
               - failure to process KSETQCK$ file
               - failure to restore the entire table datafill
```

```
Step Name: MASK_CUSTFLDS
------------------------
Implementation: APPLMASK
Procedure Name: MASK_CUSTFLDS
Procedure Type: ACTIVE_PROC
Description: This step will mask the following fields from tables
             CFW and CFX by adding the field entries to table CUSTFLDS:
             - CFW fields 1, 2 & 3
             - CFX fields 1 & 2
             These fields were made visible by step UNMASK_CUSTFLDS.
Failure Paths: An error message will be displayed to the user and a swer
               will be generated if the procedure is unable to delete the
               entries from table CUSTFLDS.
Note: This step should be removed.
```

```
Step Name: MATE_MASK_CUSTFLDS
-----------------------------
Implementation: APPLMASK
Procedure Name: MASK_CUSTFLDS
Procedure Type: INACTIVE_PROC
Description: This step will mask the following fields from tables
             CFW and CFX by adding the field entries to table CUSTFLDS:
             - CFW fields 1, 2 & 3
             - CFX fields 1 & 2
             This is the same as step MASK_CUSTFLDS except it is performed
             on the inactive side.  The fields were made visible by step
             MATE_UNMASK_CUSTFLDS.
Failure Paths: An error message will be displayed to the user and a swer
               will be generated if the procedure is unable to add the
               entries to table CUSTFLDS.
Note: This step should be removed.


Step Name: TRACE_OFF
--------------------
Implementation: APPLUTIL
Procedure Name: TRACE_OFF
Procedure Type: ACTIVE_PROC
Description: This procedure will return the trace device value to NIL and
             close the trace file on the active side.  It was started
             earlier by step TRACE_ON.
Failure Paths: None


Step Name: MATE_TRACE_OFF
-------------------------
Implementation: APPLUTIL
Procedure Name: TRACE_OFF
Procedure Type: INACTIVE_PROC
Description: This procedure will return the trace device value to NIL and
             close the trace file on the inactive side.  It was started
             earlier by step TRACE_ON.
Failure Paths: None


Step Name: PM_EXEC_DELTA
------------------------
Implementation: PRLDIMP
Procedure Name: PM_EXEC_DELTA
Procedure Type: ACTIVE_PROC
Description: This procedure will perform a delta on the execs of inservice
             series 1 peripherals between the current execs and the new load
             execs. If there is a difference in the execs, a list of the
             peripherals requiring the new execs will be displayed to the
             user. These execs will be loaded after swact.
Failure Paths: - The procedure will generate a SWCT log and message the
                 user if it is unable to obtain the inactive BCS/CSP level.
               - The procedure will return false if one of the following
                 fails to occur:
                 - startup exec delta
                 - populate active execs into the delta table
                 - allocate mate memory for pm_exec_delta_table
                 - send pm_exec_delta_table to the mate and fill in the
                   mate execs
                 - retrieve the pm_exec_delta_table from the mate
                 - determine which execs are required
                 - send data table to mate to update node_data of all PMs
                 - update inactive node data and dealloc all mate store


Step Name: CHECK_DISK_VOLS
--------------------------
Implementation: APPLDDU
Procedure Name: CHECK_DISK_VOLS
Procedure Type: ACTIVE_PROC
Description: This procedure checks to ensure that all non-critical disk
             volumes do not contain open files. Disk volumes used in
             DIRP (Device Independent Recording Package) are considered
             critical volumes.
Failure Paths: If a non-critical file is found open, a message will be
               displayed to the user indicating the volume that has the open
               files and possibly the filenames.


Step Name: DUMP_DIRPPOOL
------------------------
Implementation: APPLDIRP
Procedure Name: DUMP_DIRPPOOL
Procedure Type: INACTIVE_PROC
Description: This procedure dumps table DIRPPOOL to a file on the inactive
             called DIRP_INAC. The file is created for use by step
             CHECK_DIRPPOOL.
Failure Paths: This procedure will always return true, however a SWER will
               be generated if the table fails to DUMP.


Step Name: COPY_DIRP_REC
------------------------
Implementation: RETUTIL
```

```
Procedure Name: COPY_DIRP_REC
Procedure Type: Retrofit Restore Side
Description: **************** This procedure should be removed. *************


Step Name: CHECK_DIRPPOOL
-------------------------
Implementation: APPLDIRP
Procedure Name: CHECK_DIRPPOOL
Procedure Type: ACTIVE_PROC
Description: This is an active side procedure that displays both active and
             inactive datafill in table DIRPPOOL to allow the craftsperson
             to adjust datafill prior to the activity switch. The craftsperson
             is also advised that any TAPE volumes will need to be recovered
             after the switch of activity.  See related step DUMP_DIRPPOOL.
Failure Paths: An error message will be displayed to the user and the step
               will fail to complete successfully, if any of the following
               fails to occur:
               - get CUSTTAB data for table DIRPPOOL
               - display all tuples in table DIRPPOOL
               - retrieve file DIRP_INAC from the inactive side
               - find file DIRP_INAC in SFDEV after retrieving it.
               - open file DIRP_INAC for read
               - display all records within the file DIRP_INAC


Step Name: CHECK_DIRPSSYS
-------------------------
Implementation: APPLDIRP
Procedure Name: CHECK_ADDED_DIRPSSYSTEMS
Procedure Type: INACTIVE_PROC
Description: This step determines if bound DIRP (Device Independent Recording
             Package) subsystems are datafilled in table DIRPSSYS.  Any bound
             subsystems that are not datafilled in table DIRPSSYS will be
             datafilled with default data into tables DIRPPOOL and DIRPSSYS,
             with the exception of KT and KTRK.
Failure Paths: The step will fail to complete successfully if any of the
               following fails to occur:
               - get type id of the possible subsystem names
               - get the logical table id of DIRPSSYS
               - add tuple to table DIRPPOOL
               - add tuple to table DIRPSSYS


Step Name: UPDATE_METERS
------------------------
Implementation: INTSWCT1
```

```
Procedure Name: UPDATEMETERCOUNTS_PROC
Procedure Type: ACTIVE_PROC
Description: This procedure is used in international offices.  Its purpose is
             to update the meter counts on the new side with those of the old
             active side. Three meter counts are transferred, they are the line
             software meters, the trunk software meters, and the deassigned
             meters.
Failure paths: If any of the meters fails to transfer to the inactive side the
               step will fail.


Step Name: SNIX_HSDF_TRANSFER
-----------------------------
Implementation: HSDFEN
Procedure Name: XFER_HSDF_INFO
Procedure Type: ACTIVE_PROC
Description: This step transfers the SuperNode UNIX (SNIX) Hosted SNIX
             Distributed File (HSDF) to the inactive side.
Failure Paths: This procedure will fail if the data cannot be transferred.


Step Name: UPDATE_WAKEUP_CALLS
------------------------------
Implementation: INTSWCT1
Procedure Name: UPDATEWAKEUPCALLS_PROC
Procedure Type: ACTIVE_PROC
Description: This procedure is used in international offices. Its purpose is
             to update the wakeup data on the inactive side with that of the
             old active side.
Failure Paths: This procedure will return false if it is unable to transfer
               the wake up data.


Step Name: FBS_PRESWACT
-----------------------
Implementation: FBSEDTIM
Procedure Name: FBS_PRESWACT
Procedure Type: ACTIVE_PROC
Description: This procedure transfers FBS data, if any exists, when a valid
             software load is applied to the inactive side. FBS_PRESWACT
             ensures that FBS supports the software load being applied to the
             inactive side, and if there is FBS data to transfer, it invokes
             the PRESWACT data transfer utility to transfer the data.
             See related step FBS_ABORT_SWACT.
Failure Paths: This step will failure if any of the following fails to occur:
               - get the mate CM load identifier(formally known as BCS level)
               - transfer the FBS data to the inactive
```

```
Step Name: TRANSFER_GSM_DMS_HLR
-------------------------------
Implementation: GHPRESIM
Procedure Name: PRESWACT_UPDATE_PROCEDURE
Procedure Type: ACTIVE_PROC
Description: This procedure transfers some of the DMS-Home Location Register
             (HLR) dynamic data to the inactive side during PRESWACT.
             It helps to reduce the amount of data that needs to be
             transferred during NORESTARTSWACT.
Failure Paths: This step will SWER and return false if the following fails
               to occur:
               - initialize subscribers warm swact required flags to false
               - message the inactive to start up the GHPRESWT process
               - send and received the setup information to/from the
                 inactive side
               - transfer the DMS-HLR dynamic data
               - message inactive when complete


Step Name: SRDB_INFO_TRANSFER
-----------------------------
Implementation: SRDBMEM
Procedure Name: SEND_SRDB_INFO
Procedure Type: ACTIVE_PROC
Description: This procedure transfers the Selective Routing DataBase (SRDB)
             information to the inactive side.
Failure Paths: If the procedure fails to transfer the SRDB information, an
               error message will be displayed to the user and the procedure
               will return false.


Step Name: OCNC_INFO_TRANSFER
-----------------------------
Implementation: YOCNCINF
Procedure Name: SEND_OCNC_INFO
Procedure Type: ACTIVE_PROC
Description: This procedure transfers the Operator Centralization Night
             Closedown (OCNC) information to the inactive side.
Failure Paths: If the procedure fails to transfer the OCNC information, an
               error message will be displayed to the user and the procedure
               will return false.


Step Name: REMOVE_DTA_CON
-------------------------
```

```
Implementation: DTAUTIL
Procedure Name: REMOVE_DTA_FOR_ONP
Procedure Type: ACTIVE_PROC
Description: This procedure removes all Digital Test Access (DTA) connections
             to ensure that no DTA connection information remains in any of
             the XPM during the NORESTARTSWACT.
Failure Paths: None


Step Name: AUTOLD_TRANSFER
-------------------------
Implementation: APPLALD
Procedure Name: SEND_AUTOLD
Procedure Type: ACTIVE_PROC
Description: Reads the AUTOLOAD register and transfers value. Then invokes
             procedure SET_AUTOLD (imp: APPLALD) on the inactive side,
             which sets the AUTOLOAD route.  This is the AUTOLOAD device
             which the system attempts to boot from upon severe system
             failure.
Failure Paths:


Step Name: STOP_BILL_SERVER
---------------------------
Implementation: APPLEIOC
Procedure Name: BILL_SVR_STOP
Procedure Type: ACTIVE_PROC
Description: This procedure will inform the inactive side whether the billing
             server is in use on the active side. If the server is in use the
             user will be asked to deactivate it.  It will be reactivated in
             step CHECK_BILL_SERVER.
Failure Paths: - If the billing server is active, the user will be asked to
                 deactivate the server and the step will fail.
               - If the procedure is unable to inform the inactive side as to
                 the state of the server on the active side, the step will
                 fail.


Step Name: TRANSFER_LCM_DATA
----------------------------
Implementation: APPLLCM
Procedure Name: TRANSFER_LCM_DATA
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This step transfers the REX_ENABLED data for LCMs from the active
             to the inactive.  REX is Routine Exercise.
Failure Paths: Step fails if the data transfer fails.
```

```
Step Name: SEND_SFDEV_FILES
---------------------------
Implementation: APPLUTIL
Procedure Name: SEND_SFDEV_FILES
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This step copies files in SFDEV on the active processor to SFDEV
             on the inactive processor.  Load files, patch files, and empty
             files will not be copied.
Failure Paths: Step fails if process fails to copy any file to inactive.


Step Name: MS_CHECK
-------------------
Implementation: APPLMS
Procedure Name: MS_CHECK
Procedure Type: ACTIVE_PROC
Description: This step ensures that at least one of the two MSs contains a
             load compatible with the new CM load.
Failure Paths: This step will fail if neither MS has a compatible load or
               the procedure was unable to determine the MS and CM loads.


Step Name: MATE_MEM_CHECK
-------------------------
Implementation: MATEMCHK
Procedure Name: MM_CHECK
Procedure Type: PRESWACT_INACTIVE_PROC
Description: This procedure spawns a process which performs a check on
             mate memory looking for faults. It waits on a mailbox for the
             reply from the death of the process. If the process died
             normally by suicide, then the memory is OK. If the process
             traps, there is a problem.
Failure Paths: This procedure will return false, if any of the following
               occur:
               - unable to allocate a mailbox resource
               - the memory check process takes to long (timeout)
               - the memory check process returns something other then OK

Step Name: BUILD_SIS_ROUTING_DATA
---------------------------------
Implementation: SISCMSWT
Procedure Name: SIS_SWCT_MEXTRACT_AND_BUILD_TABLE
Procedure Type: ACTIVE_PROC
Description: Step turns off registration and builds a table for the Structured
             Interaction Service (SIS) database. This is done to improve the
             performance in sending the SIS data to the inactive side.
             See related step 'RESET_SIS_ROUTING_DATA'.
Failure Paths: Failed to successfully call the generic extraction routine.

Step Name:  TRANSFER_MAPS_DATA
------------------------------
Implementation: MAPSWTIM
Procedure Name: TRANSFER_MAPS_DATA
Procedure Type: ACTIVE_PROC
Description: This step transfers the AUTHCI username prefix and passwords
             from the active side to the inactive side.
Failure Paths: Step fails if the data transfer fails.

Step Name: TRANSFER_ASF_DATA
----------------------------
Implementation: APPLASF
Procedure Name: TRANSFER_ASF_DATA
Procedure Type: PRESWACT_ACTIVE_PROC
Description: This step transfers the ASF data from the active to the inactive.
Failure Paths: Steps fails if the data transfer fails.

Step Name: FBS_ABORT_SWACT
--------------------------
Implementation: FBSEDTIM
Procedure Name: FBS_ABORT_SWACT
Procedure Type: ABORT_PROC
Description: This procedure is executed whenever ABORTSWACT is invoked.
             It's purpose is to enqueue a trouble message for each agency
             indicating the abortion of the FBS data transfer and re-create
             the record links.  See related step FBS_PRESWACT.
Failure Paths: None


Step Name: DIRP_RECOVERY
------------------------
Implementation: APPLYPI
Procedure Name: DIRP_RECOVERY
Procedure Type: POSTSWCT_PROC
Description: ************* this procedure can and will be removed ********


Step Name: DIRP_AUDIT
---------------------
Implementation: APPLDIRP
Procedure Name: DIRP_AUDIT
Procedure Type: POSTSWCT_PROC
Description: This procedure will turn off the DIRP (Device Independent
             Recording Package) emergency alarm indicators and remove the
             B (bound) alarm for KT & KTRK if needed. Then it ensures the
             highest current dirp alarm is displayed.
Failure Paths: None


Step Name: HARDWARE_CHECK
-------------------------
Implementation: STATCHK1
Procedure Name: ACTIVE_DEVICE_CHECK
Procedure Type: POSTSWCT_PROC
Description:  This procedure checks that all nodes on the ACTIVE CPU are
              OK (INSV or ISTB) or OFFLINE. Any nodes not meeting this
              criteria are displayed to user. Note that any node stat
              modules that have been overridden with the MODCHECK command,
              will not have their node type statuses confirmed.
Failure Paths: If any nodes are found in an invalid state, the device names
               will be displayed to the user and the step will fail.


Step Name: CHECK_BILL_SERVER
----------------------------
Implementation: APPLEIOC
Procedure Name: CHECK_BILL_SERVER
Procedure Type: POSTSWCT_PROC
Description: This procedure works in conjunction with step STOP_BILL_SERVER.
             If the billing server was active prior to the ONP, the user
             would be instructed to re-activate the billing server.
Failure Paths: If the billing server was active, a message would be displayed
               to the user and the step would fail.


Step Name: SNIX_HSDF_STARTUP
----------------------------
Implementation: HSDFEN
Procedure Name: POSTSWCT_HSDF_STARTUP
Procedure Type: POSTSWCT_PROC
Description: This procedure obtains the SuperNode UNIX (SNIX) process indexes
             for each process to be started and makes a call to sosgdady to
             start up the hsdf server process.
Failure Paths: An error message will be displayed to the user and a SWER
               will be generated, if a process cannot be started.


Step Name: PROCESS_ONP_TIME
---------------------------
Implementation: APPLUTIL
Procedure Name: PROCESS_ONP_TIME
Procedure Type: POSTSWCT_PROC
Description: This procedure reads timing data for ONP processes that executed
             on the old load from a file, and stores this timing data in data
             structures on this new load.
Failure Paths: The step will fail if the file containing the ONP time records
               is empty or has a bad record.


Step Name: CLEAR_INVALID_REGS
-----------------------------
Implementation: APPLREG
Procedure Name: RESET_INWAT_REGS
Procedure Type: POSTSWCT_PROC
Description: This procedure will initialize the INWATS registers to
             zero.
Failure Paths: None.


Step Name: SLU_INSTALL
----------------------
Implementation: SLUZC1
Procedure Name: INSTALL_TABLES
Procedure Type: POSTSWCT_PROC
Description: Performs the equivalent of a SLU_INSTALL on tables
             TRA125I1, TRA125I2, TRA250I1, ENG640I1.  SLU is Subscriber
             Line Usage.
Failure Paths: Step fails if any problems are encountered while installing
               any of these tables.


Step Name: RESTORE_DTDETECT
---------------------------
Implementation: APPLDIGT
Procedure Name: RESTORE_DTDETECT
Procedure Type: POSTSWCT_PROC
Description: Checks a bool set by step DTDETECT_TRANSFER which specifies if
             DTDETECT was in use on the old load.  If it was in use on the
             old load, this proc will attempt to start DTDETECT on the new
             load.  DTDETECT is DigiTone DETECTion.  Steps WRITE_DTDETECT
             and READ_DTDETECT implement this functionality for retrofits.
Failure Paths: Fails only if DTDETECT can not be started.


Step Name: OMMASTER_RESTORE
---------------------------
Implementation: SOMPPIMP
Procedure Name: OMMASTER_RESTORE_PROC
Procedure Type: POSTSWCT_PROC
Description: This step creates child process SOMPOSTP.  This process is the
             OM distribution process.
Failure Paths: Step fails if unable to unprotect DS to set protected
               variables, or if the process cannot be started because
               the OMMASTER command is in use.


Step Name: TRUNK_RESTORE
------------------------
Implementation: TRKMTCH1
Procedure Name: TRUNK_RESTORE
Procedure Type: POSTSWCT_PROC
Description: This procedure restores the trunk members contained within the
             mb$trks, mb$trks2, res$trks and res$trks2 files following the
             activity switch.  These files were created by step
             TRANSFER_TRUNK_STATES.
Failure Paths: - unable to restore the trunks contained within the mb$trks and
                 mb$trks2 to the cnd_man_busy state
               - unable to restore the trunks contained within the res$trks and
                 res$trks2 to the cnd_in_only state


Step Name: RESTORE_MB_LINES
---------------------------
Implementation: LNMTCH1
Procedure Name: RESTORE_MB_LINES
Procedure Type: POSTSWCT_PROC
Description: This procedure restores the lines listed the MB$LNS file
             to the man busy (MB) state. The procedure allows for a large
             number of lines in MB state by using a second file called
             MB$LNS2.  These files were created by step DUMP_LINE_STATES.
Failure Paths: This step will fail if any of the following occur:
               - unable to allocate the line match buffers
               - unable to restore lines to the MB state


Step Name: RTS_INI_TRUNKS
-------------------------
Implementation: TRKMTCH1
Procedure Name: RTS_INI_TRUNKS
Procedure Type: POSTSWCT_PROC
Description: This procedure performs a bsy and rts on every trunk found in
             the ini state. It performs the bsy and rts on sixteen members at
             a time.
Failure Paths: unable to bsy or rts all the trunks contained within the file
```

```
Step Name: RESTORE_MB_TOPSMP
----------------------------
Implementation: YMPMTCH1
Procedure Name: RESTORE_MB_TOPSMP
Procedure Type: POSTSWCT_PROC
Description: This step sets all positions listed in file TOPSMP$MB to a MB
             state.  This file was created by step DUMP_TOPSMP_STATES.
Failure Paths: Fails if the step cannot set all positions listed in the file
               to a MB state.


Step Name: SET_SMDR
-------------------
Implementation: APPLSMDR
Procedure Name: SET_SMDR
Procedure Type: POSTSWCT_PROC
Description: Steps saves current office parameter SMDR_LOG_RPT value and
             sets the parameter to 'ALL 32767'. Parameter SMDR_LOG_RPT
             controls whether SMDR logs are output to the log device.
             SMDR is Station Message Detail Recording.  See related step
             RESET_SMDR.
Failure Paths: Failed to conditional unprotect data store.


Step Name: SET_AMA_RPT
----------------------
Implementation: APPLAMA
Procedure Name: SET_AMA
Procedure Type: POSTSWCT_PROC
Description: Step save current office parameter SPECIAL_AMA_REPORT value and
             sets the parameter to 'Y BOTH 32767'. Parameter SPECIAL_AMA_REPORT
             controls whether AMA billing logs are output to the log device.
             See related step RESET_AMA_RPT.
Failure Paths: Failed to conditional unprotect data store.


Step Name: SET_AMAB
-------------------
Implementation: APPLAMAB
Procedure Name: SET_AMAB_LOG
Procedure Type: POSTSWCT_PROC
Description: Step saves current value of LOGAMA tuple in table AMAOPTS
             and proceed to set this tuple to 'ON'. Tuple LOGAMA controls
             the generation of the AMAB117 log reports.
             See related step RESET_AMAB.
Failure Paths: - Failed to update LOGAMA tuple in table AMAOPTS.
```

```
               - Failed to conditional unprotect data store. This is attempted
                 in order to save previous tuple values.


Step Name: MATCH_ALL_UPD
------------------------
Implementation: APPLYPI
Procedure Name: MATCHALL_PROC
Procedure Type: POSTSWCT_PROC
Description: Step matches and updates all host and peripheral patches.
             Performs the equivalent of a MATCHALL UPDATE from the PATCHER
             level.
Failure Paths: - Fails if PRSM is active and DBAUDIT fails.
               - Fails if any errors occurred during the MATCHALL PROCESS.


Step Name: BEGIN_TESTING
------------------------
Implementation: APPLYC1
Procedure Name: POSTSWACT_DRIVER
Procedure Type: POSTSWCT_PROC
Description: Step informs site to begin testing and stops the process.  The
             step stops the process, like a failed step; but it does not mark
             the step as failed.  This keeps the step from being executed over
             and over.
Failure Paths: Not Available


Step Name: ENABLE_PATCH_AUDIT_POSTSWACT
---------------------------------------
Implementation: PALARMIM
Procedure Name: ENABLE_PATCH_AUDIT
Procedure Type: POSTSWCT_PROC
Description: Step enables the patch audit process if this process was
             enabled prior to the activity switch.  See related steps
             DISABLE_PATCH_AUDIT_ACT and DISABLE_PATCH_AUDIT_INACT.
Failure Paths: Failed to unprotect data store.


Step Name: POST_MS_CHECK
------------------------
Implementation: APPLMS
Procedure Name: POST_MS_CHECK
Procedure Type: POSTSWCT_PROC
Description: Step verifies that the MS loads are compatible with the CM load.
             Assures that the MS units are within three versions of the CM.
Failure Paths: - Failed to query MS node information.
```

```
               - MS not inservice.
               - MS/CM compatibility failed.


Step Name: RESET_OFFICE_TUPLES
------------------------------
Implementation: APPLUTIL
Procedure Name: RESET_OFFICE_TUPLES
Procedure Type: ABORT_AND_POSTSWCT_PROC
Description: Restores office parameters NODEREXCONTROL and LCDREX_CONTROL
             in table OFCVAR as well as office parameter
             GUARANTEED_TERMINAL_CPU_SHARE in table OFCENG from
             previously saved values.
             See related steps SET_OFFICE_TUPLES and SET_MATE_TUPLES.
Failure Paths: - Step Fails if any of the parameters fail to restore
                 from protected memory.
               - Fails if unable to update table OFCVAR through table control.


Step Name: RESET_SMDR
---------------------
Implementation: APPLSMDR
Procedure Name: RESET_SMDR
Procedure Type: POSTSWCT_PROC
Description: Step resets office parameter SMDR_LOG_RPT to 'NONE 0'.
             Parameter is used to ensure all SMDR (Station Message Detail
             Recording) records will not be printed.  See related step
             SET_SMDR.
Failure Paths: Failed to conditional unprotect data store.


Step Name: RESET_AMA_RPT
------------------------
Implementation: APPLAMA
Procedure Name: RESET_AMA
Procedure Type: POSTSWCT_PROC
Description: Restores office parameter SPECIAL_AMA_REPORT to its
             original value. Parameter determines whether AMAB logs are
             generated if office is using NT format AMA.  See related
             step SET_AMA_RPT.
Failure Paths: Failed to conditional unprotect data store.


Step Name: RESET_AMAB
---------------------
Implementation: APPLAMAB
Procedure Name: RESET_AMAB_LOG
Procedure Type: POSTSWCT_PROC
Description: Resets tuple LOGAMA in table AMAOPTS to OFF. Tuple LOGAMA
             is used to control whether AMAB logs are generated if office
             is using BC format AMA.  See related step SET_AMAB.
Failure Paths: Unable to update tuple LOGAMA through table control.


Step Name: RESTORE_PASSWORDS
----------------------------
Implementation: UDATINIT
Procedure Name: RESTORE_PASSWORDS
Procedure Type: POSTSWCT_PROC
Description: Restores the OPERATOR and ADMIN's passwords to the
             customer's set value.
Failure Paths: Not Available.


Step Name: ENABLE_AUTOIMAGE
---------------------------
Implementation: ADUMPCTL
Procedure Name: ENABLE_AUTOIMAGE
Procedure Type: ABORT_AND_POSTSWCT_PROC
Description: Enables the auto image dump feature which was previously
             disabled by step DISABLE_AUTOIMAGE.
Failure Paths: Not Available.


Step Name: RESUME_ATT
---------------------
Implementation: APPLATT
Procedure Name: ATT_RUNATT
Procedure Type: ABORT_AND_POSTSWCT_PROC
Description: Resumes scheduled Automatic Trunk Testing (ATT).
             ATT was halted on the active by step HALT_ACTIVE_ATT, and on
             the inactive by HALT_ATT.
Failure Paths: - Failed to start ATT testing.
               - Failed to activate ATT testing.

Step Name: RESUME_CTT
---------------------
Implementation: CTTAPPL
Procedure Name: CTT_RUN
Procedure Type: ABORT_AND_POSTSWCT_PROC
Description: Resumes scheduled Cell-site Terminal Testing (CTT). CTT was
             stopped on the active and inactive by steps HALT_ACTIVE_CTT
             and HALT_CTT respectively.
Failure Paths: - Unable to send a wake-up message to the CTT scheduler.
```

```
Step Name: APPLY_LOG_SETTINGS
-----------------------------
Implementation: LOGINFTC
Procedure Name: ONP_APPLY_LOG_SETTINGS
Procedure Type: POSTSWCT_PROC
Description: Applies temporary log control information (class, suppression,
             and threshold values) from the old load into the new load.
             These values are in hidden table LOGINFO.
Failure Paths: None.


Step Name: CLEANUP_SFDEV_FILES
------------------------------
Implementation: APPLUTIL
Procedure Name: CLEAN_UP_SFDEV
Procedure Type: POSTSWCT_PROC
Description: Erases ONP related files residing on SFDEV at the completion of
             the postswact process. Files which are erased have the following
             filename endings: DRNOW, INADR, PATCH$FILE, $PATCH. Step also
             erases the files matching these filenames: APPLIED_PATCH_FILE and
             RELEASE_SUFFIX_FILE. Process will prompt for (y/n) response
             before erasing files on SFDEV ending in '$PATCH'.
Failure Paths: Not Available.


Step Name: DISPLAY_SWACT_TIME
-----------------------------
Implementation: SWCTNR0
Procedure Name: DISPLAY_SWACT_TIME
Procedure Type: POSTSWCT_PROC
Description: Displays the data regarding peripheral recovery and execution times
             as well as call processing outage time during the activity switch.
Failure Paths: Not Available.


Step Name: RESET_TABXFR_TARGET
------------------------------
Implementation: APPLUTIL
Procedure Name: RESET_TABXFR_TARGET
Procedure Type: POSTSWCT_PROC
Description: Resets TABXFR target back to STANDARD platform. STANDARD
             indicates datamove is being performed on a standard
             configuration.
Failure Paths: Failed to reset target back to Standard.
```

```
Step Name: RESET_SIS_ROUTING_DATA
---------------------------------
Implementation: SISCMSWT
Procedure Name: SIS_PRESWACT_RECOVERY
Procedure Type: ABORT_PROC
Description: Step releases the table built by step 'BUILD_SIS_ROUTING_DATA'
             and turns on registration for Structured Interaction Service
             (SIS) database.
Failure Paths: Not Available.

Step Name: CLEANUP_LTC_ADNUM
----------------------------
Implementation: APPLLTC
Procedure Name: CLEANUP_LTC_ADNUM
Procedure Type: POSTSWCT_PROC
Description: This procedure will check for zero in the ADNUM field
             of table LTCINV tuples. If it finds a zero, it will
             allocate a valid ADNUM number and datafill it in the
             ADNUM field. (Zero is an invalid ADNUM number.) The
             ADNUM field is used for EADAS Operational Measurements.
Failure paths: Unable to access table ltcinv
               Unable to establish the pmid of the current tuple
               Unable to assign the adnum number in the adnum data
               structures

Step Name: CLEANUP_VR_MATE_DATA
-------------------------------
Implementation: VRPSWCT
Procedure Name: VR_CLEANUP
Procedure Type: POSTSWCT_PROC
Description:  Remove the mate version data associated with the Version
              Registry platform.  Version Registry's mate version data
              needs to be removed after an upgrade since it may no longer
              be reliable if the switch should ever "drop-sync".  This
              step is further documented in the PRS Document BX40713.
Failure Paths: - unable to delete mate data; the only impact to the failure
                 of removing mate data would occur if the switch dropped sync
                 and then a query for mate data was executed.  The mate data
                 may no longer be valid if someone  has booted a different
                 load on the inactive side.  The probability  of this occurring
                 is very minimal.
               - this step will not stop POSTSWACT should an error occur in
                 our cleanup utility.
```

CC WarmSWACT summary

SWitch of ACTivity (SWACT) is a generic term referring to a process by which activity is switched between two processors. CC WarmSWACT is a controlled SWACT where a sequence of steps is executed to ensure minimal degradation call processing when switching activity between the CPUs in the core of the switch (e.g., CM in Supernode).

Note: Only "simple" 2-port and echo calls that are in a stable talking state (that is, not in a transition state such as dialing) will survive a CC WarmSWACT. Survival means that the call is kept up until the next signaling message is received (hopefully, for example, a terminate message, but on any other message as well, such as an attempt to use the conference feature). See *Appendix B* for a procedure for testing call survivability over a CC WarmSWACT.

The three parts in this section are divided as follows:

1. Explanation of CC WarmSWACT and its major steps
2. Explanation of CC WarmSWACT commands
3. Explanation of CC WarmSWACT logs

CC WarmSWACT steps

CC WarmSWACT is a method by which a new software load can be efficiently activated in a DMS-100F switch. It ensures a controlled activity switch while minimizing degradation of service to the subscriber. To achieve this goal the process performs the following steps.

- Precheck to ensure the environment is right for the intent (e.g., switch is out of sync and inactive side is not jammed)
- Establish communication between the two CPUs
- Obtain required semi-dynamic data from the active CPU and transfer it to the inactive CPU
- Setup and allocate required resources to transfer dynamic data (e.g., originating and terminating party of calls being supported)
- Stop call processing. Freeze everything so nothing can change while activity is being switched
- Obtain and transfer all dynamic data
- SWitch ACTivity from the active CPU to the inactive CPU
- Perform additional checking to ensure sanity of new CPU and initiate recovery

- Insert the dynamic data that was transferred before the SWACT
- Resume call processing
- Cleanup and deallocate any resources used to execute the CC Warm-SWACT

CC WarmSWACT commands

The commands required to perform/monitor/report a CC WarmSWACT are as follows.

SWCT (BCS32 or lower)—directory where all commands for CC Warm-SWACT may be found. User must be in the BCSUPDATE directory to go to this directory in BCS31 and BCS32, i.e., BCSUPDATE;SWCT

System Response: Prompt changes to SWCT:

SWACTCI (BCS33 and higher)—same as SWCT but was changed in BCS33 to distinguish CC WarmSWACT from XPM SWACT.

System Response: Prompt changes to SWACTCI.

QUIT—gracefully exits SWCT/SWACTCI CI increment.

System Response: Prompt returns to previous state.

FORCESWCT/FORCESWACT—displays, enables or disables the ability for the newly active CPU to switch activity back to the previously active CPU if an abnormal condition exists (more than 10% of PMs on the newly active side are not OK). FORCESWCT for BCS32 or lower. FORCESWACT for BCS33 and higher.

Optional parameter:

- no parameter queries the status of FORCESWCT/FORCESWACT (IN EFFECT or NOT IN EFFECT is displayed).
- ON forces activity to stay on the newly active CPU even if an abnormal condition exists. This is the default setting.
- OFF allows activity to switch back if the abnormal condition exists. This should not be used unless the user definitely does not want to stay on the newly active CPU to correct problems, etc.

System Response: Log is produced when ON/OFF optional parameter is used (SWCT104).

LOADEXECS—displays, enables or disables the exec loading process which occurs after a CC WarmSWACT to download new execs to the PMs for call processing. This command has been obsoleted in BCS35 or greater.

Optional parameter:

- no parameter queries the status of LOADEXECS (ENABLED or DISABLED is displayed).
- ON enables exec loading. This is the default setting and is required when performing a CC WarmSWACT between different BCSs, i.e., BCSn -> BCSn+, BCSn+ -> BCSn.
- OFF disables exec loading after CC WarmSWACT. This should not be used unless the user definitely understands the implications of not downloading execs to PMs after a CC WarmSWACT.

System Response: Log is produced when ON/OFF optional parameter is used (SWCT104).

DISP/DISPLAY—displays information regarding CC WarmSWACT. DISP in BCS30 or lower. DISPLAY in BCS31 and higher.

Optional parameter:

- BADNODES will display all hardware devices whose status is NOT OK or OFFLINE on the active side of the switch.
- MISMATCH displays mismatches found from comparing device statuses between the active side and the inactive side of the switch.
- SWCTTIME (SWACTTIME in BCS35 and higher) displays all times collected for CC WarmSWACT (SWCT101 time, EXECTIME, RECVTIME).
- ALARM has been obsoleted in BCS33 and beyond.

System Response: Information is displayed to terminal.

QUERYSWACT—this command checks the office configuration to determine which CC WarmSWACT command (RESTARTSWACT or NORESTARTSWACT) should be used.

System Response: If the office supports NORESTARTSWACT the response is as follows:

“NORESTARTSWACT is recommended for initiating a CC WarmSWACT. Further checking will be done when SWACT is invoked.”

otherwise the following message is displayed:

“RESTARTSWACT must be used for initiating a CC WarmSWACT.”

NORESTARTSWACT—this command executes the CC WarmSWACT process and performs all of the necessary prechecks to activate the CC WarmSWACT. PRESWACT steps of BCSUPDATE must all be executed successfully before a NORESTARTSWACT will be allowed. This command is valid only in BCS36 and greater.

Optional parameter:

- NOMATCH will disable the matching of device statuses between the active and inactive sides of the switch. WARNING: Do not use this option unless there is no other choice. Device statuses after the CC WarmSWACT are not guaranteed hence several devices may be out of service after the CC WarmSWACT.

System Response: The steps being executed as part of the CC WarmSWACT will be displayed to the terminal and logs will be generated (SWCT102). SWACT will occur. If a NORESTARTSWACT cannot be executed in this office the following message will be displayed:

“RESTARTSWACT should be used instead of NORESTARTSWACT. NORESTARTSWACT command aborted.”

RESTARTSWCT/RESTARTSWACT—this command executes the CC WarmSWACT process and performs all of the necessary prechecks to activate the CC WarmSWACT. In BCS31 and greater the PRESWACT steps of BCSUPDATE must all be executed successfully before a RESTARTSWCT/RESTARTSWACT will be allowed. RESTARTSWCT in BCS32 and lower. RESTARTSWACT in BCS33 and higher.

Optional parameter:

- NOMATCH will disable the matching of device statuses between the active and inactive sides of the switch. WARNING: Do not use this option unless there is no other choice. Device statuses after the CC WarmSWACT are not guaranteed hence several devices may be out of service after the CC WarmSWACT.

System Response: The steps being executed as part of the CC WarmSWACT will be displayed to the terminal and logs will be generated (SWCT102). SWACT will occur followed by a COLD restart. If a NORESTARTSWACT is supported by this office the following message will be displayed which requires a YES/NO response from the user:

"NORESTARTSWACT should be used instead of RESTARTSWACT.
Do you wish to continue with RESTARTSWACT?"

ABORTSWCT/ABORTSWACT—this command executes the CC WarmSWACT process and performs all of the necessary prechecks to activate the CC WarmSWACT. In BCS31 and lower the RESTARTSWCT command should be used in place of this command. This command does not require PRESWACT to be performed before execution. This command should only be used when aborting a BCS application. ABORTSWCT in BCS32 and lower. ABORTSWACT in BCS33 and higher.

Optional parameter:

- NOMATCH will disable the matching of device statuses between the active and inactive sides of the switch. WARNING: Do not use this option unless there is no other choice. Device statuses after the CC WarmSWACT are not guaranteed hence several devices may be out of service after the CC WarmSWACT.
- NOCHECK will override the requirement for all devices to be OK before a CC WarmSWACT. Therefore a device can be CBSY for instance and the CC WarmSWACT will still be allowed. Available in BCS34 and higher. WARNING: Use this option only as a last choice after exploring other choices.

System Response: The steps being executed as part of the CC WarmSWACT will be displayed to the terminal and logs will be generated (SWCT102). SWACT will occur followed by COLD restart.

STATUSCHECK—this command matches statuses for devices between the active and inactive side of the switch. It verifies that the STATUSUPDATE step executed in PRESWACT was successful.

System Response: A SWCT109 log is generated for each type of device that has passed the STATUSCHECK process. A SWCT110 log is generated for each type of device that has failed the STATUSCHECK process. Each device of the failed type that mismatches is displayed to the terminal.

MODCHECK—this command checks for necessary CC WarmSWACT application modules on the inactive side and outputs any modules which are missing. Missing modules will cause the CC WarmSWACT to fail and therefore must be investigated or overridden via the OVERRIDE option to MODCHECK. This command is valid in BCS32 and beyond.

Optional parameter:

- no parameter invokes checking for all CC WarmSWACT application modules.
- OVERRIDE will disable the checking for requested missing modules and hence disable the functions performed by those CC WarmSWACT applications.
- RESET will enable the checking for requested missing modules and hence enable the functions performed by those CC WarmSWACT applications.

System Response: SWCT113 log will be output if MODCHECK is successful. SWCT114 log will be output if MODCHECK fails. SWCT115 log will be for every missing module. SWCT116 log will be output for every module for which the OVERRIDE/RESET options are used.

RESUMEPM—should not be used. This is a very dangerous command and should only be used by qualified personnel.

System Response: Some PMs may go SYSB. Do not use this command.

RESTOREXECS—this command will load execs to any or all PM types.

CAUTION
This command should only be used in emergency situations by qualified personnel.

Non optional parameter:

- <PM_TYPE> {TM, LM, DCM, RLM, XPM, ALL}

System Response: PM type(s) chosen will have execs loaded. No response to terminal.

CC SWACT logs

SWCT101—Information log only. This log does not indicate a service affecting problem. Displays the WarmSWACT time.

SWCT102—Information log only. This log does not indicate a service affecting problem. Indicates which CC WarmSWACT step successfully completed.

SWCT103—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates which CC WarmSWACT step failed.

SWCT104—Information log only. This log does not indicate a service affecting problem. Indicates a condition or state of the CC WarmSWACT process.

SWCT105—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates why a CC WarmSWACT step failed.

SWCT106—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates the underlying problem of why a CC WarmSWACT step failed.

SWCT107—Information log only. This log does not indicate a service affecting problem. Indicates that exec loading occurred successfully to the reported PM type. This log has been obsoleted in BCS35 and beyond.

SWCT108—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates that exec loading failed to the reported PM type. This log has been obsoleted in BCS35 and beyond.

SWCT109—Information log only. This log does not indicate a service affecting problem. Indicates that a STATUSCHECK application passed.

SWCT110—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates that a STATUSCHECK application failed.

SWCT111—Information log only. This log does not indicate a service affecting problem. Indicates that the PRELOAD_EXECS step of PRESWACT completed successfully.

SWCT112—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates that PRELOAD_EXECS failed for an XPM, one log will be reported for every XPM that failed.

SWCT113—Information log only. This log does not indicate a service affecting problem. Indicates that the MODCHECK command passed successfully.

SWCT114—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates that the MODCHECK command failed.

SWCT115—Trouble log. This log indicates a service affecting problem and must be investigated in order for the CC WarmSWACT to complete successfully. Indicates which modules are missing on the inactive side according to the MODCHECK command, one log will reported for every missing module.

SWCT116—Information log only. This log does not indicate a service affecting problem. Indicates that a module has been OVERRIDDEN/RESET for checking by the MODCHECK command.

SWCT117—Information log only. This log does not indicate a service affecting problem. Displays information about the CC WarmSWACT process.

Appendix B: Supplementary Procedures

Appendix B table of contents

Procedure B-1
Steps to execute manual TABAUDIT

Note: The following procedure is one method of verifying every table in the office. Completing these steps will manually activate a TABAUDIT session from the device on which the commands are entered.

1 **Site/ACT** Set-up and execute a manual TABAUDIT.

a.
```
> TABAUDIT
TABAUDIT:
```
Enters the TABAUDIT increment.

b.
```
TABAUDIT:
> INCLUDE ALL
```
This option will include all tables in the office.

c. Start the TABAUDIT session.

> **CAUTION**
> **Do not attempt to view the Summary File until the TABAUDIT has completed.**
> Doing so will terminate the TABAUDIT session.

```
> EXECUTE
```
This first shows a STATUS. If correct, confirm with "YES" when prompted.

TABAUDIT now executes the various data integrity checks on each tuple of every table in the office.

****** TERMINATION—To stop an active TABAUDIT session (not automated) type *<break> hx*.
This quits the TABAUDIT increment which will terminate the session and clear all settings made for this *session.*

—continued—

Procedure B-1
Steps to execute manual TABAUDIT (continued)

The following example illustrates the correct use of the EXECUTE command of TABAUDIT.

Example of tabaudit execute

```
TABAUDIT:
> EXECUTE

-----------------------------------------------------
|                                                   |
|                 TABAUDIT STATUS                   |
|                                                   |
-----------------------------------------------------

The following tables are included:

   From table ACTPATCH (0)                  to table SSRFORM
(479).

The following tables are excluded:

   No tables have been excluded.

Please confirm ("YES", "Y", "NO", or "N"):

> yes
Creating TABAUDIT summaryfile: SUMMARY$MAP on SFDEV.
Starting DMS data verification ...
   .
   .
Data verification is completed.
```

2 After the TABAUDIT has finished running, obtain an error REPORT to see the results of the testing.

Note: Several useful options are available with the REPORT command. Type HELP REPORT to see the options.

```
TABAUDIT:
> REPORT ERRORS
```

The REPORT ERRORS command displays tuples that are in question. To see detailed information on why a tuple has failed, position on the table tuple in question and issue a CHECK command. This will display the verify proc messages that fail.

—continued—

Procedure B-1
Steps to execute manual TABAUDIT (continued)

3 After reviewing reported table data errors, *correct all the errors*. Then execute TABAUDIT on the fixed tables to verify there are no further errors.

```
TABAUDIT:
> INCLUDE <table_name>

TABAUDIT:
> EXECUTE

TABAUDIT:
> REPORT <table_name>
```

Nortel recommends that data integrity checks using TABAUDIT be made a regular and ongoing part of normal maintenance procedures. By using the automatic scheduling function this can be accomplished with minimal effort.

A more detailed explanation of how TABAUDIT operates is included in *Appendix A* under “Using TABAUDIT.”

Procedure B-2
Converting one PM to another

During a software application it is possible to change the key field LTCNAME from an LGC to an LTC. This would be done during the data transfer to eliminate having to delete and re-add the peripheral. Telco then is responsible to change the FRAMENAME, LOADNAME, and EXEC LINEUP information to meet their needs. This procedure should be used to accomplish this.

Do the following after the CC SWACT when you have converted one PM type to another PM type (for example, an LGC to LTC conversion).

1 **Telco/ACT** Busy the inactive unit of the peripheral to be reloaded.

2 Make appropriate changes in table LTCINV for the FRAMENAME, LOADNAME, and EXECS for the peripheral being modified.

3 Load, patch, and rts unit x nodatasync on the inactive unit.

4 Perform a cold swact to the newly loaded side.

5 Busy the newly inactive unit.

6 Set the patch set against that unit, load and perform a regular rts.

7 Repeat steps 1–6 for any remaining peripherals to be modified.

Procedure B-3
MATE IMAGE capture

For SuperNode only the following procedure for dumping an inactive (mate) image may be useful in an abort situation or whenever an image of the inactive CM is required.

When dumping a mate image of a TABXFER'ed load it is important to realize that the image you are taking will have all peripherals in an OFFL state; therefore, this image is not BOOTABLE as it will not have the minimum configuration of at least one IOC (or IOM) and one TERMDEV in an "IN-SERVICE" state. To avoid this problem we will be RTS'ing the minimum configuration manually (steps 5 and 6).

1 **App/ACT** Prepare the SLM volume to be used to dump the inactive (mate) image.

```
> DISKUT
```

```
> LF S00DIMAGE
```
{example}

Lists the volume on which you want to put the image.

---

2 **ACT** From the active side MATEBIND the SLM volume you have chosen.

```
> MATELINK RTS
```

```
> MATEIO
```

```
> MATEBIND S00DIMAGE S00DIMAGE
```
{example}

---

3 **ACT** From the active side MATELOG to the mate side.

```
> MATEIO
```

```
> MATELOG <device>
```

where <device> is the INACT terminal.

---

4 **INACT** On the inactive side LOGIN as OPERATOR ORERATOR.

---

5 **INACT** `Mate> MAPCI;MTC;IOD;IOC 0;BSY IOC` *{or IOM}*

Mate MAPCI will not display.

---

6 **INACT** On the Mate side BSY and RTS the same location that the MAP is datafilled on the active side (example: CARD 2 PORT 0;BSY 0;RTS 0).

The RTS will fail, but this is expected.

—continued—

Procedure B-3
MATE IMAGE capture (continued)

7 **INACT**

```
Mate> QUIT MAPCI
```

```
Mate> PRINT MATEIODIR
```

You should see the file S00DIMAGE in MATEIODIR.

8 **INACT** From the inactive side DUMP the inactive (mate) image.

```
Mate> MATEIO
```

```
Mate> DUMP IMAGE S00DIMAGE ACTIVE RETAIN NODE CM
```

{example}

This command will give a couple of messages about not being able to translate the IOC devices—ignore the messages.

Wait for image dump to complete.

9 **ACT** From the active side loadmate the mate image just dumped to verify it.

Note: LDMATE time will be approximately 10 to 15 minutes.

```
> LDMATE DIRECT DISK S00DIMAGE
```

{example}

Upon completion, system response should be:

```
DIRECT LDMATE complete.
Confirm successful initialization on Inactive CPU
before proceeding.
```

Similarly, the RTIF response is:

```
\BOOTING...
Loading completed...
```

Wait for loading to complete. Initialization is confirmed when the inactive processor flashes A1.

Procedure B-4
Enabling PRSM

Following is information on enabling (or disabling) the PRSM patching manager. This is only a brief introduction to PRSM.

PRSM retains all functionality of Patcher, plus:

- works on single or a set of patches,
- applies patches in correct order,
- determines the destination for patches.

In NA004B (Base05) Patcher is default, and PRSM is available for activation (POST-application).

In NA005 (Base06) PRSM is default, but Patcher is still available.

Beyond NA005 only PRSM can be used.

Note: Currently a password is required to enable PRSM. This requirement should be removed as PRSM is further deployed.

To enable PRSM from a map

```
Commands to enter:   Action taking place:
------------------   --------------------

prsmdbg              % Enters the PRSMDBG CI.

qprsm                % Verify Patcher is the active patching
                       manager.

enableprsm           % Enables PRSM if Patcher is the active
                       patching manager.

*** PRSM may require the user to enter a password ***

*** Obtain the password from your NORTEL support person ***

<password>           % Enter the password to enable PRSM.

qprsm                % Verify PRSM has been enabled.
```

—continued—

Procedure B-4
Enabling PRSM (continued)

****** Example of enable**

```
> prsmdbg

PRSMDBG:

> qprsm

Currently, Patcher is the active patch manager.

> enableprsm

A password must be entered to enable PRSM.

Please enter the password:

Performing all PRSM steps to SYNC PRSM to Patcher.
```

Follows with various messaging and audits....

```
PRSM is now enabled.

> qprsm

Currently, PRSM is the active patch manager.

record stop onto sfdev
```

To disable PRSM from a map

```
Commands to enter:   Action taking place:
------------------   --------------------

prsmdbg              % Enters the PRSMDBG CI.

qprsm                % Verify PRSM is the active patching
                       manager.

disableprsm          % Disables PRSM if PRSM is the active
                       patching manager.

yes                  % Enter "yes" to the prompt asking if you
                       really wish to switch back to Patcher.

qprsm                % Verify PRSM has been disabled.
```

—continued—

Procedure B-4
Enabling PRSM (continued)

****** Example of disable**

```
> prsmdbg

PRSMDBG:

> qprsm

Currently, PRSM is the active patch manager.

> disableprsm

Are you sure you want to switch back to Patcher?

Please confirm ("YES", "Y", "NO", or "N"):

> y

Performing all Patcher steps to SYNC Patcher to PRSM.
```

Follows with various messaging and audits....

```
Patcher is now enabled.

> qprsm

Currently, Patcher is the active patch manager.

record stop onto sfdev
```

Procedure B-5
DIRP and billing preparation

(formerly PRESWACT DIRP and billing)

Site and Applicator can work together to prepare the PRIMARY DIRP billing subsystems for the CC switch of activity (SWACT). This procedure gives the steps to accomplish this preparation.

PRESWACT step CHECK_DIRPPOOL is an active side procedure which displays both active and inactive datafill in table DIRPPOOL to allow the craftsperson to adjust datafill prior to the activity switch. The craftsperson is also advised that any TAPE volumes will need to be recovered after the switch of activity.

1 Disk drive PRIMARY billing

With this step primary disk volumes can recover automatically after SWACT.

a. **Site/ACT** If on disk (DDU), from the DIRP level ROTATE any active billing subsystem (such as AMA SMDR OCC CDR).

b. If required by Telco policy copy unprocessed DIRP files to back-up tape (using DIRPAUTO or DIRPCOPY commands).

c. Verify that table DIRPHOLD contains no unprocessed billing files (if DIRPAUTO was used above).

d. **Site and App/INACT** Ensure that regular disk volumes are in table DIRPPOOL on the inactive side. This allows the disk to be recovered by DIRP after SWACT. If necessary manually datafill the volume names in DIRPPOOL on the inactive side before SWACT.
PRESWACT step CHECK_DIRPPOOL displays the datafill for table DIRPPOOL on the inactive side.

---

2 Tape drive PRIMARY billing

With this step primary tape volumes can recover automatically after SWACT.

a. **Site/ACT** If on tape (MTD), from the DIRP level ROTATE any active billing subsystem (such as AMA SMDR OCC CDR), CLOSE the standby file, and DMNT the standby volume.
Example:

```
ROTATE AMA
CLOSE AMA STDBY 1
DMNT AMA T1   {standby volume}
```

b. Remove the demounted standby tape from the tape drive, and put up a new tape to be used as the next DIRP volume.

—continued—

Procedure B-5
DIRP and billing preparation (continued)

c. Prepare a new standby volume as follows.

```
> MOUNT <x> FORMAT <volume_id>
```

where <x> is the standby device number, and <volume_id> is the name of the standby volume.

If prompted enter the first filename, or if system response is:
"request aborted. Tape not expired (use ERASTAPE)"
then select an unused or expired tape for formatting.

```
> DEMOUNT T<x>
```

Leave the standby volume at load point and ON LINE. Immediately following SWACT, it will become the ACTIVE volume of the appropriate subsystem.

d. **Site and App/INACT** Ensure that regular tape volumes are in table DIRPPOOL on the inactive side. This allows the tape to be recovered by DIRP after SWACT. If necessary manually datafill the volume names in DIRPPOOL on the inactive side before SWACT. *PRESWACT step CHECK_DIRPPOOL displays both active and inactive datafill in table DIRPPOOL.*

3 DPP/BMC PRIMARY billing

With this step primary DPP/BMC volumes can recover automatically after SWACT.

a. **Site/ACT** Perform this step to close the last file on the DPP and open a new one. Telco may POLL the DPP if desired when this is complete.

```
> MAPCI;MTC;IOD;DPP AMA;IDXMAINT CREATE FILE AMA
```

This re-establishes the block header on the DPP.

b. **Site and App/ACT** If SMDR recording is on BMC (datafilled as TAPE in table DIRPPOOL) and NO standby volume is available for BMC, then mount a temporary STDBY TAPE volume. In table DIRPPOOL add the TAPE device as a standby BMC. Also add the device on the inactive side (see the following substep). Leave the STDBY TAPE demounted. DO NOT ROTATE the BMC. This volume will be used to rotate the BMC port OUT and back IN during POSTSWACT.

Note: Some SMDR recording applications on BMC collect SMDR records based on customer group ID only, and it is necessary to rotate the BMC tape port IN during POSTSWACT to ensure that any changes to the customer group IDs are passed to the BMC upon rotate (to ensure the RECORD HEADER is correct).

—continued—

Procedure B-5
DIRP and billing preparation (continued)

c. **Site and App/INACT** Ensure that regular DPP/BMC volumes are in table DIRPPOOL on the inactive side. If necessary manually datafill the volume names in DIRPPOOL on the inactive side before SWACT. *PRESWACT step CHECK_DIRPPOOL displays both active and inactive datafill in table DIRPPOOL.*

```
Mate> TABLE DIRPPOOL;POS <pool_#>
```

where <pool_#> is the number for DPP AMA pool.

Verify field DEVTYPE in table DIRPPOOL is DPP (not TAPE).

CAUTION
In a pool of DPP or BMC volumes, field DEVTYPE in table DIRPPOOL should be DPP (not TAPE).
Otherwise, if datafilled as TAPE you will have to recover the volume manually after SWACT.

4 **Parallel DIRP**

App Applicator should make a note of how the parallel devices are allocated in table DIRPPOOL.

Site Telco is responsible to recover parallel AMA after SWACT as needed. Parallel DDU should come up automatically, parallel tape will have to be remounted, preferably with new tape.

Note: DIRP no longer supports (BCS34) parallel AMA recording on a DPP or BMC volume. Table control prohibits the filling of devtype DPP in a parallel pool.

CAUTION
Recently recorded parallel data may be overwritten.
Site should copy the parallel files to tape to prevent loss of parallel data if that is Telco policy.
- If a single parallel volume is in use, information on the volume will be lost over SWACT.
- If more than one parallel volume is allocated, DIRP will start recording after SWACT on the volume with the oldest time stamp. Hence information on that volume will be lost over SWACT.

Testing call survivability over a CC WarmSWACT

This section provides a procedure for testing call survivability over a CC WarmSWACT and sample call scripts. These are provided as guidelines for the testing of calls being supported over the CC WarmSWACT.

Procedure for testing call survivability

1. Ensure that the best possible mix of the above call scripts are used for the following procedure.

---

2. Establishing call—Just before the CC WarmSWACT perform the following:

On the originating set:

- Take handset off hook and dial the desired number
- Wait for terminating set to pick up
- Ensure that a voice path has been established by blowing into phone on originating set and listening for the blowing on the terminating set

On the terminating set:

- Wait for ringing
- Take handset off hook
- Ensure that a voice path has been established by blowing into phone on terminating set and listening for the blowing on the originating set

Leave both handsets offhook

Note: Only stable (in a talking state—not in transition like dialing or feature activation mode) two port calls are maintained over CC WarmSWACT. Any call which involves a feature/extension data block or service circuit will not be maintained (e.g., call waiting, call forwarding, conference call).

---

3. Testing call—Right after the new CPU takes activity (i.e., during the restart or recovery sequence on the newly active CPU) perform the following:

On the originating set:

- Ensure that a voice path has been maintained by blowing into phone on originating set and listening for the blowing on the terminating set

On the terminating set:

- Ensure that a voice path has been maintained by blowing into phone on terminating set and listening for the blowing on the originating set

Leave both handsets offhook.

As soon as you are able to log into the switch (i.e., once A1 is flashing) perform the following sequence:

On the originating set:

- Ensure that a voice path has been maintained by blowing into phone on originating set and listening for the blowing on the terminating set

On the terminating set:

- Ensure that a voice path has been maintained by blowing into phone on terminating set and listening for the blowing on the originating set

Leave both handsets offhook

Once the SWCT101 log is issued (i.e., SWACT is done and dial tone has been re-established) perform the following sequence:

On the originating set:

- Ensure that a voice path has been maintained by blowing into phone on originating set and listening for the blowing on the terminating set

On the terminating set:

- Ensure that a voice path has been maintained by blowing into phone on terminating set and listening for the blowing on the originating set

Place both handsets on hook (i.e., terminate call).

Note: Any activation of a feature on a call maintained over CC WarmSWACT will cause the call to be dropped (e.g., a call maintained over CC WarmSWACT cannot activate a conference call, cannot activate call forwarding, cannot come out of hands-free mode, cannot be put on hold etc. without causing the call to be torn down).

4. Ensuring call processing—Re-establish call as described in Step 1.

If at any time during this procedure any of the following conditions exist: one-way speech path, no dial tone, no speech path, constant ringing, no ringing, crosstalk, busy signal—perform the following actions:

a. Check hardware involved for faults (e.g., check set, line card, ring generator, etc.).

b. Post line or trunk at MAP position and confirm proper state or transition of state is set (e.g., if supposed to be in talking mode ensure both the originating and terminating set show CPB, when you put handset onhook the state should change from CPB to IDL).

c. Obtain a QDN for both the originating and terminating set.

d. Obtain a TRAVER for the call between the originating and terminating set.

e. Collect SWCT, ENET, NET, NETM, ENCP, PM, TRK and LINE logs from both sides of the switch (i.e., both active and inactive CPU).

Sample call scripts for testing call survivability

1. Verify ISDN calls:

line (KSET-Disp M5317T) -> trunk -> line (KSET-Disp M5209T)

(e.g., 968-xxxx ---> 6-456-xxxx)

line (PPHONE-Disp M5317T) -> trunk -> line (1FR-Disp Maestral)

(e.g., 968-xxxx ---> 9-969-xxxx)

line (KSET-Disp M5317T) -> trunk -> line (PPHONE-Meridan Bus.)

(e.g., 968-xxxx ---> 9-1-819-456-xxxx)

line (KSET-Disp M5317T) -> line (KSET-PSET)

(e.g., 968-xxxx ---> 968-xxxx)

line (BRAMFT set) -> line (BRAFS set)

(e.g., 968-xxxx ---> 968-xxxx)

line (1FR) -> line (BRAKS set)

(e.g., 968-xxxx ---> 968-xxxx)

---

2. Verify regular POTS calls:

line (PPHONE) -> trunk -> line (1FR)

(e.g., 969-xxxx ---> 9-1-514-970-xxxx)

line (1FR) -> line (1FR)

(e.g., 969-xxxx ---> 969-xxxx)

---

3. Verify CMS calls:

line (1FR) -> line (1FR)

(e.g., 969-xxxx ---> 969-xxxx)

---

4. Verify use of different trunk types:

line (1FR) -> PTS trunk -> line (1FR)

(e.g., 968-xxxx ---> 9-969-xxxx)

line (1FR) -> ISUP trunk (all variants supported by office)

-> line (KSET-Disp M5209T)

(e.g., 968-xxxx ---> 6-456-xxxx)

---

5. Verify use of different PM types:

line (LM) -> line (RLM)

(e.g., 969-xxxx ---> 969-xxxx)

line (LCM) -> line (RLCM)

(e.g., 969-xxxx ---> 969-xxxx)

Procedure for Loading the BMMI Data Dictionary

Loading the BMMI DD from SLM tape

1 Upon receiving the SLM tape containing the BMMI Data Dictionary, the craftsperson should INSERT and LIST the tape.

The SLM tape will have four files associated with it:

```
csp04-1.F.fvocab
csp04-1.F.evocab
csp04-1.F.pbook
bmmiloadfile
```

---

2 Once the SLM tape has been listed, the craftsperson just has to enter the following command:

```
> EXECUTE BMMILOADFILE
```

After a short time period, the text on the MAPCI terminal will appear in French.

Supplemental BMMI DD procedures

The actual BMMI Data Dictionary (DD) for the CDN004 release is comprised of three files with the following naming convention:

csp<XX>-<N>.f.<filetype>

where:

“csp”is communications software platform

<XX> is the csp number

<N> is a BMMI DD sequence number

“f” is signifies that this is a French BMMI DD

<filetype> will be one of fvocab, evocab, or pbook.

So when listing the contents of the SLM tape, you will likely see something like this:

csp04-1.F.fvocab
csp04-1.F.evocab
csp04-1.F.pbook

There is a fourth file called “bmmiloadfile” that contains the actual commands to load the BMMI DD. This file is provided in order to simplify the operation of loading of the BMMI DD for the craftsperson. The typical contents of file “bmmiloadfile” are:

```
bmmi
baseload csp04-1.F.fvocab french
baseload csp04-1.F.evocab french
baseload csp04-1.F.pbook french
ispeak french
```

Note: Each bmmiloadfile may look slightly different depending on the sequence number and the “csp” number of the generated DD files.

Manual loading

- To *manually load* the BMMI DD files without using the bmmiloadfile, you must first enter the “bmmi” utility -

  ```
  > BMMI
  BMMI:
  ```

  Now you are ready to load the actual files using the “baseload” command -

  ```
  > BASELOAD <filename> FRENCH
  ```

  so, for example, we would enter -

  > baseload csp04-1.F.fvocab french

  > baseload csp04-1.F.evocab french

  > baseload csp04-1.F.pbook french

  While still in the “bmmi” utility, activate the French BMMI functionality by using the “ispeak” command -

  ```
  BMMI:
  ```

  ```
  > ISPEAK FRENCH
  ```

  Or, to go back to English enter -

  ```
  > ISPEAK ENGLISH
  ```

  Note: This will only affect the terminal that you are using.

Unloading

- Some time it may be necessary to *unload* the DD files. This is accomplished with the “baseunload” command.

  You must first enter the “bmmi” utility -

  ```
  > BMMI
  BMMI:
  ```

  Now you can unload the files using the command -

```
> BASEUNLOAD <DD_filetype>
```

where the only acceptable filetypes are:

“TVOCAB” (refers to the csp04-1.F.fvocab file)

“EVOCAB” (refers to the csp04-1.F.evocab file)

“PHRASEBOOK” (refers to the csp04-1.F.pbook file)

For example -

> baseunload TVOCAB

> baseunload EVOCAB

> baseunload PHRASEBOOK

Note: The “T” in “TVOCAB” is not a spelling error, rather it stands for Translated VOCAB.

Appendix C:Test Call Plan

Appendix C table of contents

This section presents generic guidelines for creating a test call plan to be used both *before and after* CC SWACT.

The purpose of test calls is to verify the performance of newly installed software for telephone switching systems. Test calls can ensure the smooth operation of thousands of calling situations with different combinations of telephone sets, service features, and traffic conditions on the network. Such testing helps ensure the availability and reliability of features and services for telephone users.

The term “Test Call Scripts” refers to the verification calls as predefined by the Telephone/Carrier Operating Company. These are test calls to be performed after activating the new software load in order to confirm acceptability of the new load. In the ONP MOP the Test Call Scripts are put to use as follows:

1. As a part of site preparation, the procedure *Fill in Test Call Scripts* instructs Telco to “Fill in and test the Test Call Scripts.” This is to provide a thorough test plan exercise for validating the new software load.
2. Then, after activating the new software load, the procedure *Do Test Calls* instructs Telco to “Perform TEST CALLS that were identified ahead-of-time.”

The call scripts provided below are only examples of call types that could be included in the Test Call Scripts. These samples include only basic call processing tests and provide some, but not all, critical test calls. These are provided only as a guideline. Each telephone/carrier office should determine the best mix of test calls to use based on the office’s unique configuration.

CAUTION

The test calls listed in the following procedures will not apply to every office

The telephone office will have to customize the list according to its own particular configuration.

CHECKLIST

Nortel recommends including the following items in your POSTSWACT testing routine.

Add any other items that are determined to be important.

POSTSWACT call checklist

___ Verify date, time and DIALTONE

___ Perform Critical Call Tests (for example, 0- and 0+7/10)
See the following procedure.

___ Perform IDDD (International Direct Distance Dialing) calls

___ Check Equal Access origination and termination

___ Perform CCIS (Common Channel Inter-office Signaling) calls

___ Verify ACTS (Automatic Coin Toll Service)

___ Verify DRAMS (announcements and SIT tones)

___ Check WATS (Wide Area Telephone Service), INWATS, OUTWATS, 2-way

___ Verify Pay Station Coin Control (Coin Collect and Coin Return)

___ Verify miscellaneous services such as 311, 411, 611, and repair services

___ Check EAS (Extended Area Service) calls

___ Perform TOPS (Traffic Operator Position System) calls

___ Perform MCCS (Mechanized Calling Card Service) calls

___ Perform DISA (Direct Inward System Access) calls

___ Verify Custom Calling Features

Procedure C-1
Critical test calls

	Test description	From: Line type or CLLI	To: Call type Digits dialed	Test Result
1	Check for DIALTONE on all line modules			
2	Verify '0' minus route			
3	Verify '0' plus route			
4	Verify ONI 1-7, 1-10 digits			
5	Verify EAS incoming/outgoing routes			
6	Verify CAMA routes			
7	Verify local Tandem routes			
8	Verify DDO route			
9	Verify Directory Assistance			
10	Verify critical service routes (911, police, file, hospitals, and radio stations)			
11	Verify TOLL COMP (DTS) route			
12	Verify 1FR intra-office call			
13	Verify Remotes: - EAS outgoing (one route) - '0' plus '0' minus - CAMA ANI - 1FR intra- Remote			
14	Verify Operator Intercept route			
15	Verify all tones, group alarms, and announcements functional			
16	Verify all (idle) customer/network Trunks			

Procedure C-2
Additional test calls sample

1 Verify regular POTS calls:

line (1FR) –> line (1FR)

(969-xxxx –> 969-xxxx)

line (PPHONE) –> trunk –> line (1FR)

(969-xxxx –> 9-1-514-970-xxxx)

---

2 Verify use of different PM types:

line (LM) –> line (RLM)

(969-xxxx –> 969-xxxx)

line (LCM) –> line (RLCM)

(969-xxxx –> 969-xxxx)

---

3 Verify use of different trunk types:

line (1FR) –> PTS trunk –> line (1FR)

(968-xxxx –> 9-969-xxxx)

line (1FR) –> ISUP trunk (all variants supported by office)–>
line (KSET-Disp M5209T)

(968-xxxx –> 6-456-xxxx)

---

4 Verify ISDN calls:

line (KSET-Disp M5317T) –> trunk –> line (KSET-Disp M5209T)

(968-xxxx –> 6-456-xxxx)

line (PPHONE-Disp M5317T) –> trunk –> line (1FR-Disp Maestro)

(968-xxxx –> 9-969-xxxx)

line (KSET-Disp M5317T) –> trunk –> line (PPHONE-Meridan Bus.)

(968-xxxx –> 9-1-819-456-xxxx)

line (KSET-Disp M5317T) –> line (KSET-PSET)

(968-xxxx –> 968-xxxx)

line (BRAMFT set) –> line (BRAFS set)

(968-xxxx –> 968-xxxx)

line (1FR) –> line (BRAKS set)

(968-xxxx –> 968-xxxx)

DMS–100 Family

Software Delivery

ONP SNSE to SuperNode/ENET
Conversion

Address comments to:
Northern Telecom
Dept. 3550
100 Perimeter Park Drive
Morrisville, North Carolina
U.S.A. 27650

Publication number: 287-8991-304
Document status: Preliminary
Document release: 01.09
Date: February 1998

Printed simultaneously in Canada and the United States of America by Northern Telecom

NORTEL
NORTHERN TELECOM