

297-8991-540

DMS-100 Family

Post-Release Software Manager (PRSM)

Reference Guide

BASE14 and up Standard 10.01 February 2000

DMS-100 Family

Post-Release Software Manager (PRSM)

Reference Guide

Publication number: 297-8991-540

Product release: BASE14 and up

Document release: Standard 10.01

Date: February 2000

Copyright © 1995, 1996, 1997, 1998, 1999, 2000 Nortel Networks,
All Rights Reserved

Published in U.S.A.

NORTEL NETWORKS CONFIDENTIAL: The information contained herein is the property of Nortel Networks and is strictly confidential. Except as expressly authorized in writing by Nortel Networks, the holder shall keep all information contained herein confidential, shall disclose the information only to its employees with a need to know, and shall protect the information, in whole or in part, from disclosure and dissemination to third parties with the same degree of care it uses to protect its own confidential information, but with no less than reasonable care. Except as expressly authorized in writing by Nortel Networks, the holder is granted no rights to use the information contained herein.

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

DMS, MAP, NORTEL, NORTEL NETWORKS, NORTHERN TELECOM, NT, and SUPERNODE are trademarks Northern Telecom.

Publication history

February, 2000

Standard 10.01. BASE14. Clarified information on SPM patching. Added new PRSM301 log reason information.

August, 1999

Standard 09.01. BASE13 Added information on SPM patch file support and services to the following chapters:

- Overview
- Basic manual functions
- Advanced functions
- Automated processes
- Tables
- Command output
- Database fields (Field DROPSYNC is called FIELD SPLIT when running XA-Core)
- Predefined select definitions
- Predefined set definitions
- Logs (added new PRSM303 log and new entry to PRSM470 log)
- Log301 reasons (added new reason code for PRSM301 log)

May, 1999

Standard 07.03. BASE11. Added information on PRSM LFAUDIT..

February, 1999

Standard 07.03. BASE11. Added information on PRSM LFAUDIT..

August, 1998

Standard 07.02. BASE11. Revised PRSM301 log example.

May, 1998

Standard 06.02. BASE10. Revised sample output from REPORT.

February, 1998

Standard 06.01. BASE10. Added information on PRSM Activation Timestamp, feature PR1002, and PRSM Auto Dependency Calculation, feature AN1879. Removed tables on status changes for individual commands in the chapter “Basic manual functions.” Added information on PPXL-related set definitions and select definitions.

October, 1997

Standard 05.02. BASE09. Added example of blocked action to PRSU fields table in the chapter “Database fields.” Changed PPXL information in the chapter “Overview”.

August, 1997

Standard 05.01. BASE09. Added information for ISN Autoimage Support, feature AN1896, PRSM Year 2000 Support, feature AN1897. Added PRSM302 log information. Changed the name of the document from PRSM Operating Procedure to PRSM Reference Manual. Add new section for PRSM interfaces. Added new information from feature AN1895 in reference to SPMs. Updated output examples. Added DCH500 log information.

March, 1997

Standard 04.01. BASE08. Initial publication for BASE08. Added information on unequal PRSUs. Added information on PRSU downloads through the SDM. Added information on ISTBAUDIT command. Revised information on REPORT command. Added information on MATESTAT, MATEMISMATCH and DESTID fields. Revised information on DEST, ONHOLD and HOLD fields. Added information on log PM128 and its support for ISTBAUDIT. Removed information on NN status. Added publication history. Added command examples to the chapters “Basic manual functions” and “Advanced manual functions”.

Contents

About this document	ix
When to use this document	ix
Overview of document	ix
How to check the version and issue of this document	x
References in this document	x
What precautionary messages mean	x
How commands, parameters, and responses are represented	xii
Input prompt (>)	xii
Commands and fixed parameters	xii
Variables	xii
Required parameters and variables	xii
Optional parameters and variables	xii
Special characters	xiii
Responses	xiii
Example	xiii
1 Overview	1-1
Introduction	1-1
Release notes	1-1
PRSM SPM patching	1-1
Terminology	1-1
PRSM	1-1
PRSU	1-1
PRSUID	1-2
DEST	1-2
DESTID	1-3
PRSU file names	1-3
\$PATCH	1-3
\$DF	1-3
\$INC	1-3
Naming conventions	1-3
Non-permanent PRSUs	1-3
Permanent PRSUs	1-4
Special types of PRSUs	1-4
SPM Permanent PRSUs	1-5
Activatable PRSUs	1-5
Installable PRSUs	1-5
Unequal PRSU files	1-5
Special types of DESTs	1-5

- PPXLs 1-6
- SPM loadfile destinations 1-6
- PRSU file search on devices 1-6
 - PRSM CI 1-6
 - Patching after loading 1-7
 - Automated processes 1-7
 - One night process 1-8
- PRSM work space on devices 1-9
- Functions 1-9
 - Manual functions 1-9
 - Automatic functions 1-9
- PRSM Command Interpreter (CI) 1-9
- PRSM automated processes 1-10
 - File Audit process 1-10
 - Autoapply process 1-11
 - Autoinstall process 1-11
 - Status Audit process 1-11
- PRSM prompts 1-11
- PRSM statuses 1-12
- PRSM sets 1-12
 - Simple set definitions (PRSUs or DESTs) 1-13
 - Complex set definitions (PRSUs and DESTs) 1-14
- PRSM field conditions 1-14
 - Special characters 1-15
 - PRSM set syntax 1-16
- Logs 1-19
 - PRSM logs 1-19
 - DCH logs 1-19
 - PM logs 1-20
- Tables 1-21
 - PRSM tables 1-21
 - Related tables 1-21
- User controlled options 1-22
 - PRSM responses 1-22

2 Basic manual functions 2-1

- Validating, applying, and removing patches 2-1
 - VALIDATE command 2-1
 - APPLY command 2-5
 - REMOVE command 2-8
- Displaying information 2-11
 - DISADMIN command 2-11
 - REPORT command 2-12
 - HELP command 2-16

3 Advanced manual functions 3-1

- Using set and select definitions 3-1
 - SETDEF command 3-1
 - SELDEF command 3-5
 - PRSMSET command 3-7

- Generating custom reports 3-11
 - SELECT command and PRSM reports 3-11
- Managing the PRSM database 3-14
 - ASSIGN command 3-14
 - DBAUDIT command 3-15
 - LFAUDIT command 3-17
- Manually managing automated processes 3-18
 - AUTOPROC command 3-18
 - FILEAUDIT command 3-20
- Other commands 3-20
 - FREEMEM command 3-20
 - ISTBAUDIT command 3-22
- Controlling the PRSM session 3-23
 - PRSM ABORT 3-23
 - PRSM RECONNECT 3-23

4 Automated processes 4-1

- Introduction 4-1
- File Audit process 4-3
 - Finding new PRSUs 4-3
 - Revalidating PRSUs in an intermediate state 4-4
 - Verifying database entries 4-4
 - Finding SPM and XPM PRSU files 4-4
- Autoapply process 4-5
 - Log monitoring 4-8
- Autoinstall process 4-10
- Status Audit process 4-10
 - Status Audit DBAUDIT 4-11
 - Status Audit EXT alarms 4-11

5 Tables 5-1

- Table PRSMOPTS 5-1
- Table PRSMALRM 5-2
 - Predefined alarms 5-2
- Table AUTOOPTS 5-4
- Table AUTOMON 5-5
- Table AUTOPRSU 5-6
- Table PADNDEV 5-7

6 Command output 6-1

- Introduction 6-1
- QUIET response format 6-2
- BRIEF response format 6-2
- VERBOSE response format 6-2

7 Statuses 7-1

- Command status changes 7-1
- Peripheral state required for changing PRSU status 7-2
- PRSU flags 7-3
 - AOR flag 7-3

	Placing a PRSU and DEST on hold	7-4
	New host software release	7-5
	Pre-patched XPM loads	7-5
	PPXLs and ONP	7-8
	SPM Destinations	7-8
	SPM Loadfile Destinations	7-8
	SPM Peripheral Destinations	7-8
8	Database fields	8-1
	PRSU fields	8-1
	DEST fields	8-9
9	Predefined set definitions	9-1
	Introduction	9-1
	PRSU set definitions	9-1
	DEST set definitions	9-7
10	Logs	10-1
	Trouble logs	10-1
	PRSM300—General trouble	10-2
	PRSM301—PRSM file failure	10-2
	PRSM302—File management failure	10-3
	PRSM303—PRSM PRSU discrepancy	10-3
	PRSM360—File audit failure	10-4
	PRSM380—Automated process failure	10-4
	PRSM381—Switch sanity failure	10-4
	Service summary logs	10-4
	PRSM400—PRSM action summary	10-5
	PRSM401—Assignment summary	10-5
	PRSM470—Status audit summary	10-6
	Information logs	10-7
	PRSM600—General information	10-7
	PRSM603—PRSMOPTS option change	10-7
	PRSM680—Automated process information	10-8
	PRSM681—AUTOPROCESS CI command	10-8
	PRSM682—AUTOOPTS option change	10-8
	PRSM683—AUTOPRSU option change	10-8
11	PRSM301 log reason interpretation	11-1
	PRSU errors	11-1
12	Interface to PRSM	12-1
	Introduction	12-1
	ISN Autoimage support	12-1

About this document

When to use this document

Use this document as a reference for Post-Release Software Manager (PRSM), the patching utility that replaced PATCHER. This document describes the functionality, commands, processes, responses, and fields used by PRSM. This document may be used by operating company personnel with a range of patching and maintenance experience.

Overview of document

Following is a description of each chapter in this document:

- "Overview" provides a functional overview of PRSM, including terminology, concepts, and syntax.
- "Basic manual functions" describes the commands used to perform basic PRSM operations.
- "Advanced manual functions" describes the commands used to perform advanced PRSM operations.
- "Automated processes" describes the maintenance functions automated by PRSM.
- "Tables" describes the datafillable tables.
- "Command output" shows the formats of command output available through PRSM.
- "Statuses" describes the commands and events that can cause discrepancies and change the status of PRSUs.
- "Database fields" describes the selectable fields used by PRSM in the PRSU and DEST databases.
- "About this document" describes the select definitions predefined by PRSM.
- "Predefined set definitions" describes the set definitions predefined by PRSM.
- "Interface to PRSM" describe external systems that interface with PRSM.

How to check the version and issue of this document

The version and issue of the document are indicated by numbers, for example, 01.01.

The first two digits indicate the version. The version number increases each time the document is updated to support a new software release. For example, the first release of a document is 01.01. In the *next* software release cycle, the first release of the same document is 02.01.

The second two digits indicate the issue. The issue number increases each time the document is revised but rereleased in the *same* software release cycle. For example, the second release of a document in the same software release cycle is 01.02.

To determine which version of this document applies to the software in your office and how documentation for your product is organized, check the release information in the *DMS-10 and DMS-100 Family Product Documentation Directory*, 297-8991-001.

This document is written for all DMS-100 Family offices. More than one version of this document may exist. To determine whether you have the latest version of this document and how documentation for your product is organized, check the release information in the *DMS-10 and DMS-100 Family Product Documentation Directory*, 297-8991-001.

References in this document

The following documents are referred to in this document:

- *DMS-100 Family PRSM Basic Reference Guide*, 297-8991-541
- *DMS-100 Family PRSM Advanced Reference Guide*, 297-8991-542
- *Feature Description Manual*, 297-8001-801
- *North American DMS-Log Report Reference Manual*, 297-8001-840
- *One Night Process Software Delivery Procedures*,
- *Peripheral Module Software Release Document*,
- *Product Documentation Directory*, 297-8991-001
- *Spectrum Peripheral Module Release Document*, 297-1771-598

What precautionary messages mean

The types of precautionary messages used in Nortel Networks documents include attention boxes and danger, warning, and caution messages.

An attention box identifies information that is necessary for the proper performance of a procedure or task or the correct interpretation of information or data. Danger, warning, and caution messages indicate possible risks.

Examples of the precautionary messages follow.

ATTENTION

Information needed to perform a task

If the unused DS-3 ports are not deprovisioned before a DS-1/VT Mapper is installed, the DS-1 traffic will not be carried through the DS-1/VT Mapper, even though the DS-1/VT Mapper is properly provisioned.

DANGER

Possibility of personal injury



DANGER

Risk of electrocution

Do not open the front panel of the inverter unless fuses F1, F2, and F3 have been removed. The inverter contains high-voltage lines. Until the fuses are removed, the high-voltage lines are active, and you risk being electrocuted.

WARNING

Possibility of equipment damage



DANGER

Damage to the backplane connector pins

Align the card before seating it, to avoid bending the backplane connector pins. Use light thumb pressure to align the card with the connectors. Next, use the levers on the card to seat the card into the connectors.

CAUTION

Possibility of service interruption or degradation

**CAUTION****Possible loss of service**

Before continuing, confirm that you are removing the card from the inactive unit of the peripheral module.

Subscriber service will be lost if you remove a card from the active unit.

How commands, parameters, and responses are represented

Commands, parameters, and responses in this document conform to the following conventions.

Input prompt (>)

An input prompt (>) indicates that the information that follows is a command:

```
>BSY
```

Commands and fixed parameters

Commands and fixed parameters that are entered at a MAP terminal are shown in uppercase letters:

```
>BSY INACTIVE
```

Variables

Variables are shown in lowercase letters:

```
>BSY pm_no
```

The letters or numbers that the variable represents must be entered. Each variable is explained in a list that follows the command string.

Required parameters and variables

Required parameters and variables are shown within < > brackets.

```
>LOADPM UNIT <unit_no>
```

```
>APPLY <prsu_set_def>
```

Optional parameters and variables

Optional parameters and variables are shown within [] brackets.

```
>LOADPM UNIT <unit_no> [NOWAIT]
```

```
>REMOVE <prsu_set_def> [<IN dest_set_def>]
```

Special characters

The following table lists the special characters used in PRSM commands in this document.

Table 1 Special characters

Character	Description	Usage
'	Single quote	Introduces a special character within a PRSM command line
=	Equal	A field is equal to a specified value
<>	Not equal	A field is not equal to a specified value
>	Greater than	A field is greater than a specified value
<	Less than	A field is less than a specified value
>=	Greater than or equal to	A field is greater than or equal to a specified value
<=	Less than or equal to	A field is less than or equal to a specified value
&	And	The common elements between the sets on both sides of the &
	or	All the elements of the sets on both sides of the
*	Wild card	Wild card
+	Plus sign	Command continued to next line

Responses

MAP responses that correspond to MAP commands are shown in a different type style.

```
LOADPM UNIT 1 LOADED
```

The following excerpt from a procedure shows the command syntax used in this document:

Example

The following excerpt from a procedure shows the command syntax used in this document:

- 1 Validate the PRSU in the PM unit. Type

>VALIDATE prsuid

and press the Enter key.

where:

prsuid

is the name of the PRSU

Example of command:

>VALIDATE SCB01CB5

Example of MAP response:

File SCB01CB5\$PATCH for PRSU SCB01CB5 found on device
SFDEV.

No Special Application Instructions

No Restart Required

This PRSU is autoapplicable

This PRSU is NOT activatable

SCB01CB5 validates against 1 DEST

PRSM Database admin info updated

1 Overview

This chapter provides a functional overview of Post-Release Software Manager (PRSM). It describes terminology, types of post release software updates (PRSU), manual functions, automatic processes, set definitions, and field conditions.

Introduction

PRSM is a DMS utility that provides manual and automated processes to apply software updates distributed after a milestone load release.

Release notes

PRSM SPM patching

This section describes changes associated with Spectrum Peripheral Module (SPM) patching. For information on the SPM, see the *DMS-100 Family Spectrum Peripheral Module Release Document, 297-1771-598*.

Terminology

PRSM and this document introduce the following terms.

PRSM

Post-Release Software Manager (PRSM) adds and tracks the software updates distributed after a milestone load release.

PRSU

A post-release software update (PRSU) represents a software change necessary for normal maintenance of a DMS-100 Family office. A PRSU is distributed after a milestone load release. Examples of PRSUs include a patch, a permanent PRSU, and an increment.

Patch

A patch is a post release software change for DMS products. A patch is delivered as a separate file from load files. It may be applied to and removed from a destination.

Permanent PRSU

A permanent PRSU is a post-release software change that is a permanent part of a load. It is built into the load at Nortel and cannot be removed from the load. An example of this type of PRSU is a fix delivered as part of an SPM maintenance load.

Increment

An increment is a post release software change for Generic Services Framework (GSF) code. An increment is used only for the object-oriented code used by GSF. An increment can provide either corrective content or new functionality

A PRSU file is software created for one of the following purposes:

- a procedure replacement to correct software deficiencies delivered to all affected sites,
- an enhancement to the original design delivered to all affected sites,
- a data corruption software fix delivered only to the affected office, or
- a delivery mechanism for early feature deployment that contains new features and is activated on a per site basis by a controlled password.

PRSUID

A prsuid is the name of a PRSU.

DEST

DEST stands for destination. A destination is an object that can contain PRSUs. Examples of DESTS include a peripheral unit or a loadfile.

For instance, LTC 0 0 is a destination, LTC 0 1 is a destination, and each of the following nodes is a destination:

- LTC 0 0
- LTC 0 1
- MS 0
- MS 1
- LIM 0 0
- LIU7 15
- ESA 5
- SPM 0 CEM 0
- SPMLOAD CEM013A_010028
- PPXL ESI07BI1_970119

DESTID

A destid is the name of a DEST.

PRSU file names

A PRSU file name consists of a base file name and an extension. The base file name identifies the targeted processor, and the release. The extension identifies the type of PRSU file. Following are the types of PRSU extensions.

\$PATCH

The \$PATCH file is commonly known as a patch file. The \$PATCH file is the SOS file associated with the prsuid.

\$DF

The \$DF file contains a subset of the information contained in a \$PATCH file. This subset of information can be used to determine if a PRSU can be applied in an office.

Note: By creating and validating a \$DF file, it can be determined if the \$PATCH file is appropriate for the office.

\$INC

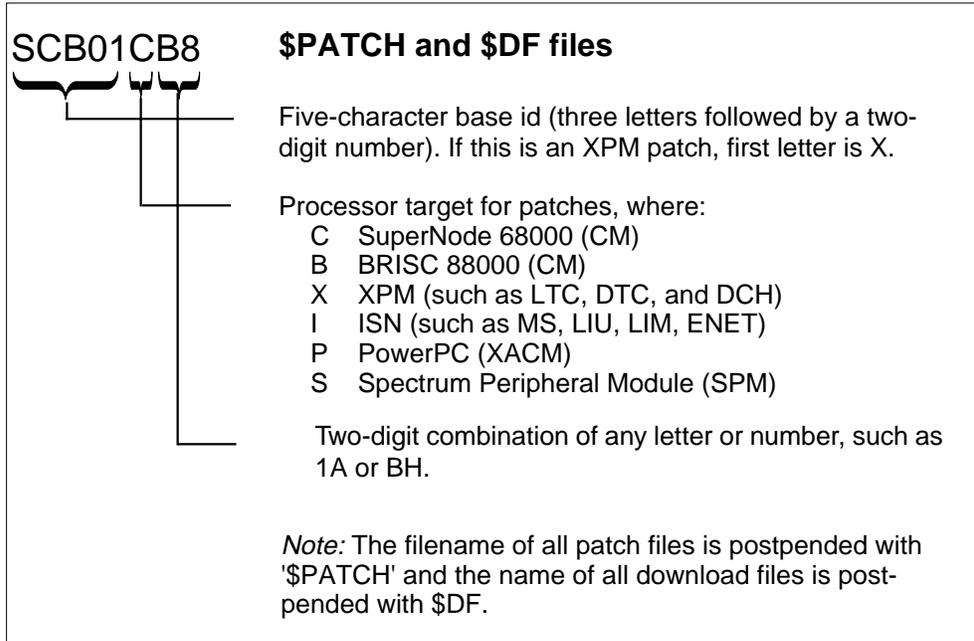
A \$INC file is an increment file. The \$INC file is available only in offices with GSF.

Note: \$DF files are not used with increments.

Naming conventions**Non-permanent PRSUs**

The following figure illustrates the naming conventions for non-permanent PRSUs.

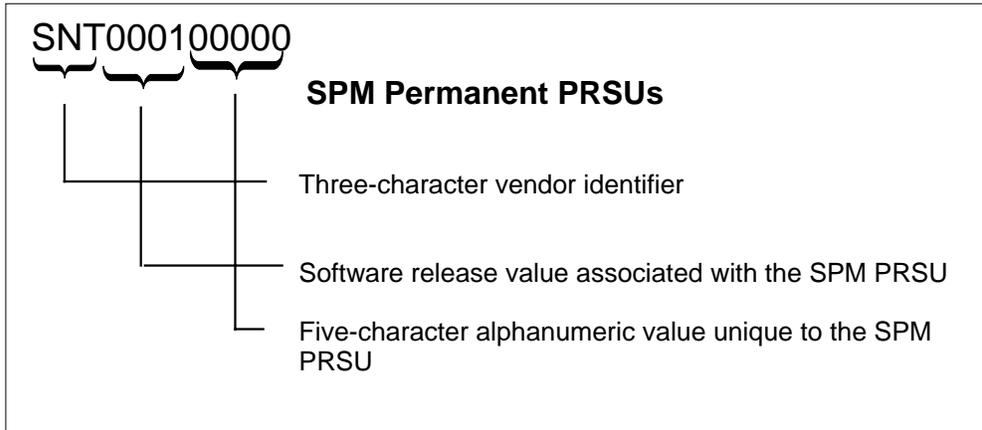
Figure 1-1 Naming conventions for Non-permanent PRSUs



Permanent PRSUs

The following figure illustrates the naming convention for permanent PRSUs.

Figure 1-2 Naming convention for Permanent PRSUs



Note: An example of a permanent PRSU is a fix delivered as part of an SPM maintenance load.

Special types of PRSUs

This section describes special types of PRSUs.

SPM Permanent PRSUs

An SPM permanent PRSU contains a fix delivered as part of an SPM maintenance load. PRSM supports the tracking of SPM destinations and PRSUs. PRSM recognizes an SPM destination after the DBAUDIT command is performed on the SPM loadfile.

Activatable PRSUs

An activatable PRSU is a PRSU in which the code does not take effect until the PRSU is applied and activated.

A PRSU is activated or deactivated by using the ASSIGN command to change the value of the ACTIVE field. If the field is set to Y, for yes, the PRSU is activated. If the field is set to N, for no, the PRSU is not activated.

When an APPLY or REMOVE is performed on an activatable PRSU, PRSM informs the user that the PRSU can be activated. A PRSU must be deactivated before it can be removed.

Two predefined selects list information on activatable PRSUs: NOTACTIVATED and ACTIVATED. These predefined reports print out the PRSUs that have not been activated in a DMS office and the PRSUs that have been activated, respectively.

Installable PRSUs

An installable PRSU is a PRSU in which the code does not take effect until the PRSU is applied and installed. Presently, only increments are installable.

Field INSTREQD in the PRSM database identifies whether a PRSU can be installed. Use the SELECT or DISADMIN command to display the field.

If installation is not required, the increment becomes active when applied. If installation is required, use the ASSIGN command to change the value to Y, yes, and the increment will be installed.

Unequal PRSU files

PRSU files may be equal or unequal. Equal PRSU files are identical PRSU files with the same name. Unequal PRSU files are PRSUs with the same name but different content. PRSM will not operate on unequal PRSUs because it cannot determine which PRSU file to use.

Note: Refer to "Automated processes" on page 1-7 in this document for further information on unequal PRSU files.

Special types of DESTs

This section describes special types of DESTs.

PPXLs

Pre-patched XPM loads (PPXLs) have source code XPM PRSUs built into the loads. PPXL loadfiles and their PRSU content must first be datafilled in PRSM before they can be entered as an active or backup loadfile in table PMLOADS. This PRSM datafill is accomplished through the LFAUDIT command. Through this command, XPM PRSUs that are built into the PPXL are datafilled in the PRSM database if they were not already known by PRSM. If the PRSU files are available, the PRSUs are set to A (applied) against the PPXL loadfile destination.

Following are some advantages of PPXLs over traditional XPM loads.

- Patching after loading (PAL) is faster, because fewer PRSUs must be applied as part of the patching step. PPXLs do not reduce the number of PRSUs for a given load. PPXLs reduce the number of PRSUs applied manually to the load.
- Fixes are provided unconditionally before static data is downloaded. PAL methods vary in the order that PRSUs and static data are downloaded.

SPM loadfile destinations

When an SPM loadfile is datafilled in table PMLOADS, a destination is added to the PRSM database with the format SPMLOAD . The SPM PRSUs associated with the SPM loadfile are datafilled in the PRSM database during the DBAUDIT.

The DBAUDIT command can be issued manually on all destinations or on separate destinations in an office. Also, DBAUDIT updates load files with the PRSUs that transfer during one-night process (ONP). This action resolves discrepancies between the load file and the PRSM database, causing the PRSUs to be set to A (applied) status against that load file destination.

PRSU file search on devices

In the DMS switch, a storage area is defined as a device. For example, disk drive units (DDU) and system load module (SLM) volumes are both defined as devices. The switch searches certain devices when searching for PRSU files. The sequence in which devices are searched depends upon which process is searching for the PRSUs.

PRSM CI

When searching for PRSUs, PRSM CI processes search the following devices in the following order:

1. user directory
2. file identifier stored in the PRSM database

Patching after loading

When searching for PRSUs, patching after loading (PAL) processes search the following devices in the following order:

1. file identifier stored in the PRSM database
2. all devices datafilled in table PADNDEV, except SFDEV
3. SFDEV
4. the active device for the load, as datafilled in field ACTVOL in table PMLOADS

Note: SPM destinations do not support PAL.

Automated processes

The automated processes of PRSM run in the following order:

1. File Audit
2. Autoapply
3. Autoinstall, in switches with GSF
4. Status Audit

Every DMS switch employs three or more automated processes. Every switch supports the following:

- File Audit process
- Autoapply process
- Status Audit process

The File Audit process always does a search, and one part of the Status Audit process may do a search, as described later in this chapter under the heading "Status Audit." Refer to chapter 4, "Automated processes" in this document for a more detailed description of PRSM's automated processes.

File Audit

The File Audit process always searches for PRSUs. The validation portion of the process searches the following devices in the following order:

1. SFDEV
2. devices datafilled in table PADNDEV, except SFDEV

SPM and XPM Patches. The PRSM File Audit process also verifies that any SPM or XPM patch marked as apply on reload (AOR) = Y (in the PRSM database) can be found. It checks the file identifiers of SPM and XPM patches stored in the database. When searching for SPM and XPM patches, the File Audit also searches devices datafilled as ACTVOL in table PMLOADS.

The File Audit does not validate PRSU files found on the PMLOADS devices.

Autoapply

The Autoapply process does not search for PRSUs.

Autoinstall

The Autoinstall process does not search for PRSUs.

Status Audit

When searching for PRSUs, the DBAUDIT portion of the Status Audit process searches the following devices in the following order:

1. SFDEV
2. devices datafilled in table PADNDEV, except SFDEV

One night process

There are three parts of one night process (ONP) that use a search order.

1. Determining which computing module (CM) PRSUs are to be applied to the new side of the CM
2. Applying CM PRSUs to the new side of the CM
3. Updating the PRSM database (on the new side) with PM PRSU information

To determine which CM PRSUs are to be applied on the new side of the CM, the CM gets a list of PRSUs that are currently applied on the new side and compares that list with the PRSUs that are applicable for the new side.

The CM looks for PRSUs, for the new side, in the following places.

- the devices specified in table PADNDEV
- SFDEV

The CM sends the PRSUs from the old side to SFDEV on the new side. While the new side is INACTIVE, PRSM applies the PRSUs that were put in SFDEV.

When the new side has been patched and is active during the Matchall step of a postswact, the PRSM database is updated with PM PRSU information. PRSM searches all devices in that DMS switch for PRSUs that might be found in the PMs. DBAUDIT searches for PRSUs in SFDEV, on devices that are specified in table PADNDEV, on SLM disks, and on 9-track tape drives.

Note: The MATCHALL step performs a DBAUDIT, not a match.

PRSM work space on devices

PRSM has several needs for temporary file space. When PRSU files are downloaded to the DMS, disk space is required to store the files. Some commands in PRSM need disk space as temporary work space. Whenever PRSM needs disk space, PRSM tries to locate temporary file space in SFDEV.

Functions

PRSM functionality is controlled either manually or automatically. This allows the user to choose between complete automation and manual intervention.

Manual functions

The manual functionality in PRSM includes basic and advanced functionality. Basic manual functionality allows a user to perform the following functions:

- validate a PRSU - determines whether a PRSU will apply in a destination in a DMS office
- apply a PRSU - causes software in a DEST to be modified based on information contained within the PRSU
- remove a PRSU - removes software modifications applied to the DEST
- display a PRSU - displays all copies of the PRSU file known by PRSM
- obtain HELP on PRSM - displays HELP information

Refer to the chapter 2, "Basic manual functions" and chapter 3, "Advanced manual functions" for more information on PRSM manual functionality.

Automatic functions

In addition to being able to execute certain functions manually, PRSM schedules the following processes that run automatically.

- Validate new (previously downloaded) PRSU files.
- Audit PRSU files on the DMS.
- Apply PRSUs.
- Install installable PRSUs.
- Audit the statuses of PRSUs on the DMS.

PRSM Command Interpreter (CI)

PRSM CI provides the ability to perform PRSM functions manually. Commands and functions performed from the PRSM CI are as follows:

- VALIDATE
- APPLY
- REMOVE

- DISADMIN
- REPORT
- HELP
- SELECT
- SETDEF
- SELDEF
- PRSMSET
- ASSIGN
- DBAUDIT
- LFAUDIT
- AUTOPROC
- FILEAUDIT
- FREEMEM
- ISTBAUDIT

PRSM automated processes

PRSM automates certain PRSU maintenance functions through the following processes:

- File Audit
- Autoapply
- Autoinstall, in offices with GSF
- Status Audit

These automated processes can be executed manually by the AUTOPROC command or scheduled through the data tables.

Refer to chapter 4, "Automated processes" in this document for information on automated processes.

File Audit process

The File Audit (FILEAUD) process runs prior to the AUTOAPP process. The File Audit searches SFDEV and the devices datafilled in table PADNDEV for new PRSUs and validates them. It also revalidates any PRSUs that are at an intermediate state and searches for unequal PRSU files. A special check is done for missing SPM and XPM PRSU files.

If the FILEALARM field is set to Y, the SPM, XPM, and unequal PRSU file information is reported in the PRSM360 log.

Note: A FILEAUDIT value of N against a PRSU prevents an alarm or a log from being generated by the File Audit, even if the PRSU meets the alarmable condition.

Autoapply process

The Autoapply process (AUTOAPP) can be used to automatically apply certain PRSUs. PRSUs which meet the conditions described in chapter 4, "Automated processes" in this document are eligible to be automatically applied by the Autoapply process. This process runs after the PRSM File Audit on the days specified by the MTWTFSS field in table AUTOPRSU.

Autoinstall process

The Autoinstall (AUTOINST) process is used with GSF loads to install PRSU increments that require installation and are installable, based on values in fields INSTREQD and AUTOINST in the PRSM database. Not all increments need to be installed, and any increments installed with the Autoinstall process must first be applied with the APPLY command or the Autoapply process.

Status Audit process

The Status Audit (STATAUD) process runs after the Autoapply process. It ensures PRSM database integrity by running a DBAUDIT which compares the applied PRSU information in the PRSM database and the DESTs. After the DBAUDIT, the Status Audit process searches for any PRSUs which are in alarmable conditions. The alarmable conditions are defined by the datafill in table PRSMALRM. The STATALARM field is set to Y for PRSUs that meet alarmable conditions. The results of the Status Audit are reported in the PRSM470 log.

Note: A STATAUDIT value of N against a PRSU prevents an alarm or a log from being generated by the File Audit even if the PRSU meets the alarmable condition.

PRSM prompts

PRSM sometimes requires confirmation of a command or action from the user. All PRSM prompts can be answered Y or YES for a positive response, or N or NO for a negative response. These responses are not case sensitive.

PRSM statuses

The status of a PRSU is tracked on a per-destination basis. An overall status is not maintained in PRSM. Instead, PRSM maintains the status of each PRSU in every destination. The following table lists the possible PRSU statuses.

Table 1-1 PRSU statuses

Status ID	Definition	Explanation
NV	Needs validation	The PRSU needs to be validated in a DEST. The PRSU may be valid in the DEST, but the PRSU has not been validated against the DEST. A PRSU is at this status until it has been validated against a destination.
VA	Validated for application	The PRSU can apply in a destination. Note: PRSUs can then be applied by APPLYing the PRSU in PRSM or by loading the DEST with a load that contains the PRSU.
A	Applied	The PRSU is currently applied in a destination. Note: PRSUs can then be applied by APPLYing the PRSU in PRSM or by loading the DEST with a load that contains the PRSU.
R	Removed	The PRSU has been removed from a destination.
NN	Not needed	The PRSU is not needed in the DEST. This status is only seen with the SPM and XPM PRSUs associated with SPM loadfile DESTs and PPXL DESTs.

PRSM sets

PRSM operates on two types of entities: PRSUs and DESTs

PRSM sets allow maintenance functions and reporting to be performed on groups of PRSUs and DESTs. The user can group PRSUs or DESTs into a set and manipulate them. PRSM sets allow a user to perform an operation on a single item or on a large group of items with one command. For example, after a set of PRSUs has been created, the set of PRSUs can be applied in all applicable DESTs. PRSM sets also simplify queries on information stored in the PRSM database.

Set definitions can be broadly divided into two categories: simple and complex.

Simple set definitions (PRSUs or DESTs)

A simple set definition requires PRSUs or DESTs. Following are three types of simple set definitions:

- single element
- named set of specific elements
- set of elements with specific qualities

Single element

A simple set definition may consist of a single PRSU or DEST. Following is an example of a set definition for PRSU sam00ib5.

```
sam00ib5
```

Following is an example of a set definition for DEST MS 0.

```
ms 0
```

Named set of specific elements

A simple set definition may consist of a named set of specific PRSUs or DESTs. Following is an example of a set definition for the set samprsubset, which consists of PRSUs sam00ib5 and sam01ib5.

```
>prsmset create 'prsubset sam00ib5 1 sam01ib5 name samprsubset
```

```
SAMPRSASET created.
```

```
>prsmset query samprsubset
```

```
SAM00IB5 SAM01IB5
```

Following is an example of a set definition for the set samdestset, which consists of DESTs MS 0 and MS 1.

```
>prsmset create 'destset ms 0 1 ms 1 name samdestset
```

```
SAMDESTSET created.
```

```
>prsmset query samdestset
```

```
MS 0
```

```
MS 1
```

Set of elements with specific qualities

A simple set definition may consist of a set of PRSUs or DESTs with specific qualities. Following is an example of a set definition for a set of general category PRSUs with a class of change.

```
'prsus WHERE category = gen class = c
```

Following is an example of a set definition for a set of DESTs with loadname msc05ao.

```
'dests WHERE loadname = msc05ao
```

Complex set definitions (PRSUs and DESTs)

A complex set definition requires both PRSUs and DESTs. A complex set definition consists of a set of PRSUs with a specific quality on a DEST. Currently, only PRSU fields have qualities based on DESTs. The DEST may be a specific DEST, a named set of specific elements, DESTs with specific qualities, or all DESTS. Following is an example of a set definition for a set of PRSUs with an applied status to MS 0.

```
'prsus WHERE status = a ON ms 0
```

Note: When used in syntax in this document, the term "set definition" refers to both simple and complex set definitions.

PRSM field conditions

PRSM field conditions are used in simple and complex set definitions to describe PRSU or DEST qualities. There are many PRSU and DEST fields available in PRSM that can be used in PRSM conditions. PRSM field conditions are made up of a field name, conditional operator, and field value as shown in the following syntax.

```
<field name> <condition><field value>
```

Where the condition can be any of the following:

= equal

<> not equal

> greater than

< less than

>= greater than or equal to

<= less than or equal to

Refer to chapter 8, "Database fields" for names, values, and descriptions of PRSU and DEST fields.

Special characters

The special characters =, <>, >, <, >=, <=, &, |, and * require the use of one single quote character (') preceding the special character. (Note that on most computer keyboards, a single quote character is produced by the apostrophe key.) Furthermore

- Each special character except the wildcard character (*) must be preceded by a space character.
- The required single quote character (') must be placed at either the beginning of some word preceding the special character or immediately prior to the special character itself.
- Regardless of how many special characters are used in a command line, one properly placed single quote (') character is sufficient for the entire line.
- If a command is continued onto the next line, it is necessary to use a plus (+) at the end of that line to be able to continue.
- If a continuation line includes a special character, the line requires its own single quote (').
- The maximum length of any PRSM command is 255 characters, including spaces.

Note: Successive single quote mark characters alternately enable and disable recognition of special characters. Any special character following a second single quote mark character (in a given command) will not be interpreted as special character.

The following two examples show syntax that requires the use of a single quote mark character (').

```
prsus 'where category = gen
```

```
prsus where category ' = gen class = c
```

The following example shows syntax for a command that requires the use of a single quote mark character (') and whose length exceeds one line.

```
SELECT prsuid status category active apptime appdate destid +
'from prsuset prsus where status = a category = act +
'active = y on cm
```

The following example shows the syntax for the wildcard character (*).

```
dests 'where destid = dtc * 0
```

PRSM set syntax

The following sections describe the syntax for deriving PRSU sets and DEST sets.

PRSU set syntax

A PRSU set is a group of one or more PRSUs. PRSU sets may be operated on by PRSM commands. You may define PRSU sets using the PRSMSET command, make use of predefined set definitions, or specify the set definition as a parameter in a PRSM command.

Following are examples of PRSU set syntax.

<prsuid>

where

<prsuid>

is the name of a PRSU

<prsu_set_name>

where

prsu_set_name

is the name of a predefined or user defined set definition

ALLPRSUS

where

ALLPRSUS

is a keyword to specify all PRSUs

PRSUS WHERE <condition>

where

field_condition

is <prsu_field<condition> <field_value>

where

<condition>

is =, <>, <, >, >=, or <=

or

<field_condition> <field_condition>

<prsu_set_def> ON <prsu_set_def>

where

prsu_set_def

is a predefined or user defined PRSU set definition

dest_set_def

is a predefined or user defined DEST set definition

<prsu_set_def> & <prsu_set_def>

where

prsu_set_def

is a predefined or user defined PRSU set definition

<prsu_set_def> | <prsu_set_def>

where

prsu_set_def

is a predefined or user defined PRSU set definition

DEST set syntax

A DEST set is a group of one or more destinations. DEST sets may be operated on by PRSM commands. You may define DEST sets using the PRSMSET command, make use of predefined set definitions, or specify the set definition as a parameter in a PRSM command.

Following are examples of DEST set syntax.

<destid>

where

destid

is the name of a DEST in one of the following forms

<destname [device [unit]]>

<dest_set_name>

<dest_set_name>

where

dest_set_name

is the name of a predefined or user defined set definition

ALLDESTS

where

ALLDESTS

is a keyword to specify all DESTs

DESTS WHERE <field_condition>

where

field_condition

is <dest_field> <condition> <field_value>

where

<condition> is =, <>, <, >, >=, or <=

or

<field_condition> <field_condition>

<dest_set_def> WITH <prsu_set_def>

where

dest_set_def

is a predefined or user defined DEST set definition

prsu_set_def

is a predefined or user defined PRSU set definition

<dest_set_def> & <dest_set_def>

where

dest_set_def

is a predefined or user defined DEST set definition

<dest_set_def> | <dest_set_def>

where

dest_set_def

is a predefined or user defined DEST set definition

An asterisk (*) may be used as a wild card in DESTs.

Example

```
dests 'where destid = dtc * 0
```

Logs

PRSM activities are reported in the following types of logs:

- PRSM logs
- DCH logs
- PM logs

PRSM logs

Following are the groups and names of PRSM logs. Refer to chapter 10, "Logs" in this document and the *North American DMS-100 Log Report Reference Manual* for additional information on these logs.

- Trouble logs (PRSM logs numbered 300 to 399)
 - PRSM300—General trouble
 - PRSM301—PRSM file failure
 - PRSM302—File management failure
 - PRSM303—PRSM PRSU discrepancy
 - PRSM360—File audit failure
 - PRSM380—Automated process failure
 - PRSM381—Switch sanity failure
- Service summary logs (PRSM logs numbered 400 to 499)
 - PRSM400—PRSM action summary
 - PRSM401—Assignment summary
 - PRSM470—Status audit summary
- Information logs
 - PRSM600—General information
 - PRSM603—PRSMOPTS option change
 - PRSM680—Automated process information
 - PRSM681—AUTOPROC CI command
 - PRSM682—AUTOOPTS option change
 - PRSM683—AUTOPRSU option change

DCH logs

The DMS switch generates a DCH500 log when a PAL fault is detected.

Following is an example of a DCH500 log.

```
DCH500 JUL29 18:01:348000INFODCH StateChange DCH 2 on LTC 8:
ISTb from InSv Patch Load Flt ISG 0
```

A user can identify the cause of a fault by entering the QUERYPM FLT command at the MAP screen. Following is an example of a MAP display that indicates the reason a fault has occurred.

```
Site Flr RPosBay_idShfDescriptionSlotEqPec
HOST01A03LTEI 0351LTC : 00801BX02
Loadnames:DCHINV - EDH81AZ1 , DCH - EDH81AZ1 ; INTL INDEX : 4
DCH is ISTb
The following inservice trouble conditions exist:
    Patch Load Flt
```

PM logs

Following are PM logs that reflect PRSM activities. Refer to the *North American DMS-100 Log Report Reference Manual* for additional information on these logs.

PM128

The switch generates a PM128 log when the switch detects a PAL fault. Following is an example of the MAP display.

```
>querypm flt
```

```
Node is ISTb
    One or Both units inservice trouble
Unit 0
The following inservice troubles exist:
    Patching After Loading Fault. Run PRSM; ISTBAUDIT
Unit1
    no fault exists
```

Following is an example of a PM128 log.

```
>open pm
```

```
Done.
```

```
* PM128 FEB04 16:55:09 1900 TBL ISTB LTC 0
Node : ISTb (Unit ISTb) From InSv
Unit0 Inact: ISTb (PAL Fault) From Insv
Unit1 Act: InSv
```

Refer to the chapter 3, "Advanced manual functions" in this document for more information on the ISTBAUDIT command.

PM181

The switch generates a PM181 log when broadcast patching is performed. The log indicates whether broadcast patching was successful, and which units passed or failed. Following is an example of the log.

```
PM181 APR18 13:55:18 0900 INFO LTC 0 UNIT 0 : Inact
Node: ISTb, Unit0 Inact: ManB, Unit1 Act: InSv
Unit0 MTCARB is functional, Unit1: MTCARB is disabled
Patches were successfully applied to the following XPM:
LTC 0 0, LTC 0 1, LTC 1 1, LTC 2 0, LTC 2 1, LTC 3 1
Patches were not applied to the following XPM:
LTC 3 0: No reply from PM
```

Tables

The following types of tables contain information on PRSM:

- PRSM tables
- related tables

Refer to chapter 5, "Tables" in this document for more information on these tables.

PRSM tables

The following tables assist the user in maintaining the PRSUs in a DMS office:

- Table PRSMOPTS identifies general user options in PRSM, such as the method for applying PRSUs.
- Table PRSMALRM identifies alarm conditions audited by the PRSM Status Audit.
- Table AUTOOPTS identifies general options for the automated processes in PRSM.
- Table AUTOMON identifies which logs are monitored before and after the PRSM automated processes Autoapply and Autoinstall.
- Table AUTOPRSU identifies scheduling options for the automated processes in PRSM.

Note: Tables PRSUDATA and DESTDATA provide information on PRSUs and DESTS. The tables are read-only tables that cannot be changed directly by operating company personnel. Nortel Networks support and design personnel use these tables for debugging and problem-solving.

Related tables

Table PADNDEV lists the devices that PRSM will search for PRSU files. The table tells downloading tools where to download files.

User controlled options

An option is available to allow a user to customize the outputs of PRSM. This is called a user controlled option. This user controlled option controls PRSM responses.

PRSM responses

PRSM output varies based on different settings of the PRSM response level. The following are the different levels of PRSM responses:

- QUIET - responds with either OK or with an error message if a failure has occurred.
- BRIEF - displays a short summary of command execution.
- VERBOSE - displays detailed information about command execution.

The response levels are cumulative. Each response setting outputs all of the information output in the previous setting, plus some additional information. For example: VERBOSE outputs all of the information output in a response level of BRIEF, plus some additional information.

These response levels apply to most PRSM commands, such as REMOVE, APPLY, and VALIDATE. These response levels do not apply to the REPORT or SELECT command.

Refer to chapter 6, "Command output" in this document for illustrations of the PRSM response levels.

2 Basic manual functions

This chapter describes the basic manual functions of post-release software manager (PRSM) and the following commands:

- VALIDATE
- APPLY
- REMOVE
- DISADMIN
- REPORT
- HELP

Validating, applying, and removing patches

The following section describes how to manage post-release software updates (PRSU) with the VALIDATE, APPLY, and REMOVE commands.

VALIDATE command

Use the VALIDATE command to validate a PRSU in a DEST. Validation determines whether a PRSU should be applied to a DEST within a DMS office. The PRSU will not necessarily apply if the VALIDATE command is successful.

The VALIDATE command can validate a PRSU against all applicable nodes in the office. Each node that requires a PRSU to be applied will have a PRSU status of validated for application (VA).

For a computing module (CM) PRSU, the VALIDATE command confirms certain prerequisites are met. However, even if all prerequisites are met and the VALIDATE passes, the PRSU will not necessarily apply. An unsuccessful apply can occur as a result of a busy loader, or if loader checks fail on some part of the apply.

The VALIDATE command changes the status of the PRSU, based on the results of the command. Refer to Figure 7-1, "Transition between statuses" on page 7-2 in the chapter "Statuses" in this document.

The VALIDATE command locates the PRSU file by searching the user's directory. If the file is not found, PRSM uses the location of the file as stored previously in the PRSM database.

Once the PRSU file is located, the patch file is opened to determine if the PRSU is applicable to any DEST in the office. If the file is available and the PRSU is applicable to at least one DEST, some PRSU information is stored in the PRSM database. In addition, the status of the PRSU on the DEST is changed to VA.

If the PRSU file cannot be located, but it is known to the PRSM database, the validation is performed using the existing administrative data stored in the PRSM database.

A PRSU can be validated more than once. Each time a PRSU is validated, information from the PRSU updates the information in the database with the new information from the file unless the no database (NODB) option is used. The NODB option specifies that the administrative information that has just been validated will not be saved in the PRSM database.

Syntax

Following is the syntax for the VALIDATE command.

```
>VALIDATE <prsu_set_def> [IN <dest_set_def>] [NODB]
```

The following table lists the parameters and variables for the VALIDATE command.

Table 2-1 Parameters and variables for VALIDATE command (Sheet 1 of 2)

Parameter or Variable	Values	Description
<prsu_set_def>		One or more PRSUs
	<prsu_id>	Specific PRSU
	<prsu_set_name>	Specific PRSU set, defined by PRSM or the user
	ALLPRSUS	All PRSUs
	PRSUS WHERE <field_condition>	All PRSUs with a specified condition in a database field
	<prsu_set_def> ON <dest_set_def>	A set of PRSUs with specific conditions on a set of DESTs with specific conditions

Table 2-1 Parameters and variables for VALIDATE command (Sheet 2 of 2)

Parameter or Variable	Values	Description
	<prsu_set_def> & <prsu;_set_def>	The common elements between two PRSU sets
	<prsu_set_def> <prsu_set_def>	All the elements of two PRSU sets
<dest_set_def>		One or more DESTs
	<device device_no> [<unit unit_no>]	Specific DEST
	<dest_set_name>	Specific DEST set, defined by PRSM or the user
	ALLDESTS	All DESTs
	DESTS WHERE <field_condition>	All DESTs with a specified condition in a database field
	<dest_set_def> WITH <prsu_set_def>	A set of DESTs, with a specific condition, that has a set of PRSUs with a specific condition
	<dest_set_def> & <dest_set_def>	The common elements between two DEST sets
	<dest_set_def> <dest_set_def>	All the elements of two DEST sets
NODB		No updating of the PRSM database

Examples

The following table lists possible uses of the VALIDATE command, with an example for each use.

Table 2-2 Uses of VALIDATE command (Sheet 1 of 2)

Use	Examples
Validate a specific PRSU in all available DESTs.	>VALIDATE sam00ib3
Validate a specific PRSU in all XPMs.	>VALIDATE xwt07x07 IN xpm
Validate a specific PRSU in a specific type of XPM.	>VALIDATE xwt07x07 IN ltc

2-4 Basic manual functions

Table 2-2 Uses of VALIDATE command (Sheet 2 of 2)

Use	Examples
Validate a specific PRSU in a specific XPM.	>VALIDATE xwt07x07 IN dtc 0
Validate a specific PRSU in a specific XPM unit.	>VALIDATE xwt07x07 IN rcc 0 0
Validate a specific PRSU in all ISNs	>VALIDATE sam00ib5 IN isn
Validate a specific PRSU in a specific type of ISN.	>VALIDATE sam00ib5 IN liu7
Validate a specific PRSU in a specific ISN.	>VALIDATE sam00ib5 IN ms 0
Validate a specific PRSU in a specific ISN unit.	>VALIDATE sam00ib5 IN lim 0 0
Validate a specific PRSU in a specific SPM CEM.	>VALIDATE scb01s01 IN spm 0 cem 0
Validate a specific PRSU in a predefined DEST set.	>VALIDATE sam00ib3 IN samdestset
Validate a specific PRSU in specific multiple DESTs.	>VALIDATE 'sam00ib5 IN liu7 100 liu7 101 liu7 109
Validate a specific PRSU in DESTs with a specific field condition.	>VALIDATE sam00ib5 IN DESTS WHERE + >'loadname = msc05ao
Validate specific PRSUs in a specific DEST.	>VALIDATE 'xwt05x07 1 xwt07x07 IN dtc 0
Validate a predefined PRSU set.	>VALIDATE nvstat
Validate a predefined PRSU set without updating the PRSM database.	>VALIDATE nvstat NODB
Validate all PRSUs with a specific field condition.	>VALIDATE 'PRSUS WHERE status = nv
Validate all PRSUs in a specific DEST with a specific status in a specific DEST.	>VALIDATE 'PRSUS WHERE status = a ON ms 0 + >IN ms 1

APPLY command

Use the APPLY command to apply PRSUs and change the software based on the information in the PRSU.

The APPLY command can apply a PRSU, a PRSU set, or a PRSU set definition in a specific DEST, all applicable nodes, a DEST set, or a DEST set definition. To successfully apply a PRSU, a PRSU file must be found on one of the DMS devices.

Note: A \$DF PRSU cannot be used with the APPLY command.

When searching for the PRSU file to apply, the APPLY command locates the PRSU file by first searching the user's directory. If the file is not found, PRSM uses the location of the file as stored previously in the PRSM database.

The PRSM database is updated with information found in the PRSU file when the APPLY command was executed. A VALIDATE is performed automatically prior to an APPLY on the specified DESTs. Subsequently, the status of the PRSU is updated to VA in the DESTs where the PRSU was successfully validated. Once the PRSU is applied, the status is updated to A.

The APPLY command changes the status of the PRSU, based on the results of the command. Refer to Figure 7-1, "Transition between statuses".

When applying a PRSU to multiple DESTs, PRSM determines the application method. For example, in the case of XMS-based peripheral modules (XPM), PRSM chooses broadcast patching whenever possible. In the case of Spectrum peripheral modules (SPM) or Integrated Services Nodes (ISN), the application is performed serially. The user can override the PRSM application method for XPMs by setting the field DESTMETH in table PRSMOPTS to NO_BROADCAST.

Syntax

Following is the syntax for the APPLY command.

```
>APPLY <prsu_set_def> [IN <dest_set_def>]
```

The following table lists the parameters and variables for the APPLY command.

Table 2-3 Parameters and variables for APPLY command (Sheet 1 of 2)

Parameter or Variable	Values	Description
<prsu_set_def>		One or more PRSUs
	<prsu_id>	Specific PRSU

Table 2-3 Parameters and variables for APPLY command (Sheet 2 of 2)

Parameter or Variable	Values	Description
	<prsu_set_name>	Specific PRSU set, defined by PRSM or the user
	ALLPRSUS	All PRSUs
	PRSUS WHERE <field_condition>	All PRSUs with a specified condition in a database field
	<prsu_set_def> ON <dest_set_def>	A set of PRSUs with specific conditions on a set of DESTs with specific conditions
	<prsu_set_def> & <prsu_set_def>	The common elements between two PRSU sets
	<prsu_set_def> <prsu_set_def>	All the elements of two PRSU sets
<dest_set_def>		One or more DESTs
	<device device_no> [<unit unit_no>]	Specific DEST
	<dest_set_name>	Specific DEST set, defined by PRSM or the user
	ALLDESTS	All DESTs
	DESTS WHERE <field_condition>	All DESTs with a specified condition in a database field
	<dest_set_def> WITH <prsu_set_def>	A set of DESTs, with a specific condition, that has a set of PRSUs with a specific condition
	<dest_set_def> & <dest_set_def>	The common elements between two DEST sets
	<dest_set_def> <dest_set_def>	All the elements of two DEST sets

Examples

The following table lists possible uses of the APPLY command, with an example for each use.

Table 2-4 Uses of APPLY command (Sheet 1 of 2)

Use	Examples
Apply a specific PRSU to all available DESTs.	>APPLY sam00ib3
Apply a specific PRSU to all XPMs.	>APPLY xwt07x07 IN xpm
Apply a specific PRSU to a specific type of XPM.	>APPLY xwt07x07 IN ltc
Apply a specific PRSU to a specific XPM.	>APPLY xwt07x07 IN dtc 0
Apply a specific PRSU to a specific XPM unit.	>APPLY xwt07x07 IN rcc 0 0
Apply a specific PRSU to all ISNs.	>APPLY sam00ib5 IN isn
Apply a specific PRSU to a specific type of ISN.	>APPLY sam00ib5 IN liu7
Apply a specific PRSU to a specific ISN.	>APPLY sam00ib5 IN ms 0
Apply a specific PRSU to a specific ISN unit.	>APPLY sam00ib5 IN lim 0 0
Apply a specific PRSU to a specific SPM CEM.	>APPLY scb01s01 IN spm 0 cem 0
Apply a specific PRSU to a predefined DEST set.	>APPLY sam00ib3 IN samdestset
Apply a specific PRSU to specific multiple DESTs.	>APPLY 'sam00ib5 IN liu7 100 liu7 101 liu7 109
Apply a specific PRSU to DESTs with a specific field condition.	>APPLY sam00ib5 IN DESTS WHERE + >'loadname = msc05ao
Apply specific PRSUs to a specific DEST.	>APPLY 'xwt05x07 1 xwt07x07 IN dtc 0
Apply a predefined PRSU set.	>APPLY nvstat

Table 2-4 Uses of APPLY command (Sheet 2 of 2)

Use	Examples
Apply all PRSUs with a specific field condition.	<code>>APPLY 'PRSUS WHERE status = nv</code>
Apply all PRSUs to a specific DEST with a specific status in a specific DEST.	<code>>APPLY 'PRSUS WHERE status = a ON ms 0 + >IN ms 1</code>

REMOVE command

Use the REMOVE command to remove a PRSU, or a set of PRSUs from the memory of a DEST or a set of DESTs.

A PRSU file does not have to be available to remove a PRSU from a SOS-based product, such as an ISN or the CM. However, the CM must process the PRSU file at least once before it can remove the PRSU. This processing also applies to the pre-patched ISN loads and ISN patches applied by a previous CM load.

A PRSU file must be available to remove a PRSU from an SPM or XPM product. Upon removal of a PRSU from any DEST, the AOR field associated with the PRSU in the DEST is set from Y to N. The AOR field is not used for SOS-based targets. Refer to the chapter "Statuses" in this document for more information on the AOR field.

Note: A special file audit checks for the presence of these PRSUs.

The REMOVE command changes the status of the PRSU, based on the results of the command. Refer to the Figure 7-1, "Transition between statuses" on page 7-2 in the chapter "Statuses" in this document.

Following is the syntax for the REMOVE command.

`>REMOVE <prsu_set_def> [IN <dest_set_def>]`

The following table lists the parameters and variables for the REMOVE command.

Table 2-5 Parameters and variables for REMOVE command (Sheet 1 of 2)

Parameter or Variable	Values	Description
<code><prsu_set_def></code>		One or more PRSUs
	<code><prsu_id></code>	Specific PRSU

Table 2-5 Parameters and variables for REMOVE command (Sheet 2 of 2)

Parameter or Variable	Values	Description
	<prsu_set_name>	Specific PRSU set, defined by PRSM or the user
	ALLPRSUS	All PRSUs
	PRSUS WHERE <field_condition>	All PRSUs with a specified condition in a database field
	<prsu_set_def> ON <dest_set_def>	A set of PRSUs with specific conditions on a set of DESTs with specific conditions
	<prsu_set_def> <prsu_set_def>	All the elements of two PRSU sets
	<prsu_set_def> & <prsu_set_def>	The common elements between two PRSU sets
<dest_set_def>		One or more DESTs
	<device device_no> [<unit unit_no>]	Specific DEST
	<dest_set_name>	Specific DEST set, defined by PRSM or the user
	ALLDESTS	All DESTs
	DESTS WHERE <field_condition>	All DESTs with a specified condition in a database field
	<dest_set_def> WITH <prsu_set_def>	A set of DESTs, with a specific condition, that has a set of PRSUs with a specific condition
	<dest_set_def> <dest_set_def>	All the elements of two DEST sets
	<dest_set_def> & <dest_set_def>	The common elements between two DEST sets

Examples

The following table lists possible uses of the REMOVE command, with an example for each use.

Table 2-6 Uses of REMOVE command (Sheet 1 of 2)

Use	Examples
Remove a specific PRSU from all available DESTs.	>REMOVE sam00ib3
Remove a specific PRSU from all XPMs.	>REMOVE xwt07x07 IN xpm
Remove a specific PRSU from a specific type of XPM.	>REMOVE xwt07x07 IN ltc
Remove a specific PRSU from a specific XPM.	>REMOVE xwt07x07 IN dtc 0
Remove a specific PRSU from a specific XPM unit.	>REMOVE xwt07x07 IN rcc 0 0
Remove a specific PRSU from all ISNs	>REMOVE sam00ib5 IN isn
Remove a specific PRSU from a specific type of ISN.	>REMOVE sam00ib5 IN liu7
Remove a specific PRSU from a specific ISN.	>REMOVE sam00ib5 IN ms 0
Remove a specific PRSU from a specific SPM CEM.	>REMOVE scb01s01 IN spm 0 cem 0
Remove a specific PRSU from a specific ISN unit.	>REMOVE sam00ib5 IN lim 0 0
Remove a specific PRSU from a predefined DEST set.	>REMOVE sam00ib3 IN samdestset
Remove a specific PRSU from specific multiple DESTs.	>REMOVE 'sam00ib5 IN liu7 100 liu7 101 liu7 109
Remove a specific PRSU from DESTs with a specific field condition.	>REMOVE 'sam00ib5 IN DESTS WHERE + >'loadname = msc05ao
Remove specific PRSUs from a specific DEST.	>REMOVE 'xwt05x07 1 xwt07x07 IN dtc 0
Remove a predefined PRSU set.	>REMOVE obs_not_rem

Table 2-6 Uses of REMOVE command (Sheet 2 of 2)

Use	Examples
Remove all PRSUs with a specific field condition.	>REMOVE 'PRSUS WHERE status = a
Remove all PRSUs from a specific DEST with a specific status in a specific DEST.	>REMOVE 'PRSUS WHERE status = a ON ms 0 + >IN ms 1
Remove a specific PRSU and all PRSUs that depend on it	>REMOVE 'PRSUS WHERE dependson = plr01bb1

When you perform the REMOVE command, and the response level is set to VERBOSE, PRSM displays a list of PRSUs in removal order. PRSM also reminds you to verify that the PRSUs in the set can be removed.

Displaying information

The following section describes how to display information using the DISADMIN, REPORT, and HELP commands.

DISADMIN command

Use the DISADMIN command to display administrative information on a PRSU, a set of PRSUs, or a PRSU set definition.

The DISADMIN command locates the PRSU file by first searching the user's directory. DISADMIN displays each copy of the PRSU file that it finds, even equal PRSU files. If the PRSU file is not found in the search, PRSM displays the administrative information as stored previously in the PRSM database.

Syntax

Following is the syntax for the DISADMIN command.

```
>DISADMIN <prsu_set_def>
```

The following table lists the parameters and variables for the DISADMIN command.

Table 2-7 Parameters and variables for DISADMIN command (Sheet 1 of 2)

Parameter or Variable	Values	Description
<prsu_set_def>		One or more PRSUs
	<prsu_id>	Specific PRSU

Table 2-7 Parameters and variables for DISADMIN command (Sheet 2 of 2)

Parameter or Variable	Values	Description
	<prsu_set_name>	Specific PRSU set, defined by PRSM or the user
	ALLPRSUS	All PRSUs
	PRSUS WHERE <field_condition>	All PRSUs with a specified condition in a database field
	<prsu_set_def> ON <dest_set_def>	A set of PRSUs with specific conditions on a set of DESTs with specific conditions
	<prsu_set_def> & <prsu_set_def>	The common elements between two PRSU sets
	<prsu_set_def> <prsu_set_def>	All the elements of two PRSU sets

Examples

The following table lists possible uses of the DISADMIN command, with an example for each use.

Table 2-8 Uses of DISADMIN command

Use	Examples
Display information on a specific PRSU.	>DISADMIN sam00ib3
Display information on a specific PRSU set.	>DISADMIN sampruset
Display information on two PRSUs.	>DISADMIN 'sam00ib5 sam01ib5
Display information on all PRSUs with a specific field condition.	>DISADMIN 'PRSUS WHERE status = va

REPORT command

Use the REPORT command to extract specific information from a limited number of pre-defined fields. The REPORT command generates the following types of reports:

- Information on specific PRSUs
- Information on PRSUs applied to a specific DEST

- PRSUs that use a specific module
- PRSUs applied on or after a specific date
- PRSUs that depend on another PRSU

The REPORT command is a shortcut to the SELECT command. The REPORT command is not as broad as the SELECT command, but it displays specific information using a simple command. Use the SELECT command to display customized or additional information.

Syntax

Following is the syntax for the REPORT command.

```
>REPORT <REPORT_NAME> <report_value>
```

The following table lists the parameters and variables for the REPORT command.

Table 2-9 Parameters and variables for REPORT command

Parameter	Name	Value	Description
<REPORT_NAME>			The type of the report
	PRSU	<prsuid>	Report on PRSU or PRSUs
	DEST	<destid>	Report on DEST or DESTs
	DATE	<date>	Report on PRSUs applied on or after a specific date, in the format YYMMDD or YYYYMMDD (six character format interprets century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20)
	MODULE	<module>	Report on all PRSUs with a specific module
	DEPEND	<prsuid>	Report on all PRSUs that depend on a specific PRSU

Examples

The following table lists possible uses of the REPORT command, with an example for each use.

Table 2-10 Uses of REPORT command

Use	Examples
Display information on a specific PRSU.	>REPORT PRSU sam00ib3
Display information on a specific DEST.	>REPORT DEST ms 0
Display PRSUs applied on or after a specific date.	>REPORT DATE 961219
Display information on a specific module.	>REPORT MODULE sysdefs
Display all PRSUs that depend on a specific PRSU	>REPORT DEPEND xyz15cb1

Following is an example of the output for a REPORT PRSU command.

>REPORT PRSU GJG03BB9

```

PRSUID      MODULE      EDIT CAT A ST STATDATE STATT FR DESTID
-----
GJG03BB9    GJGMOD1    AA01 GEN N  A 19970725 14:35 NA CM
              PRAUTUTL  AE01
    
```

Note: In the example of the output to the REPORT PRSU command, the FR column represents free memory. Refer to a discussion of the "FREEMEM command" on page 3-20. Also, see the description of the PRSU field "freemem" in Table 8-1, "PRSU fields"

Following is an example of the output for a REPORT DEST command.

>REPORT DEST ltc 0

```

PRSUID      STATDATE  STATT  CAT ACT ST LOADNAME H DESTID
-----
XPM00x81    19970725  15:17  SRC N  A  ELI81AX  N LTC 0 0
XPM01x81    19970725  15:17  SRC N  A  ELI81AX  N LTC 0 1
XPM02x81    19970725  15:17  SRC N  A  ELI81AX  N LTC 0 0
XPM03x81    19970725  15:17  SRC N  A  ELI81AX  N LTC 0 1
    
```

Following are examples of the output for a REPORT DATE command.

>REPORT DATE 970723

PRSUID	MODULE	EDIT	STATDAT	STATT	CAT	ACT	ST	DESTID
XCU06X07			19970723	16:07	SRC		A	LTC 0 0
XCU06X07			19970723	17:03	SRC		A	LTC 0 1
XCU06X07			19970723	18:21	SRC		V	ALTC 1 0
XCU06X07			19970723	18:21	SRC		V	ALTC 1 1
XCU06X07			19970723	14:34	SRC		A	PPXL
ELI07BI1_961119								
HUN25IB9	HUNMOD1	AA01	19970724	05:37	GEN	N	A	NIU 0 0
	HUNMOD2	AA01						
	HUNMOD3	AA01						
	PATCHTST	AE07						

>REPORT DATE 19970723

PRSUID	MODULE	EDIT	STATDAT	STATT	CAT	ACT	ST	DESTID
XCU06X07			19970723	16:07	SRC		A	LTC 0 0
XCU06X07			19970723	17:03	SRC		A	LTC 0 1
XCU06X07			19970723	18:21	SRC		V	ALTC 1 0
XCU06X07			19970723	18:21	SRC		V	ALTC 1 1
XCU06X07			19970723	14:34	SRC		A	PPXL
ELI07BI1_961119								
HUN25IB9	HUNMOD1	AA01	19970724	05:37	GEN		N	ANIU 0 0
	HUNMOD2	AA01						
	HUNMOD3	AA01						
	PATCHTST	AE07						

Following is an example of the output for a REPORT MODULE command.

>REPORT MODULE PTCHTSTA

PRSUID	MODULE	EDIT	STATDAT	STATT	CAT	ACT	ST	DESTID
BOB49C89			19970725	14:32	GEN	N	A	CM
	BOBMOD1	AA01						
	BOBMOD2	AA01						
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	MS 0
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	MS 1
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	NIU 0 0
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	NIU 0 1
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	LIU7 0
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	LIU7 1
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	LIU7 2
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	A	FRUI 0
BOB49C89	PTCHTSTA	AG01	19970725	15:10	GEN	N	R	XLIU 0

Following is an example of the output for a REPORT DEPEND command.

>REPORT DEPEND MAP08CT9

PRSUID	CAT	AUTOA	SPEC	ONHOLD
MAP08CT9	GEN	Y	N	N
GIF09CT9	GEN	Y	N	N
MAP09CT9	GEN	Y	N	N
RUS76CT9	GEN	Y	N	N

HELP command

Use the HELP command to access the PRSM help system. The PRSM help system is a series of text panels grouped into levels. Each level consists of help text for the current level and a list of related subtopics.

ATTENTION

Do not type QUIT to quit the PRSM help system. The QUIT command exits the user from the PRSM session. It is not necessary to quit the PRSM help system.

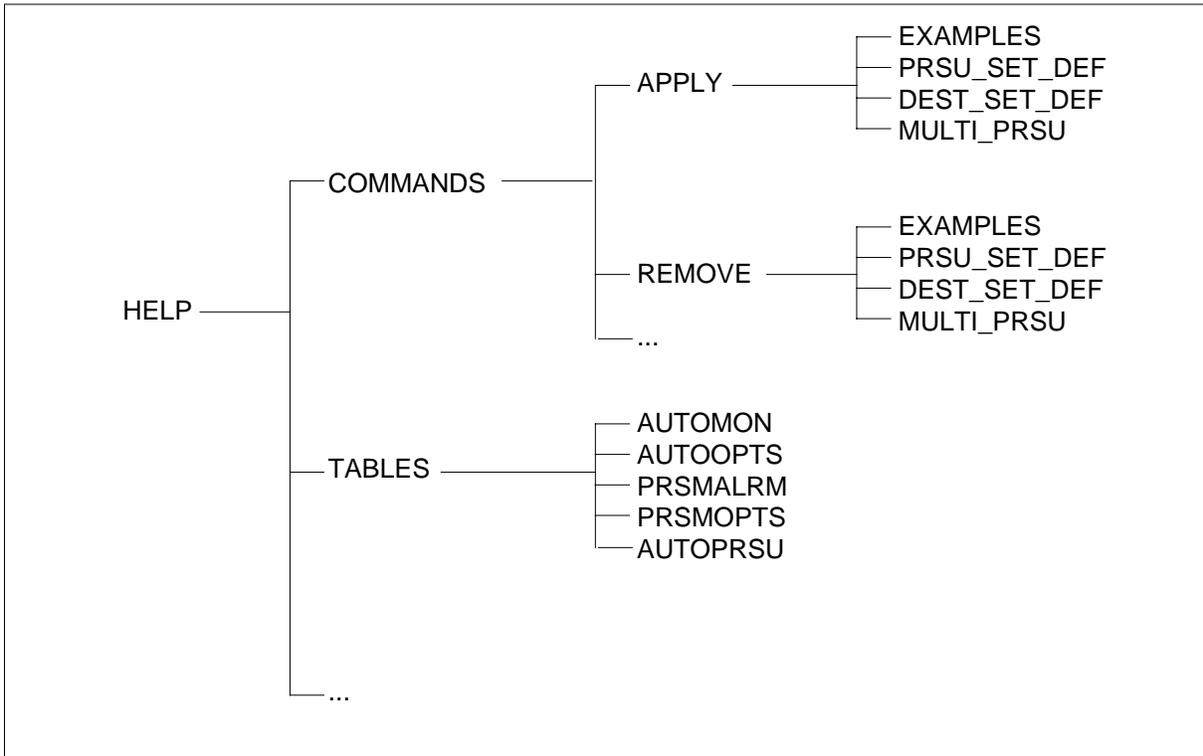
Within PRSM, type HELP to enter the top level of the PRSM help system.

ATTENTION

The subtopic WHATSNEW describes new or changed PRSM functionality for current and recent releases.

The following figure illustrates the structure of levels for the HELP subtopics.

Figure 2-1 Help panel structure for COMMANDS and TABLES



Use the **HELP** command with parameters and variables to move within the PRSM help system.

- To enter a subtopic, type **HELP** and the name of the subtopic.
- To enter the previous level, type **HELP PREVIOUS**.
- To display the name of the current level, type **HELP CURRENT**.
- To return to the top level of the help system, type **HELP** with no parameter.
- To obtain information on the PRSM help system, type **HELP HELP**.

Subtopics may be combined in a single **HELP** command to access a specific subtopic. To access subtopic **AUTOMON** from the top level, for example, type **HELP TABLES AUTOMON**.

Syntax

Following is the syntax for the **HELP** command.

```
>HELP
```

```
>HELP <action>
```

The following table lists the parameters and variables for the HELP command.

Table 2-11 Parameters and variables for HELP command

Parameter or Variable	Values	Description
<subtopic>	<subtopic_name>	The name of the subtopic
<action>	PREVIOUS	Move to the previous level.
	CURRENT	Display the current level.
	HELP	Obtain information on the help system.

Examples

The following table lists possible uses of the HELP command, with an example for each use.

Table 2-12 Uses of HELP command

Use	Examples
Enter the top level of the PRSM help system.	>HELP
Enter a subtopic.	>HELP COMMANDS
Enter a specific level within a subtopic.	>HELP COMMANDS APPLY
Enter the previous level.	>HELP PREVIOUS
Get information on the PRSM help system.	>HELP HELP

3 Advanced manual functions

This chapter describes the advanced manual functions of PRSM for the following commands:

- SETDEF
- SELDEF
- PRSMSET
- SELECT
- ASSIGN
- DBAUDIT
- LFAUDIT
- AUTOPROC
- FILEAUDIT
- FREEMEM
- ISTBAUDIT
- PRSM ABORT
- PRSM RECONNECT

Using set and select definitions

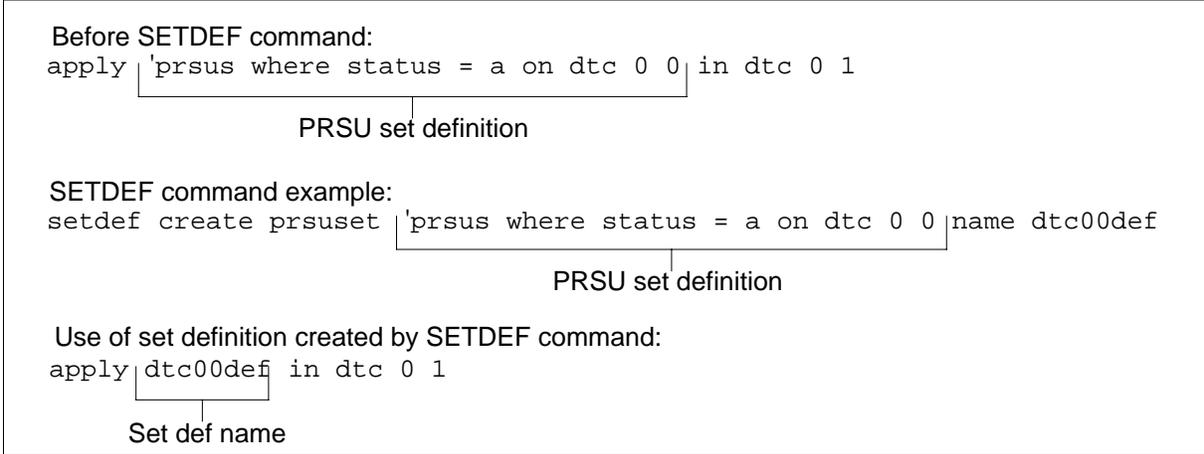
This section describes how to create and use set definitions and select definitions, using the SETDEF, SELDEF, and PRSMSET commands.

SETDEF command

Throughout the PRSM commands, the command syntax refers to set definitions. Set definitions describe sets of PRSU and DEST elements to be used in PRSM commands, such as SELECT, APPLY, and REMOVE. However, set definitions can become too long and tedious to type. The SETDEF command enables the user to store and abbreviate long set definitions.

The following figure illustrates the practical reasons for using the SETDEF command.

Figure 3-1 Example of SETDEF command



A SETDEF name may be defined by the user or PRSM. PRSM generates a maximum of 20 PRSM-named set definitions per user. PRSM deletes all PRSM-named set definitions when the user quits the PRSM session. All user-named set definitions are permanent.

The set definition remains constant, even if the contents of the set change. The following figure illustrates the use of a set definition when the contents of the set have changed.

Figure 3-2 Example of set definition with contents that change

dtc 0 0		
PRSUID	Status	<u>Now</u>
xam00x05	A	>setdef create prsuset 'prsus where status = a on dtc 0 0 name dtc00def >apply dtc00def in dtc 0 1 This command will apply the PRSUs xam00x05, xam01x05, xgg00x05, and xgg01x05 in dtc 0 1.
xam01x05	A	
xgg00x05	A	
xgg01x05	A	
xet00x05	VA	
xmj05x05	VA	
xgm03x05	R	
dtc 0 0		<u>At a Later Time</u>
PRSUID	Status	>apply dtc00def in dtc 0 1
xam00x05	A	This command will apply the PRSUs xam00x05, xam01x05, xgg00x05, xgg01x05, xet00x05, and xmj05x05 in dtc 0 1. Two more PRSUs are at A status. The set definition dtc00def dynamically finds the PRSUs at A status on dtc 0 0 at that instant in time.
xam01x05	A	
xgg00x05	A	
xgg01x05	A	
xet00x05	A	
xmj05x05	A	
xgm03x05	R	

To simplify the use of set definitions, PRSM provides a library of predefined set definitions. These predefined set definitions are permanent and cannot be deleted. Predefined set definitions can be used the same way that user-named set definitions and PRSM-named set definitions are used. Predefined PRSU set definitions and DEST set definitions are listed in chapter 9, "Predefined set definitions"

Syntax

Following is the syntax of the SETDEF command:

```
>SETDEF CREATE <prsm_set_def> [NAME <setdef_name>]
```

```
>SETDEF DELETE <setdef_name>
```

```
>SETDEF QUERY [<setdef_name>]
```

The following table lists the parameters and variables for the SETDEF command.

Table 3-1 Parameters and variables for SETDEF command

Parameter or Variable	Values	Description
CREATE		Create a set definition of the specified fields. The set definition may be named by the user or PRSM.
QUERY		Display the complete set definition or the names of all set definitions.
DELETE		Delete the specified set definition.
<prsm_set_def>		Specifies the PRSU set or DEST set
	PRSUSET <prsu_set_def>	A group of one or more PRSUs
	DESTSET <dest_set_def>	A group of one or more DESTs
<setdef_name>		The name of the set definition

If a name is not provided for a new set definition, SETDEF CREATE creates a name based on the following format:

setdef##

where

##

is a number from 00 to 19

Examples

Following are examples of the SETDEF command:

- Create a named set definition of PRSUs with GEN category.

```
>SETDEF CREATE PRSUSET prsus where category '= gen +
>NAME samdef
```

- Display the complete set definition of samdef.

```
>SETDEF QUERY samdef
```

- Display all the set definitions in the office.

```
>SETDEF QUERY
```

- Delete the samdef set definition.

```
>SETDEF DELETE samdef
```

SELDEF command

Use the SELDEF command to define selection criteria that can be used with the SELECT command. By using select definitions, a user can eliminate lengthy text strings and simplify PRSM reporting.

A SELDEF name may be defined by the user or PRSM. PRSM generates a maximum of 20 PRSM-named select definitions per user. PRSM deletes all PRSM-named select definitions when the user quits the session. All user-named select definitions are permanent.

To simplify the use of select definitions, PRSM provides a library of predefined select definitions. These predefined select definitions are permanent and cannot be deleted. Predefined select definitions can be used in the same ways that user-named select definitions and PRSM-named select definitions can be used. PRSM predefined select definitions are listed in chapter 9, "Predefined set definitions"

Syntax

Following is the syntax of the SELDEF command:

```
>SELDEF CREATE <field>[<field>...<field>] FROM <prsm_set_def>
[<select_option>] [NAME <seldef_name>]
```

```
>SELDEF DELETE <seldef_name>
```

```
>SELDEF QUERY [<seldef_name>]
```

The following table lists the parameters and variables for the SELDEF command.

Table 3-2 Parameters and variables for SELDEF command

Parameter or Variable	Values	Description
CREATE		Create a select definition. The select definition may be named by the user or PRSM.
QUERY		Display the complete select definition or the names of all select definitions.
DELETE		Delete the specified select definition.
<field>		Specifies a PRSM database field. Up to 15 fields may be specified.
<prsm_set_def>		Specifies the PRSU set or DEST set
	PRSUSET <prsu_set_def>	A group of one or more PRSUs
	DESTSET <dest_set_def>	A group of one or more DESTs
<select_option>	FLDWIDTH<field1><width>	Specifies the width of the field
<seldef_name>	<name>	The name of the select definition

If a name is not provided for a new select definition, SELDEF CREATE creates a name based on the following format:

seldef##

where

##

is a number from 00 to 19

Examples

Following are examples of the SELDEF command:

- Create a select definition that lists the id, category, and class of all general category PRSUs.

```
>SELDEF CREATE prsuid category class FROM prsuset prsus
+ >WHERE 'category = gen NAME samdef
```

- Display the complete select definition of samdef.

```
>SELDEF QUERY samdef
```

- Display all the select definitions in the office.

```
>SELDEF QUERY
```

- Delete the samdef select definition.

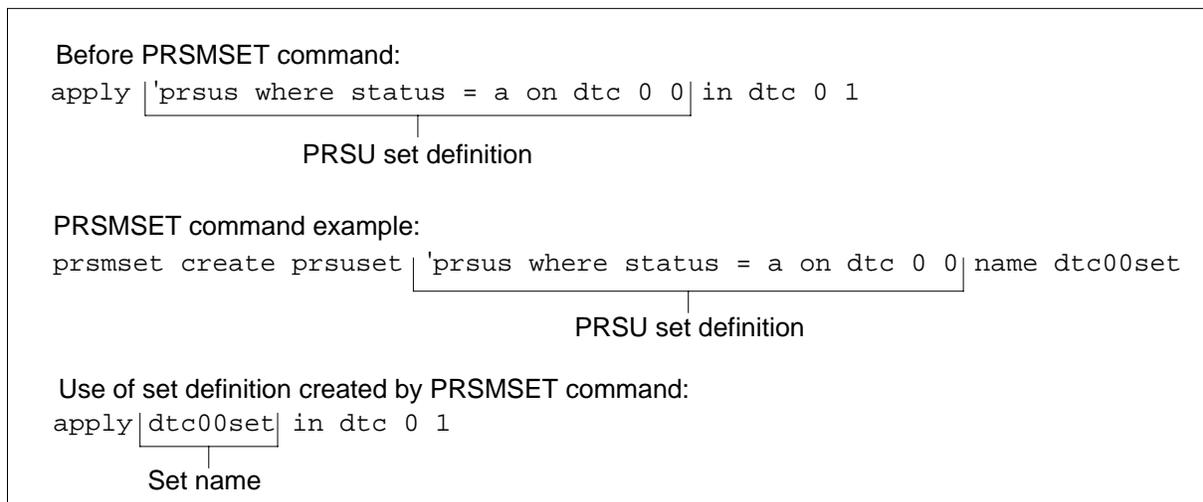
```
>SELDEF DELETE samdef
```

PRSMSET command

Use the PRSMSET command to create a temporary set of PRSU or destination elements. The PRSMSET command captures the elements at the time the set is created. All sets created by the PRSMSET command are deleted at the end of the session.

The following figure illustrates practical reason for using the PRSMSET command.

Figure 3-3 Example of PRSMSET command



A set name may be defined by the user or PRSM. PRSM generates a maximum of 20 PRSM-named set names. PRSM deletes all sets created by the PRSMSET command when the user quits the PRSM session.

The set created by the PRSMSET command reflects the contents of the set at the time of the set's creation. The following figure illustrates the use of a set when the contents of the set have changed.

Figure 3-4 Example of set with contents that change

dttc 0 0		
PRSUID	Status	<u>Now</u>
xam00x05	A	>prsmset create prsuset 'prsus where status = a on dttc 0 name dttc00set >apply dttc00set in dttc 0 1 This command applies a set containing the PRSUs xam00x05, xam01x05, xgg00x05, and xgg01x05.
xam01x05	A	
xgg00x05	A	
xgg01x05	A	
xet00x05	VA	
xmj05x05	VA	
xgm03x05	R	
dttc 0 0		<u>At a Later Time</u>
PRSUID	Status	>apply dttc00set in dttc 0 1
xam00x05	A	This command applies a set containing the PRSUs xam00x05, xam01x05, xgg00x05 and xgg01x05 even though two other PRSUs are at A status in dttc 0 0. Note: When a user quits PRSM, the sets created using PRSMSET are deleted.
xam01x05	A	
xgg00x05	A	
xgg01x05	A	
xet00x05	A	
xmj05x05	A	
xgm03x05	R	

Syntax

Following is the syntax of the PRSMSET command:

```
>PRSMSET CREATE <prsm_set_def> [NAME <set_name>]
```

The PRSMSET command silently returns the "size" of the set created. This "return code" can be preserved in a variable that can be printed, as shown in the following example.

Figure 3-5 Example of PRSMSET CREATE command

```

>prsmset create prsuset allprsus -> var1
PRSMSET00 created
>print var1
8
>prsmset query prsmset00
BAS11BL
SHR10
TL10BL
CCM10BL
CNA10BL
M88K
LEC00010
LEC0010
>

```

```
>PRSMSET ADD <prsm_set_def> NAME <set_name>
```

```
>PRSMSET REMOVE <prsm_set_def> NAME <set_name>
```

```
>PRSMSET DELETE <set_name>
```

```
>PRSMSET QUERY [<set_name>]
```

The following table lists the parameters and variables for the PRSMSET command.

Table 3-3 Parameters and variables for PRSMSET command (Sheet 1 of 2)

Parameter or Variable	Values	Description
CREATE		Create a specified PRSU or DEST set. The set may be named by the user or PRSM.
ADD		Add the specified element to the set.
REMOVE		Remove the specified element from the set.
QUERY		Display the elements in the set or the names of the sets that have been created with the PRSMSET command.
DELETE		Delete the specified set

Table 3-3 Parameters and variables for PRSMSET command (Sheet 2 of 2)

Parameter or Variable	Values	Description
<prsm_set_def>		Specifies the PRSU set or DEST set
	PRSUSET <prsu_set_def>	A group of one or more PRSUs
	DESTSET <dest_set_def>	A group of one or more DESTs
<set_name>	<name>	The name of the set

If a name is not provided for a new select definition, PRSMSET CREATE creates a name based on the following format:

prsmset##

where

##

is a number from 00 to 19

Examples

Following are examples of the PRSMSET command:

- Create a set of all MS PRSUs that are valid for application.

```
>PRSMSET CREATE prsuset prsus where status '= va on ms
+ >NAME ms_vastat
```

- Add a PRSU to the set ms_vastat.

```
>PRSMSET ADD prsuset sam01ib5 NAME ms_vastat
```

- Remove a PRSU from the set ms_vastat.

```
>PRSMSET REMOVE prsuset sam00ib5 NAME ms_vastat
```

- Display the contents of the set ms_vastat.

```
>PRSMSET QUERY ms_vastat.
```

- Display all the temporary sets in the office.

```
>PRSMSET QUERY
```

- Delete the set ms_vastat.

```
>PRSMSET DELETE ms_vastat.
```

Generating custom reports

The following section describes how to use the SELECT command to generate custom reports.

SELECT command and PRSM reports

Use the SELECT command to display detailed and customized PRSM information.

The SELECT command may be used with up to 15 database fields at a time. The SELECT command extracts information from the specified fields in the PRSM database for any of the following PRSM entities:

- PRSU
- set of PRSUs
- PRSU set definition
- DEST
- DESTs
- DEST set definition

Note: Refer to chapter 8, "Database fields" for the names of the fields in the PRSM database.

The SELECT command may also be used to gather information on an office, current PRSM sessions, and default PRSM settings.

Syntax

Following is the syntax for the SELECT command:

```
>SELECT <field1>[<field1>...<field1>] FROM <prsm_set_def>  
[select_option]
```

```
>SELECT <predef_select>
```

```
>SELECT <prsm_option> [<prsm_option>]
```

The following table lists the parameters and variables for the SELECT command.

Table 3-4 Parameters and variables for SELECT command

Parameter or Variable	Values	Description
<field>	<name>	Specifies a PRSM database field. Up to 15 fields may be specified.
<prsm_set_def>		Specifies the PRSU set or DEST set
	PRSUSET <prsu_set_def>	A group of one or more PRSUs
	DESTSET <dest_set_def>	A group of one or more DESTs
<select_option>	FLDWIDTH <field1><width>	Specifies the width of the field
<predef_select>		A pre-defined select definition, a user-defined select definition, or a PRSM-defined select definition
<prsm_option>		Identifies the PRSM option
	RESPONSE_LEVEL	The command output response level of the current PRSM sessions
	DEF_RESP_LEVEL	The default command output response level
	USER_NAME	The name of each current CI user of PRSM
	CONNECTED	Connection status of each current PRSM CI user
	PROCID	The procid of the PRSM session
	COMMAND	Commands currently executed by active PRSM sessions
	OFFICE_CLLI	The CLLI for the office, which is required in determining the password for an ACT patch
	LOG_OFFICE_ID	The office id that appears in the headers of the logs

Multiple database fields and complex set definitions can create a long SELECT command string. Following is a sample SELECT command and the command output response.

```
>SELECT prsuid module edition category class restart status
destname + >FROM prsuset allprsus on cm
```

PRSUID	MODULE	EDIT	CAT	CL	RESTART	ST	DEST
LCF24CB7	PATLCF24	AA01	GEN	C	NONE	A	CM
	PETMAP	AE01					
ACK52CB7	SWCTFAUI	AC01	GEN	C	NONE	A	CM
	SWGTLIU	AD01					
PTF40CC6	SLEAPPUI	AU01	ACT	F	NONE	A	CM

Examples

Following are examples of the SELECT command.

- Display the removed PRSUs for a specific DTC.

```
>SELECT 'prsuid status FROM PRSUSET prsus 'where status
= r on dtc 4 0
```

- Display the PRSUs that depend on a specific PRSU

```
>SELECT prsuid FROM PRSUSET prsus 'where dependson = abc00cb6
```

- Display the load names for all LTCs in the office.

```
>SELECT destid loadname FROM DESTSET ltc
```

- Display all PRSUs that are approved by the customer.

```
>SELECT prsuid custapp FROM prsuset prsus 'where custapp
= y
```

- Display all CM, ISN, SPM, and XPM PRSUs in the office.

```
>SELECT informlist
```

- Display all destination information in the office.

```
>SELECT alldests
```

- Display the load names in each patchable device.

```
>SELECT informload
```

- Display all SPM dests in the office.

```
>SELECT destid FROM DESTSET dests 'where destid = spm
```

- Display the current PRSM response level.

```
>SELECT RESPONSE_LEVEL
```

- Display all current PRSM users and their connection status.

```
>SELECT USER_NAME CONNECTED
```
- Display all current PRSM users, the commands being executed, and the current response level for each current PRSM session.

```
>SELECT USER_NAME COMMAND RESPONSE_LEVEL
```

Managing the PRSM database

This section describes how to manage the PRSM database using the ASSIGN and DBAUDIT commands.

ASSIGN command

ATTENTION

The PRSM ASSIGN command has different syntax and functionality than other ASSIGN commands used on the switch.

The PRSM ASSIGN command can conflict with the table editor ASSIGN command and other ASSIGN commands on the switch.

Use the ASSIGN command to change the values of database fields of PRSUs, destinations, and PRSM options.

The following ASSIGN commands are not enabled for SPM Patching:

- ASSIGN active
- ASSIGN installed

Syntax

Following is the syntax for the ASSIGN command:

```
>ASSIGN <field_name> <field_value> IN <prsm_set_def>
```

```
>ASSIGN <prsm_option> <option_value>
```

The following table lists the parameters and variables for the ASSIGN command.

Table 3-5 Parameters and variables for ASSIGN command (Sheet 1 of 2)

Parameter or Variable	Values	Description
<field_name>	<name>	Specifies the PRSM database field
<field_value>	<value>	Specifies the value in the field

Table 3-5 Parameters and variables for ASSIGN command (Sheet 2 of 2)

Parameter or Variable	Values	Description
<prsm_set_def>		Specifies the PRSU set or DEST set
	PRSUSET <prsu_set_def>	A group of one or more PRSUs.
	DESTSET <dest_set_def>	A group of one or more DESTs
<prsm_option>		Identifies the PRSM option
	RESPONSE_LEVEL	The command output response level of the current PRSM sessions
	DEF_RESP_LEVEL	The default command output response level

Examples

Following are examples of the ASSIGN command:

- Prevent a PRSU from being auto-applied.

```
>ASSIGN custapp n IN prsuset jpp28cb5
```
- Remove a PRSU from the File Audit process.

```
>ASSIGN fileaudit n IN prsuset sch01cb5
```
- Activate an ACT PRSU.

```
>ASSIGN active y IN prsuset sch00cb5
```
- Assign a load to a PM.

```
>ASSIGN upgrade_ld nlg03bc IN destset lgc 0
```
- Change the default format of the command output response to verbose.

```
>ASSIGN def_resp_level verbose
```
- Change the format of command output response for the current session to quiet.

```
>ASSIGN response_level quiet
```

DBAUDIT command

Use the DBAUDIT command to compare the applied PRSU information between the PRSM database and the destination PRSU database. If there are discrepancies between the databases, the PRSM database is updated to reflect the information in the destination PRSU database.

PRSM performs the following tasks during a DBAUDIT:

1. Compares the loadname in the destination to the loadname for the destination in the PRSM CM database and updates the CM database accordingly.
2. Audits SPM loadfiles datafilled in Table PMLOADS and updates the CM database accordingly.
3. Compares PRSUs marked as applied in the CM database but not truly applied in the destination and updates the CM database accordingly.
4. Compares PRSUs marked as applied in the destination with PRSUs not marked as applied in the CM database and updates the CM database accordingly.
5. Compares PRSU information of PRSUs that are applied and known to the CM database. This step includes auditing of the FREEMEM and ACT statuses in SOS-based destinations.
6. Audits the BUILTIN field and verifies that the field is accurate (XPMs and SPMs only).
7. For ISNs, audits the recovery database to determine if corruptions still exist. When ISN recovery database corruptions are found or cleared, PRSM logs are generated.

DBAUDIT can be run manually or automatically. DBAUDIT is run automatically as part of the Status Audit process. The Status Audit is scheduled to be run each night. The DBAUDIT should be run manually when PRSUs are loaded outside of PRSM, such as when PRSUs are loaded as part of patching after loading (PAL) and pre-patched ISN images.

The CM PRSM database cannot be fully updated when a DBAUDIT finds a PRSU applied in a destination that has not been processed by the PRSM database. If this occurs, PRSM will search SFDEV and the devices datafilled in table PADNDEV for the PRSU file. If the file cannot be found in the search, the PRSM database cannot be fully updated.

Syntax

Following is the syntax for the DBAUDIT command:

```
>DBAUDIT [<dest_set_def>]
```

The following table lists the parameters and variables for the DBAUDIT command.

Table 3-6 Parameters and variables for DBAUDIT command

Parameter or Variable	Values	Description
<dest_set_def>		A group of one or more DESTs.

Examples

Following are examples of the DBAUDIT command:

- Compare all destination databases with the PRSM database and update any discrepancies.

```
>DBAUDIT
```

Note: PRSM may take several hours to complete a DBAUDIT of all DESTs in a switch.

- Compare the database of both units of a single PM with the PRSM database and update any discrepancies.

```
>DBAUDIT LGC 0
```

- Compare the databases of all PMs in a destination set with the PRSM database and update any discrepancies.

```
>DBAUDIT samdestset
```

- Compare the databases of all PMs with a specific load with the PRSM database and update any discrepancies.

```
>DBAUDIT 'dests where loadname = mcs05ao
```

LFAUDIT command

Use the LFAUDIT command to update PRSM data before adding a new SP load or entering a new PPXL as an active or backup loadfile in table PMLOADS. LFAUDIT checks the PPXL or SPM loadfile, determines the patch content of the loadfile, and updates the PRSM database with information from the PPXL and its associated PRSU or patch files. Refer to the procedure "Prepare for a manual PM update" from your PM Release Document to perform the LFAUDIT command. If the LFAUDIT command is not performed before the manual update of table PMLOADS, the MAP terminal displays an error message similar to the following example.

```
PRSM data missing.
To correct, enter the PRSM command level.
Then type:
LFAUDIT PPXL ELIO9AX ELIO9AX_980107 S00DPMLOADS
```

LFAUDIT is automatically performed during PMUPGRADE. Refer to the procedure "Prepare for an automated PM update using PMUPGRADE" for instructions on PMUPGRADE.

Syntax

Following is the syntax for the LFAUDIT command:

```
>LFAUDIT PPXL <loadname> <lfname> <device>
```

The following table lists the parameters and variables for the LFAUDIT command.

Table 3-7 Parameters and variables for LFAUDIT command

Parameter or Variable	Values	Description
KEYWORD	PPXL	The keyword that indicates this is a PPXL loadname.
<loadname>	loadname	The base loadname of the PPXL.
<lfname>	loadfilename	The complete filename of the PPXL with the date extension.
<device>	disk device or SFDEV	The disk device where the loadfile resides.

Example

Following is an example of the LFAUDIT command and MAP response. The PRSM database is successfully updated with data from the PPXL.

```
>LFAUDIT PPXL ELI09AX ELI09AX_980107 S00DPMLOADS
```

Audit ok.

Manually managing automated processes

This section describes how to manage automated PRSM processes with the AUTOPROC and FILEAUDIT commands.

AUTOPROC command

Use the AUTOPROC command to start, stop, delay, or query the PRSM automated processes.

Syntax

Following is the syntax for the AUTOPROC command:

```
>AUTOPROC <process> START
```

```

>AUTOPROC <process> STOP
>AUTOPROC <process> DELAY
>AUTOPROC <process> QUERY
>AUTOPROC ALL STOP
>AUTOPROC ALL DELAY
>AUTOPROC ALL QUERY

```

The following table lists AUTOPROC parameter definitions.

Table 3-8 Parameters and variables for AUTOPROC command

Parameter	Value	Description
<process>	AUTOAPP	The Autoapply process. Log monitoring is not performed when AUTOAPP is started with this command.
	AUTOINST	The Autoinstall process. Log monitoring is not performed when AUTOINST is started with this command.
	STATAUD	The Status Audit process
	FILEAUD	The File Audit process
ALL		All the PRSM automated processes. The START operation is not supported for this value. If a user wishes to start all automated operations, alter tables AUTOOPTS and AUTOPRSU to enable the PRSM scheduler.
START		Start an automated process. This operation is not supported with the PRSM process value ALL.
STOP		Stop an automated process if it is already running.
DELAY		Delay an automated process by one scheduled cycle.
QUERY		Display the current state of an automated process.

Examples

Following are examples of the AUTOPROC command:

- Start the File Audit process.

```
>AUTOPROC fileaud start
```

- Stop the Status Audit process.

```
>AUTOPROC stataud stop
```

- Delay the Auto Apply process.

```
>AUTOPROC autoapp delay
```

- Determine the status of all automated processes.

```
>AUTOPROC all query
```

FILEAUDIT command

Use the FILEAUDIT command to execute the part of the automated File Audit process that checks for unequal PRSU files and searches for missing applied XPM PRSU files. All File Audit alarms are cleared and regenerated based on the condition of the PRSU files. CLEAR is the only option supported on the FILEAUDIT command.

Syntax

Following is the syntax for the FILEAUDIT command:

```
>FILEAUDIT CLEAR
```

Other commands

This section describes how to use the FREEMEM and ISTBAUDIT commands.

FREEMEM command

Use the FREEMEM command to deallocate memory that was displaced by the application of a PRSU.

The following conditions apply to the use of the FREEMEM command:

- The FREEMEM command can only be used on a SOS-based destination.
- Once a PRSU has its memory freed in a destination, the PRSU cannot be removed from that destination.
- Before the memory can be freed, the PRSU must be applied for at least 90 days *and* a restart must have been performed with the PRSU applied in that destination.

Note: This condition does not apply to increments.

- Since the results of the FREEMEM command are permanent, only one PRSU can be freed from one destination with one command. The user cannot specify a set of PRSUs or a set of destinations with the FREEMEM command.

Note: This condition does not apply to increments.

- If a PRSU has its memory freed and the SOS destination is reloaded with an image that does not have the PRSU freed, the PRSU does not automatically have its memory freed again.

The FREEMEM command does not change the status of a PRSU. However, the FREEMEM field in the database tracks whether or not a PRSU can be or has been freed in a destination. The following table lists the values of the FREEMEM field.

Table 3-9 Values of FREEMEM field

Value	Description
NA	The PRSU is not available to have its memory freed in the destination.
AV	The PRSU is available to have its memory freed in the destination.
F	The memory for this PRSU has been freed from its destination.

Syntax

Following is the syntax for the FREEMEM command:

```
>FREEMEM <prsuid> IN <destid>
```

The following table lists the parameters and variables for the FREEMEM command.

Table 3-10 Parameters and variables for FREEMEM command

Parameter or Variable	Values	Description
<prsuid>		A specific PRSU or PRSU set
<destid>		A specific DEST

Example

Following is an example of the FREEMEM command:

- Deallocate the memory used by a PRSU in an ISN destination.

```
>FREEMEM sam00ib5 IN lim 0 0
```

ISTBAUDIT command

Use the ISTBAUDIT command to correct PAL failures in specified XPM destinations. The command corrects PAL failures with the following actions:

1. Runs a DBAUDIT on specific ISTb destinations to sync the destinations with the PRSM database.
2. Removes any extra PRSUs applied to the specific ISTb destinations, which may occur with a PPXL that contained a bad PRSU.
3. Applies any missing PRSUs that were not applied to the ISTb destinations during PAL.
4. Clears the PAL alarm if no additional PRSUs need to be applied or removed.

Note: The ISTBAUDIT command may also be used to apply patches to SPMs. The PAL functionality is not available for SPMs. To simplify manual application of SPM patches after an SPM reload from disk file, use the ISTBAUDIT command.

Syntax

Following is the syntax for the ISTBAUDIT command:

```
>ISTBAUDIT [ <dest_set_def> ]
```

The following table lists the parameters and variables for the ISTBAUDIT command.

Table 3-11 Parameters and variables for ISTBAUDIT command

Parameter or Variable	Values	Description
<dest_set_def>		A group of one or more DESTs

Examples

Following are examples of the ISTBAUDIT command:

- Perform an ISTBAUDIT on a specific XPM destination.

```
>ISTBAUDIT ltc 0 1
```

- Perform an ISTBAUDIT on a specific XPM destination set.

```
>ISTBAUDIT xpmdestset
```

- Perform an ISTBAUDIT on all XPM destinations with a specific load.

```
>ISTBAUDIT 'dests where loadname = eli07bh1
```

- Perform an ISTBAUDIT on all XPM destinations.

```
>ISTBAUDIT XPM
```

- Perform an ISTBAUDIT on all SPM destinations.

```
>ISTBAUDIT SPM
```

Controlling the PRSM session

This section describes how to use the PRSM ABORT and PRSM RECONNECT commands.

PRSM ABORT

Use the PRSM ABORT command to abort PRSM commands before their normal termination. Following is an example of the command:

```
><break>-HX>
```

```
>PRSM ABORT
```

PRSM will terminate the currently running maintenance command at the next available safe stopping point.

PRSM RECONNECT

Use the PRSM RECONNECT command to reconnect a PRSM session that has been disconnected through the break-HX. Following is an example of the command:

```
><break>-HX>
```

```
>PRSM RECONNECT
```

PRSM RECONNECT does not reconnect the session if the session was disconnected by the PRSM ABORT command.

4 Automated processes

This chapter describes the maintenance functions performed automatically by PRSM.

Introduction

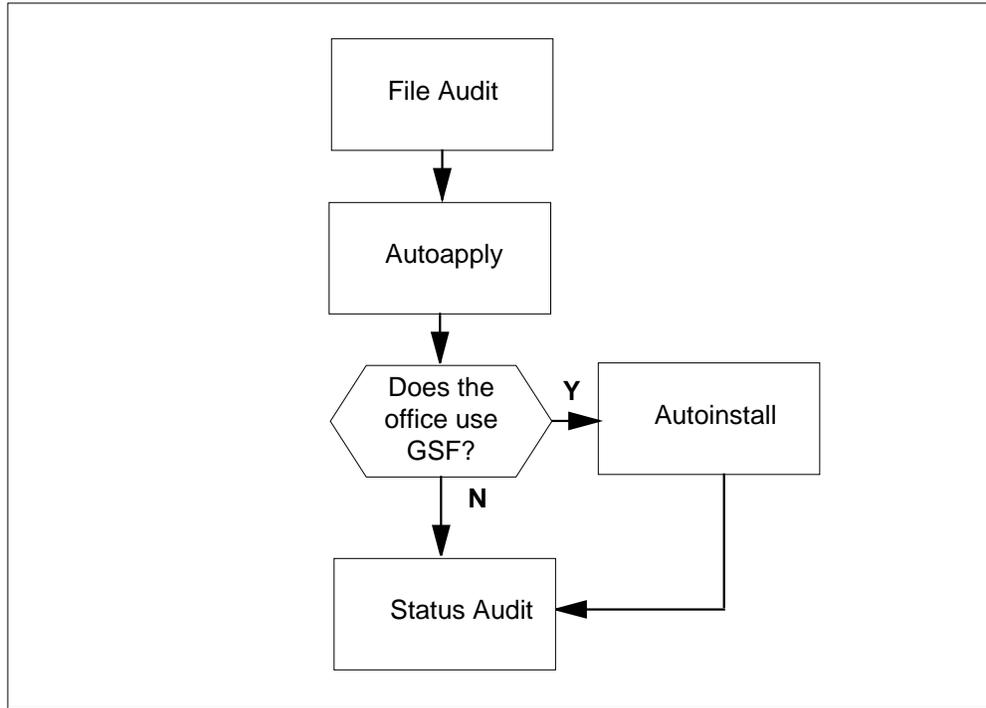
PRSM automates certain PRSU maintenance functions through the following processes:

- File Audit
- Autoapply
- Autoinstall, in offices with GSF
- Status Audit

The automated processes in PRSM are scheduled to run every day to audit PRSU files, automatically apply PRSUs, and audit the PRSM database.

The following figure illustrates the order in which these processes run.

Figure 4-1 Order of PRSM automated processes



The File Audit scans new PRSU files and performs the following tasks:

- validates the PRSUs
- revalidates PRSUs that are in an intermediate state
- verifies file identifiers for PRSUs in the database
- ensures the existence of required SPM and XPM files for removal, reload, and recovery

The Autoapply (AUTOAPP) process runs after the File Audit, if the process is scheduled in table AUTOPRSU. The Autoapply process obtains and applies PRSUs that meet all of the following criteria in the PRSM database:

- PRSUs that are validated (PRSUs that have a status of VA in the PRSM database)
- PRSUs that are customer approved (the CUSTAPP field has a Y value in the PRSM database)
- PRSUs that meet the criteria to be automatically applied (the AUTOAPP field has a Y value in the PRSM database)
- PRSUs that are not on hold from PRSM application, removal, activation or deactivation (the ONHOLD field has an N value in the PRSM database)

- PRSUs that have a PRSU pstate of OK (the PSTATE field has a PRSU_OK value in the PRSM database)
- PRSUs that are removed and the CUSTAPP field has a Y value in the PRSM database

Note: The Autoapply process does not datafill a table with the PRSUs to be applied during the Autoapply session. The PRSUs to be applied are obtained from the PRSM database. The PRSM SELECT command can be used to query the PRSUs to be applied during Autoapply.

The Autoinstall process is used with GSF loads to install PRSU increments that require installation and are installable, based on values in fields INSTREQD and AUTOINST in the PRSM database. Not all increments need to be installed, and any increments installed with the Autoinstall process must first be applied with the APPLY command or the Autoapply process.

The Status Audit process runs after the Autoapply process to ensure PRSM database integrity. The Status Audit first initiates a PRSM DBAUDIT. The DBAUDIT compares the applied PRSU information between the CM PRSM database and the destination PRSU database. Once the DBAUDIT is complete, the Status Audit attempts to find any PRSUs that meet any alarmable conditions.

File Audit process

The PRSM File Audit is an automatic process that is scheduled every day by the PRSM scheduler. The MONDAY to SUNDAY fields in the PRSM automated processes options table, AUTOOPTS, determine when the automatic processes in PRSM begin. The following illustration shows the default datafill for table AUTOOPTS including the MONDAY to SUNDAY fields.

Figure 4-2 AUTOOPTS table default datafill

KEY	FILEALRM	APPROVAL				MONDATA	DURATION	MAXATMPT
DROPSYNC	MONDAY	TUESDAY	WEDNSDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	MNTTIME
AUTOOPTS		Y	N			N	120	\$
N	2300	2300	2300	2300	2300	2300	2300	2310

Finding new PRSUs

During the File Audit process, the File Audit searches for PRSU files on SFDEV and on the devices datafilled in table PADNDEV. It then verifies that the PRSU files found have entries in the PRSM database. If a PRSU entry cannot be found for a PRSU in the PRSM database, the File Audit validates the PRSU. The validation creates a PRSU entry in the PRSM database and

determines if, and where, the PRSU is applicable. Thus, the new PRSU is found.

Note: SFDEV is automatically searched and does not need to be included in the valid devices datafiled in table PADNDEV.

Revalidating PRSUs in an intermediate state

Next, the File Audit process searches through the PRSM database for PRSUs that are at NV (needs validation) status or have a PRSU pstate other than PRSU_OK or NOT_APPLICABLE. A PRSU might have obtained a PRSU pstate of MISSING_NEEDS. This occurs when a PRSU validates for application, but cannot be applied until PRSUs that it requires are applied first. Therefore, the PRSUs that are in an intermediate PRSU pstate are re-validated during the File Audit process to automatically update any PRSU pstate changes that might have occurred.

Verifying database entries

In addition, the File Audit process verifies that the PRSU entries in the database are valid. If the PRSU is marked invalid, the file audit attempts to find a file for the invalid PRSU entry. If a file is found, the PRSU is re-validated with the information from the PRSU file.

Unequal PRSU files

The File Audit process determines if there are any unequal PRSU files found on SFDEV and on the PADNDEV devices. PRSM also checks for unequal PRSUs in either of the following conditions:

- validation and application of SOS PRSUs
- removal of SOS PRSUs when SPAPP = Y
- validation, application, removal, and dbaudit of SPM and XPM PRSUs

If the File Audit process identifies any unequal PRSUs, it generates a PRSM360 log listing the unequal PRSUs and their locations. PRSM will not operate on the PRSUs because it cannot determine which PRSU file to use.

Finding SPM and XPM PRSU files

The File Audit checks for required SPM and XPM PRSU files. The actual old and new code segments exist in both SPM and XPM PRSU files and are used during the application/removal process.

Also, the XPM PRSU files must remain stored on a device so that, if necessary, the PRSU can be found and applied after loading the XPM during recovery. The File Audit process ensures that any of the XPM PRSUs needed to recover an XPM are present on a valid DMS device.

When an SPM or XPM PRSU that is marked in the database as AOR = Y cannot be found, a search of active PMLOADS devices (devices datafilled as ACTVOL in table PMLOADS) is done. Any SPM or XPM PRSU files with AOR = Y that are not found by the search are logged in the PRSM360 log.

If an SPM or XPM PRSU file is validated on a device other than one of these devices, the PRSU file must remain on that device. If moved, the PRSU file must be moved to a valid File Audit search device so PRSM can locate the file. The File Audit lists missing SPM or XPM PRSU files in the PRSM360 log, and flags the missing SPM or XPM PRSUs in the PRSM database through the FILEALARM flag.

The datafillable option, FILEALRM, in table AUTOOPTS, is used to ensure that required SPM or XPM PRSU files are present on the switch. When the FILEALRM field is set to Y, the File Audit generates a MINOR EXT alarm if it cannot find required SPM or XPM PRSU files. If FILEALRM is set to N, no alarm is generated. Figure 4-2 shows the datafill for table AUTOOPTS including the FILEALRM field.

Alarming and logging missing SPM and XPM PRSU files can be turned off using the FILEAUDIT flag in the PRSM database. The PRSM FILEAUDIT flag can be assigned to N for specific SPM or XPM PRSU files. This keeps the missing SPM and XPM PRSU files from causing an EXT alarm and from being logged in the PRSM360 log.

When the FILEAUDIT flag is assigned to N for a specific PRSU, the PRSU does not cause the EXT alarm to be generated, and the PRSU is not logged in the PRSM360 log. However, the FILEALARM flag is set against the missing SPM or XPM PRSU even if the FILEAUDIT flag is set to N.

Autoapply process

The Autoapply process (AUTOAPP) runs after the File Audit process. However, several factors specify how, if, and when the Autoapply process runs. For example, field MTWTFSS in table AUTOPRSU indicates the days of the week that the Autoapply process will run.

The PRSM AUTOPROC command can also affect when the Autoapply process is scheduled to run. For example, the AUTOPROC AUTOAPP DELAY command delays the currently scheduled Autoapply process until the next scheduled Autoapply process time.

PRSM Autoapply is an automatic process that is scheduled to obtain a set of PRSUs and apply them in the office. However, several external factors determine exactly how the Autoapply process performs with regard to:

- when the Autoapply process runs
- how long the Autoapply process is permitted to apply PRSUs or the maximum number of PRSUs the Autoapply process is permitted to apply
- if the Autoapply process monitors logs before and after PRSU application
- if the Autoapply process drops sync on the CM prior to PRSU application
- which PRSUs to automatically apply

The time specified by the day of the week fields (MONDAY to SUNDAY) in table AUTOOPTS specifies the start time of the PRSM automated processes. When scheduled, the Autoapply process runs directly after the daily File Audit process. A seven boolean value field, MTWTFSS in table AUTOPRSU, indicates the days of the week that the Autoapply process runs. If all booleans are set to N, the Autoapply process will not run.

The AUTOPROC AUTOAPP DELAY command can also affect when the Autoapply process runs. The AUTOPROC AUTOAPP DELAY command delays the scheduled Autoapply process by one scheduled cycle.

Therefore, the Autoapply process runs after the File Audit process on the days of the week specified in the MTWTFSS field unless manually delayed.

The DURATION field in table AUTOOPTS determines how long the Autoapply session can run. Once the DURATION time period is exceeded, the Autoapply process finishes executing the current command and then stops. The application is stopped at a different point for XPMs and ISNs.

For example, if the DURATION time limit is exceeded during the application of a PRSU to a set of XPMs, the Autoapply process finishes applying the PRSU to all applicable destinations and then stops.

On the other hand, if the DURATION time limit is exceeded during the application of a PRSU to a set of ISNs, the Autoapply process finishes applying the PRSU to that ISN, and then stops.

If there are PRSUs left to be applied after reaching the DURATION time limit, the PRSUs are applied during the next scheduled Autoapply process.

The MAXATMPT field in table AUTOOPTS determines the maximum number of PRSUs that Autoapply attempts to apply.

The MONDATA field determines if the Autoapply process monitors DMS logs before attempting to apply PRSUs and after PRSU application. For more information on the log monitoring portion of the Autoapply process see "Log Monitoring" in this chapter.

Prior to automatically applying any PRSUs, the Autoapply process checks to make sure that the CM is in synchronization. If the CM is out-of-sync, the Autoapply process fails and a PRSM380 failure log is generated.

The DROPSYNC field in table AUTOOPTS determines if the Autoapply process drops synchronization on the CM before applying CM PRSUs. After the CM is out-of-sync, the PRSUs are only applied to the active side of the CM.

Note: Field DROPSYNC is called field SPLIT in offices using the XA-Core platform.

Therefore, the applied CM PRSUs are not present in the inactive side until CM synchronization is established. If DROPSYNC is enabled and POSTMON fails, the switch is not synced.

If the DROPSYNC field is set Y, the Autoapply process drops sync on the CM before automatically applying CM PRSUs. If switch sanity is maintained after CM PRSU application, the Autoapply process automatically synchronizes the CM. The Autoapply process does not apply any CM PRSUs if the DROPSYNC field is Y, but the Autoapply process is unable to drop sync.

If the DROPSYNC field is N, CM synchronization is maintained during Autoapply CM PRSU application.

Once scheduled, the Autoapply process obtains and applies the set of PRSUs that meet all of the following criteria in the PRSM database:

- PRSUs that are validated (PRSUs that have a status of VA in the PRSM database)
- PRSUs that are customer approved (the CUSTAPP field has a Y value in the PRSM database)
- PRSUs that meet the criteria to be automatically applied (the AUTOAPP field has a Y value in the PRSM database)
- PRSUs that are not on hold from PRSM application, removal, activation or deactivation (the ONHOLD field has an N value in the PRSM database)
- PRSUs that have a PRSU pstate of OK (the PSTATE field has a PRSU_OK value in the PRSM database)
- PRSUs that are removed and have the CUSTAPP field set to Y

During the first validation of a new PRSU, PRSM sets field CUSTAPP in the PRSM database to the value of field APPROVAL in table AUTOOPTS. For example, if field APPROVAL is set to N, then field CUSTAPP for new PRSUs will be set to N and new PRSUs will not be picked up by the Autoapply process.

Log monitoring

One datafillable option in table AUTOOPTS determines whether logs are monitored as part of the automated processes. The MONDATA field determines if the Autoapply process monitors logs, determines how long the Autoapply process monitors logs (PREMON and POSTMON), and indicates the limit of the change in logs that can prevent Autoapply from running.

When the appropriate AUTOOPTS options are set for log monitoring, the Autoapply process monitors logs before and after automatically applying PRSUs. The log monitoring interface obtains a count of the logs specified in the AUTOMON table at the PREMON time specified prior to running. Autoapply then takes another count of the logs after the PREMON time and determines if the MONLIMIT was exceeded.

If the logs are still within the MONLIMIT, the Autoapply process begins to run. If the MONLIMIT is exceeded, the Autoapply process is cancelled and a PRSM381 failure log is generated.

Note: The count is a log peg count or the number of times the log has been generated since the last restart.

After the Autoapply process applies PRSUs, the log monitoring interface takes another count of the specified logs. After the POSTMON time period, another log count is taken. If the MONLIMIT is exceeded, a PRSM381 failure log is generated along with the POST_AUTOPROC_SANITY alarm. Otherwise, the session completes.

A snapshot of the AUTOMON table is taken prior to beginning the Autoapply process. This snapshot of the logs is used throughout the processing session. Therefore, the snapshot of the AUTOMON logs is not updated to reflect logs that are added/removed from the AUTOMON table while the Autoapply process is running.

The formulas shown in the following illustration are used to calculate and compare the log counts obtained during the Autoapply log monitoring process.

Figure 4-3 PRSM log monitoring formulas

$$(A_{t_2} - A_{t_1}) \times \frac{60}{(t_2 - t_1)} = C_1 \frac{\text{logs}}{\text{hr}}$$

$$\frac{A_{t_1}}{24 \times D} \times \left(1 + \frac{N}{100}\right) = C_2 \frac{\text{logs}}{\text{hr}}$$

A_{t_1} Log peg count at time t_1

A_{t_2} Log peg count at time t_2

t_1 Initial log peg count time

t_2 Log peg count time after monitor time period

D Number of days that log pegs have been taken

N The MONLIMIT value

C_1 Monitored log peg counts per hour

C_2 Calculate log peg count average per hour increased by the allowed MONLIMIT

The log peg count taken at the start of monitoring is averaged over the time that the log system has taken log peg counts.

Note: The log system peg counts are reset over all restarts. Therefore, the average (logs per hour) is determined by using the count of number of days since a restart occurred.

The MONLIMIT value associated with the MONDATA field from the AUTOOPTS table is the percentage increase in log peg counts allowed during monitoring. Therefore, the average per hour count is increased by the MONLIMIT. The resulting C_2 value is shown in Figure 4-3, "PRSM log monitoring formulas."

For example, the log system has been taking log peg counts for three days. At the start of log monitoring the log peg count is 576. If a MONLIMIT value is

20 (20 percent increase in the log count is allowed), the average per hour is calculated to be 9.6 or 10 logs allowed per hour.

Next, the count obtained at the start of monitoring is subtracted from the new log peg count. This value is normalized to obtain the count per hour. For example, if there is a count of 3 logs at 12:00 and 5 logs at 12:15 (assuming the pre/post monitoring value is T15), the normalized count is 8 logs per hour. This is the C1 value shown in the illustration "PRSM log monitoring formulas."

Compare the C1 and C2 logs per hour values. If C1 is greater than C2, monitoring has failed. The log count has been exceeded and the Autoapply process quits or never attempts to run. If C1 is less than C2, the process continues.

This method is used to check all logs datafiled in the AUTOMON table snapshot. The Autoapply log monitoring fails if any of the logs exceed their allotted peg counts.

Autoinstall process

The Autoinstall process is used with GSF loads to install PRSU increments that require installation and are installable, based on values in fields INSTREQD and AUTOINST in the PRSM database. Not all increments need to be installed, and any increments installed with the Autoinstall process must first be applied with the APPLY command or the Autoapply process.

If log count limits are met, Autoinstall performs the following tasks.

- Checks that the CM is in sync. If the CM is out of sync, Autoinstall verifies that Autoapply was responsible for dropping sync if Autoapply is scheduled to run.
- Installs autoinstallable increments in the order established when the increments were applied. If at least one increment in that order is not autoinstallable, any increments applied after the non–autoinstallable increment will not be automatically installed until that increment is manually installed.

PREMON occurs before the Autoinstall process, and POSTMON occurs after the Autoinstall process.

Status Audit process

The PRSM Status Audit is an automatic process that is scheduled every day by the PRSM scheduler. The MONDAY to SUNDAY fields in the PRSM options table AUTOOPTS determine when the automatic processes in PRSM begin.

The Status Audit runs after the Autoapply process. During the Status Audit, the PRSM database and peripheral databases are audited. The Status Audit also detects any differences in PRSU content between multiple DEST units. Once audited, any data discrepancies or conditions are flagged through PRSM Status Audit flags, EXT alarms, or a summary log.

Status Audit DBAUDIT

The Status Audit first initiates a PRSM DBAUDIT. The DBAUDIT compares PRSM database data with DEST data and PRSU data and, if necessary, updates the PRSM database information to match the DEST information.

After the DBAUDIT is complete, the Status Audit attempts to find any PRSUs that meet any alarmable conditions that are specified and enabled in table PRSMALRM.

Refer to the section on the DBAUDIT command in chapter 3, "Advanced manual functions" for more information on DBAUDIT. Refer to chapter 5, "Tables" to for more information on table PRSMALRM.

Status Audit EXT alarms

After the DBAUDIT, the Status Audit process scans for any alarmable PRSM database conditions. The alarmable PRSM database conditions are datafilled in table PRSMALRM. PRSUs that meet any of the alarm conditions in table PRSMALRM cause the appropriate EXT alarm to be generated.

The PRSUs that meet the alarm conditions are also individually flagged in the PRSM database. The Status Audit sets the STATALARM flag to Y for the PRSUs that meet the alarmable condition. A PRSM470 summary log is also generated listing the number of PRSUs at each alarmable condition and the PRSUIDs causing the alarms.

A STATAUDIT flag also exists for each PRSU in the PRSM database. The STATAUDIT flag provides a way of keeping certain PRSUs from causing alarms or generating logs during the daily Status Audit.

When the STATAUDIT flag is set to N for a PRSU, the PRSU does not generate an EXT alarm or get listed in the PRSM470 log even if it meets an alarmable condition. However, the STATALARM flag always gets set against an alarmable PRSU even if the STATAUDIT flag is set to N.

5 Tables

This chapter describes the following datafillable tables:

- PRSMOPTS
- PRSMALRM
- AUTOOPTS
- AUTOMON
- AUTOPRSU
- PADNDEV

Table PRSMOPTS

Table PRSMOPTS identifies general user options in PRSM. The following table lists the fields in table PRSMOPTS.

Table 5-1 Fields in table PRSMOPTS (Sheet 1 of 2)

Field	Value	Meaning
DESTMETH	broadcast	PRSM will use the optimal method when applying PRSUs to XPMs.
	no_broadcast	PRSM will apply PRSUs serially to XPMs.
COEXDUR (Note)	number	The time limit before Call Migration is entered. If the time limit is reached but the number of remaining old calls is above the limit set in COEXCMAX, the increment will remain in coexistence until the limit is reached.
Note: Fields COEXDUR, COEXCMAX, and COEXCMIN are available only in offices that use Generic Services Framework (GSF).		

Table 5-1 Fields in table PRSMOPTS (Sheet 2 of 2)

Field	Value	Meaning
COEXCMAX (Note)	number	The maximum limit of old calls that may remain before the switch can enter Call Migration. The switch cannot enter Call Migration until the number of remaining old calls is less than this limit.
COEXCMIN (Note)	number	The minimum limit of old calls that may remain before the switch enters Call Migration
Note: Fields COEXDUR, COEXCMAX, and COEXCMIN are available only in offices that use Generic Services Framework (GSF).		

Table PRSMALRM

Table PRSMALRM identifies alarm conditions audited by the PRSM Status Audit. Each alarm condition is associated with a PRSM set definition. If the audit detects PRSUs meeting that condition, the specified alarm is activated.

The following table lists the fields in table PRSMALRM.

Table 5-2 Fields in table PRSMALRM

Field	Value	Meaning
ALRMNAME	alarm_name	The name of the alarm
ALRMCOND	setdef_name	The name of the PRSM set definition corresponding to this alarm
ALRMSEV	NA MN MJ CR	The alarm severity
ENABLED	Y, N	Allows offices to turn alarms off

Predefined alarms

Several alarm definitions are predefined by PRSM. These predefined alarm definitions are permanent and cannot be deleted. However, you can change the ALRMSEV and ENABLED fields. The following table lists the predefined alarms in table PRSMALRM.

Table 5-3 Predefined alarms in table PRSMALRM (Sheet 1 of 2)

ALRMNAME	ALRMCOND	ALRMSEV	ENABLED
ACT_NOT_APPLIED	ACT_NOT_APP	NA	Y
ACT_NOT_ACT	NOTACTIVATED	NA	Y

Table 5-3 Predefined alarms in table PRSMALRM (Sheet 2 of 2)

ALRMNAME	ALRMCOND	ALRMSEV	ENABLED
ACT_PW_NOT_ACT	ACTPW_NOTACT	NA	Y
DBG_APPLIED	DBG_APP	NA	Y
DBG_NOT_APPLIED	DBG_NOT_APP	NA	Y
DNR_NOT_APPLIED	DNR_NOT_APP	NA	Y
EMG_NOT_APPLIED	EMG_NOT_APP	NA	Y
GEN_NOT_APPLIED	GEN_NOT_APP	NA	Y
LTD_NOT_APPLIED	LTD_NOT_APP	NA	Y
MAN_NOT_APPLIED	MAN_NOT_APP	NA	Y
SRC_NOT_APPLIED	SRC_NOT_APP	NA	Y
OBS_NOT_REMOVED	OBS_NOT_REM	NA	Y
OBE_NOT_REMOVED	OBE_NOT_REM	NA	Y
REMOVED_PRSUS	REM_PRSUS	NA	Y
APPLIED_NOT_INSTALLED	INCNOTINST	NA	Y
UNIT_MISMATCH	UNITMISMATCH	NA	Y

Table AUTOOPTS

Table AUTOOPTS identifies general options for the automated processes in PRSM. The following table lists the fields in table AUTOOPTS.

Table 5-4 Fields in table AUTOOPTS (Sheet 1 of 2)

Field	Value	Meaning
APPROVAL	Y, N	The default value assigned to the CUSTAPP field for autoapplicable PRSUs. If APPROVAL is set to N, then the CUSTAPP field associated with each PRSU to be applied by Autoapply must be manually changed to Y.
DROPSYNC (Note)	Y, N	Indicates whether switch synchronization should be dropped prior to automatically applying PRSUs or installing PRSUs. note: DROPSYNC is renamed SPLIT when running on the XA Core platform.
DURATION	0 to 1380	Indicates the number of minutes after which Autoapply looks for a safe stopping point. A duration of 0 means no time limit.
FILEALRM	Y, N	Indicates if a minor EXT alarm will be generated when the PRSM File Audit cannot find PRSU files for applies SPM or XPM PRSUs.
MAXATMPT	\$ 1-100	The maximum number of PRSUs that Autoapply will attempt to apply during a single session. A value of \$ means that there is no limit to the number of PRSUs that Autoapply attempts to apply.
MNTTIME	hhmm	The approximate time that the Autoapply or Autoinstall process will begin execution during the next running of the PRSM automated processes.
MONDATA	Y, N	Indicates if log monitoring is turned on or off.
PREMON	T15, T30, T45, T60, T90, T120	If MONDATA is Y, PREMON indicates how long logs should be monitored before Autoapply is executed.
Note: Field DROPSYNC is called Field SPLIT in offices using the XA-Core platform.		

Table 5-4 Fields in table AUTOOPTS (Sheet 2 of 2)

Field	Value	Meaning
POSTMON	T15, T30, T45, T60, T90, T120	If MONDATA is Y, POSTMON, indicates how long logs should be monitored after Autoapply is executed.
MONLIMIT	0 to 100	If MONDATA is Y, MONLIMIT indicates the percent increase allowed for logs (specified in table AUTOMON) before log monitoring fails.
MONDAY	hhmm	The time every Monday that the PRSM Scheduler executes the scheduled automated processes
TUESDAY	hhmm	The time every Tuesday that the PRSM Scheduler executes the scheduled automated processes
WEDNSDAY	hhmm	The time every Wednesday that the PRSM Scheduler executes the scheduled automated processes
THURSDAY	hhmm	The time every Thursday that the PRSM Scheduler executes the scheduled automated processes
FRIDAY	hhmm	The time every Friday that the PRSM Scheduler executes the scheduled automated processes
SATURDAY	hhmm	The time every Saturday that the PRSM Scheduler executes the scheduled automated processes
SUNDAY	hhmm	The time every Sunday that the PRSM Scheduler executes the scheduled automated processes
Note: Field DROPSYNC is called Field SPLIT in offices using the XA-Core platform.		

Table AUTOMON

Table AUTOMON identifies which logs are monitored before and after the Autoapply and Autoinstall processes. Log monitoring is only performed if the MONDATA field in table AUTOOPTS is Y. Log monitoring is not performed during a manual invocation of the Autoapply process. Fields in table AUTOMON can be added and deleted at the discretion of the user.

The following table lists the fields in table AUTOMON.

Table 5-5 Fields in table AUTOMON

Field	Value	Meaning
LOGNAME	logname	The name of the log to be monitored
LOGNUM	log_number	The number of the log to be monitored

The following table lists the default values in table AUTOMON.

Table 5-6 Default values in table AUTOMON

LOGNAME	LOGNUM
LOST	101
LOST	104
LOST	105
LOST	106
MM	100
CM	103
CM	111
CM	112
CM	116
CM	119
CM	122
CM	125

Table AUTOPRSU

Table AUTOPRSU identifies scheduling options for the automated processes in PRSM. One tuple is visible for each process enabled on the CM.

The following table lists the fields in table AUTOPRSU.

Table 5-7 Fields in table AUTOPRSU

Field	Value	Meaning
FIELDNAME	fileaud	Identifies the Fileaudit process
	premon	The premonitoring of logs when log monitoring is turned on in table AUTOOPTS
	autoapp	The Autoapply process
	autoinst	The Autoinstall process
	postmon	The postmonitoring of logs when log monitoring is turned on in table AUTOOPTS
	stataud	The Status Audit process
MTWTFSS	YYYYYYY NNNNNNN any Y/N combo	The days of the week, starting with M for Monday, that the automated process will run
<p>Note 1: Only the autoapp and autoinst tuples can have their fields altered. All other fields are read-only.</p> <p>Note 2: The autoinst tuple is available only in offices that use GSF.</p>		

Table PADNDEV

Table PADNDEV is used by PATCH downloading (PADN) application to determine which devices to use for its operations. PRSM automatically looks for PRSU files on SFDEV whenever it looks for PRSUs on the devices specified in PADNDEV.

Table 5-8 Fields in table PADNDEV

Field	Value	Meaning
DEVKEY	1, 2, or 3	Indicates the choice of the corresponding device. Device 1 is the first device searched for PRSU files. Device 2 is the second device searched for PRSU files. Device 3 is the third device searched for PRSU files.
DEVICE	device_name	The name of the device with the PRSU file
<p>Note: It is not necessary to include SFDEV as a PADNDEV device. PRSM checks SFDEV regardless of SFDEV's datafill in table PADNDEV.</p>		

6 Command output

This chapter describes the formats of command output responses that are available through PRSM.

Introduction

PRSM provides a choice of three formats of command output responses:

- QUIET
- BRIEF
- VERBOSE

The format for the command output response may be set as a default format or a format for the current session. The default format applies to all new users and new PRSM sessions. The format for the current session applies only to the current user for the current session.

Use the ASSIGN command with parameters to set the format of the command output response. The following table lists the commands and parameters to set the command output response format.

Table 6-1 Commands to set command output response format

Session	Format	Command
Current	QUIET	<code>>assign response_level quiet</code>
	BRIEF	<code>>assign response_level brief</code>
	VERBOSE	<code>>assign response_level verbose</code>
Default	QUIET	<code>>assign def_resp_level quiet</code>
	BRIEF	<code>>assign def_resp_level brief</code>
	VERBOSE	<code>>assign def_resp_level verbose</code>

Note: Loader warnings will appear regardless of the the response level.

QUIET response format

The following figure illustrates a QUIET response format for the REMOVE command.

Figure 6-1 Example of QUIET response format

```
>ASSIGN RESPONSE_LEVEL QUIET
Ok
>REMOVE ABC00CB7
PTCHTST3.AG01 is being unpatched to PTCHTST3.AE01
Module "C0ABCMOD" has been removed from the package.
WARNING:PROCVAR table entry LS_DEACT_TABLE [512]# 0,target procedure
C0ABC_DE, in C0ABCMOD has been removed.
WARNING:PROCVAR table entry LS_ACT_TABLE [512]# 0,target procedure
C0ABC_AC, in C0ABCMOD has been removed.
Done.
```

BRIEF response format

The following figure illustrates a BRIEF response format for the REMOVE command.

Figure 6-2 Example of BRIEF response format

```
>ASSIGN RESPONSE_LEVEL BRIEF
Ok
>REMOVE ABC00CB7
No Special Application Instructions
ABC00CB7 validates for removal against 1 DEST
PTCHTST3.AG01 is being unpatched to PTCHTST3.AE01
Module "C0ABCMOD" has been removed from the package.
WARNING:PROCVAR table entry LS_DEACT_TABLE [512]# 0,target procedure
C0ABC_DE, in C0ABCMOD has been removed.
WARNING:PROCVAR table entry LS_ACT_TABLE [512]# 0,target procedure
C0ABC_AC, in C0ABCMOD has been removed.
Done.
ABC00CB7 removed from 1 DEST
```

VERBOSE response format

The following figures illustrate a VERBOSE response format for the REMOVE command.

Figure 6-3 Example of VERBOSE response format

```

>ASSIGN RESPONSE_LEVEL VERBOSE
Ok
>REMOVE ABC00CB7

=====
No Special Application Instructions

Attempting to validate for removal the following record...
*****Command Record
LPATCH record: C0ABCMOD, edition AA01, in package SOSBILGE
Note: Package SOSBILGE exists in CM.
*****

Attempting to validate for removal the following record...
*****Command Record

CI CMD record containing 4 CIAPP commands and 2 CIREM commands.
*****

Attempting to validate for removal the following record...
*****Command Record
SPATCH record: PTCHTST3, old edition AE01, new edition AG01
Note: Module PTCHTST3.AG01 is valid and exists in CM.
*****

ABC00CB7 validates for removal against 1 DEST

-----
Attempting to remove the following action...
SPATCH record: PTCHTST3, old edition AE01, new edition AG01
PTCHTST3.AG01 is being unpatched to PTCHTST3.AE01
-----

-----
Attempting to remove the following action...
CI CMD record containing 4 CIAPP commands and 2 CIREM commands.
  ci command: PACKAGE EXCLUDE SOSBILGE C0ABCMOD;
Module "C0ABCMOD" has been removed from the package.
  ci command: UNLOAD C0ABCMOD
WARNING:PROCVAR table entry LS_DEACT_TABLE [512]# 0,target procedure
  C0ABC_DE, in C0ABCMOD has been removed.
WARNING:PROCVAR table entry LS_ACT_TABLE [512]# 0,target procedure
-----

```

Figure 6-4 Example of VERBOSE response format (continued)

```
C0ABC_AC, in C0ABCMOD has been removed.
Done.
-----

-----

Attempting to remove the following action...
LPATCH record: C0ABCMOD, edition AA01, in package SOSBILGE
New modules are unloaded via PRSUs through CI commands.
No action taken.
-----

Verifying all the applicable actions have been patched/unpatched.....
Finished verifying the actions have been patched/unpatched.....

PRSU ABC00CB7 removed from destination CM.
Status updated to R in CM.
ACT status updated to OFF in CM.
ABC00CB7 removed from 1 DEST
=====
```

7 Statuses

This chapter describes the commands and events that can cause PRSM PRSU status changes and discrepancies.

Command status changes

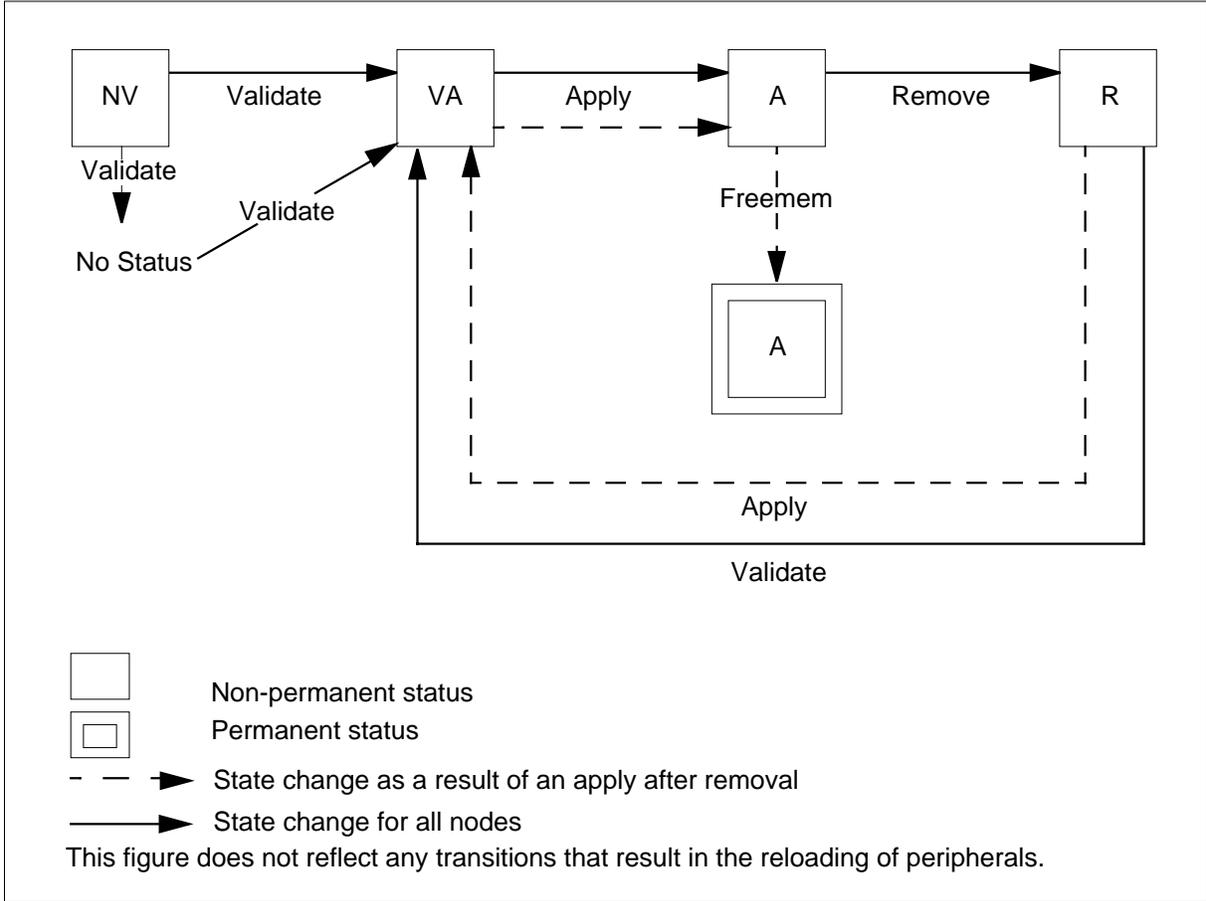
The status of a PRSU is tracked on a per-destination basis. An overall status is not maintained in PRSM. Instead, PRSM maintains the status of each PRSU in every destination. The following table lists the possible PRSU statuses in PRSM.

Table 7-1 PRSU statuses

Id	Status	Description
NV	Needs validation	The PRSU needs to be validated in a DEST. The PRSU may be valid in the DEST, but the PRSU has not been validated against the DEST. A PRSU is at this status until it has been validated against a destination.
VA	Validated for application	The PRSU can apply in a DEST.
A	Applied	The PRSU is currently applied in the DEST.
R	Removed	The PRSU has been removed from the DEST.
NN	Not needed	The PRSU is not needed in the DEST. This status is only seen with the SPM and XPM PRSUs associated with SPM loadfile DESTs and PPXL DESTs.

The following figure illustrates the transition between statuses for a PRSU during selected PRSM operations.

Figure 7-1 Transition between statuses



Peripheral state required for changing PRSU status

Peripherals must be at a certain status before the APPLY and REMOVE commands can be used. These statuses are defined for different types of nodes in the following table.

Table 7-2 Peripheral state prerequisites for applying or removing PRSUs

PRSU	ISN status	XPM status
Source Code	MANB, ISTB, INSV (Note)	MANB, ISTB, INSV
Opcode	Not supported	MANB
Note: When performing a PRSM command on an MS, the MS must be in-service.		

PRSU flags

Numerous flags are available against a PRSU stored in the PRSU databases. These flags can be changed with the PRSM ASSIGN command. These flags can also be used to define sets or seldefs. These flags can be used to create a run time set of PRSUs in a particular condition, such as having the apply on reload (AOR) flag set to Y.

AOR flag

The AOR flag is available for all PRSUs for all PMs in PRSM. However, at this time the AOR flag is explicitly used only for XPMs and enhanced DCHs (EDCH). The AOR field is updated to Y and N automatically by PRSM whenever a PRSU is applied and removed, respectively. The AOR flag can be updated manually by the user with the ASSIGN command.

For ISNs, the AOR flag indicates that the PRSU should be applied in the ISN after the ISN is reloaded. However, ISN PRSUs are not automatically applied after loading like XPM PRSUs. ISNs are SOS-based nodes and support load imaging, in which any PRSUs applied at the time the image is dumped are stored in the loadfile. These applied PRSUs become part of the load and are applied by default when the SOS node is reloaded from the loadfile.

For XPM PRSUs, the AOR flag is used during Patching After Loading (PAL) to determine which PRSUs should be applied. If a PPXL is used to load the XPM, the AOR settings are also used to determine if PRSUs resident in the PPXL are removed once the peripheral is loaded.

The AOR field specifies if a PRSU will be re-applied to an XPM during a manual or system reload. In order to keep a PRSU from applying on reload to a given load, the AOR field must be set to N for all destinations with the given load.

The AOR field can be changed using the ASSIGN command on a per destination basis. However, the PRSU will not be excluded from PAL until the AOR flag is set to N for all destinations with the given load. As long as any one destination has AOR set to Y for a PRSU, the PRSU will be applied on reload, regardless of how the AOR field is set for the other destinations with the same load.

Once a PRSU has been applied to a destination, the AOR field is set to Y for that destination. Once a PRSU is removed from a single destination, the AOR field is set to N for that destination only. However, the PRSU will apply on reload as long as the AOR field is still set to Y for any destination with the given load. Once the PRSU has been removed from all destinations with the given load, the PRSU will not apply on reload.

7-4 Statuses

To exclude a PRSU from PAL for a given load, use the ASSIGN command as in the following example to change the AOR field to N in all destinations with a given load at once. The loadname in the following example is ELI05BC.

```
>assign aor n in 'prsuset XJP53X05 on dests where loadname = ELI05BC
Assignment completed.
```

To exclude a PRSU from PAL for all destinations regardless of the load, the following command can be used.

```
>assign aor n in prsuset XJP53X05
Assignment completed.
```

Note: To block a PRSU from being applied after reload, an ASSIGN command is used to reset the AORs and change the AOR against the PPXL destinations. The PRSU can be removed from all DESTs except the PPXL DEST. Therefore, the AOR flag cannot be reset against the PPXL DEST by using the REMOVE command.

To check the values of the AOR flag on all DTC destinations, type a command similar to that shown in the following example.

```
>select prsuid status aor destname device unit from prsuset prsus +
>'where prsuid = xjp53x05 on dtc
```

PRSUID	ST	AOR	DEST	DEV	UNIT
XJP53X05	VA	N	DTC	0	0
XJP53X05	VA	N	DTC	0	1
XJP53X05	VA	N	DTC	1	0
XJP53X05	VA	N	DTC	1	1
XJP53X05	VA	N	DTC	2	0
XJP53X05	VA	N	DTC	2	1

To query which PRSUs will be applied to the ELI05BC load during PAL, type

```
>select prsuid 'from prsuset prsus where aor = y on dests where '+'
>'loadname = eli05bc
```

Placing a PRSU and DEST on hold

In addition to the different statuses that a PRSU can have, a PRSU can also be placed ONHOLD. If a PRSU is placed ONHOLD, the APPLY, VALIDATE, REMOVE, and FREEMEM commands cannot be used on the PRSU. In

addition, the ASSIGN command cannot be used to activate or deactivate ACT category PRSUs.

A PRSU can be placed ONHOLD or removed from ONHOLD using the ASSIGN command. The following is an example of the syntax to place a PRSU ONHOLD.

```
>ASSIGN ONHOLD y in PRSUSET xza016x07
```

A DEST can be placed on HOLD. If a DEST is placed on HOLD, the APPLY, REMOVE, VALIDATE or FREEMEM commands cannot be used on the DEST. If a PRSU is placed ONHOLD, the APPLY, VALIDATE, REMOVE, and FREEMEM commands cannot be used on the PRSU. In addition, the ASSIGN command cannot be used to activate or deactivate ACT category PRSUs.

A DEST can be placed on HOLD using the ASSIGN command. The following is an example of the syntax used to place a DEST on HOLD

```
>ASSIGN HOLD y in DESTSET ltc 0
```

New host software release

When a new host software release is loaded into the CM and a switch of activity (SWACT) is performed, the PRSM PRSU database will have some discrepancies between the main PRSU database and the PRSU databases maintained in the peripherals. These discrepancies will continue until the main PRSU database resynchronizes itself with the peripheral databases. This synchronization is accomplished by a DBAUDIT that is automatically performed during the ONP post-SWACT process.

All the necessary data that populates the PRSM database is not maintained in the peripheral. During this database synchronization, all of the PRSU files for PRSUs applied in the peripherals must be available on one of the PRSU devices. The PRSU files must be opened to obtain the administrative information about a PRSU.

Pre-patched XPM loads

PRSM supports the tracking of PRSUs against PPXLs. PPXLs are recognized in PRSM as DESTs. PRSM recognizes a PPXL as a DEST once the PPXL files are datafilled in table PMLOADS.

PRSUs included in the PPXL file must be available before the PPXL files can be datafilled in table PMLOADS. Datafilling the PPXL files in table PMLOADS puts the PPXL destination in the PRSM database.

The PRSM File Audit searches for XPM PRSU files, because XPM PAL requires that the XPM PRSU files be always available. XPM PRSU files are

needed for application and removal and to populate the PRSM database. SOS PRSUs are only required for application and to populate the PRSM database.

Use the REPORT command to query PPXL destinations. Following is an example of the command.

```
>REPORT DEST ppxl
```

Following is an example of the MAP response.

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
XCU06X07	19970723	14:34	SRC		A	ESI07BI1	N	PPXL ESI07BI1_961119
XMP06X07	19970723	14:34	SRC		A	ESI07BI1	N	PPXL ESI07BI1_961119
XZA04X07	19970723	14:34	SRC		A	ESI07BI1	N	PPXL ESI07BI1_961119
XZA06X07	19970723	14:34	SRC		A	ESI07BI1	N	PPXL ESI07BI1_961119
XZA16X07	19970723	14:34	SRC		A	ESI07BI1	N	PPXL ESI07BI1_961119

In the previous example, the DESTID includes the word "PPXL" and the PPXL file name "ESI07BI1_961119." Beginning with BASE08, PRSM includes the PPXL file name in the DESTID field.

The PRSUs against the loadname that are included in the PPXL file are listed as A, for Applied. The PRSUs that are applied against the loadname but are not included in the PPXL are listed as NN against the PPXL DEST.

Following is an example of PRSUs listed as NN. In the example, three PRSUs not in the PPXL were applied to a unit with the ELI07BI1 load.

```
>REPORT DEST ppxl
```

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
XCU06X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XMP06X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XZA04X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XZA06X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XZA16X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XCL04X07	00000000	00:00	SRC		NN	ELI07BI1	N	PPXL ELI07BI1_961119
XCL05X07	00000000	00:00	SRC		NN	ELI07BI1	N	PPXL ELI07BI1_961119
XCL06X07	00000000	00:00	SRC		NN	ELI07BI1	N	PPXL ELI07BI1_961119

Following is an example of a report for a specific PPXL.

```
>REPORT DEST ppxl ELI07BI1_961119
```

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
XCU06X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XMP06X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XZA04X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XZA06X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XZA16X07	19970723	14:34	SRC		A	ELI07BI1	N	PPXL ELI07BI1_961119
XCL04X07	00000000	00:00	SRC		NN	ELI07BI1	N	PPXL ELI07BI1_961119
XCL05X07	00000000	00:00	SRC		NN	ELI07BI1	N	PPXL ELI07BI1_961119

To confirm that the PRSUs are applied, perform a REPORT on a DEST with the PRSUs. Following is an example.

>REPORT DEST smu 0 1

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
XJY71x07	19970723	18:46	SRC		A	ESI07BI1	N	SMU 0 1
XJY60x07	19970723	18:46	SRC		A	ESI07BI1	N	SMU 0 1
XJY72x07	19970723	18:46	SRC		A	ESI07BI1	N	SMU 0 1
XJY74x07	19970723	18:46	SRC		A	ESI07BI1	N	SMU 0 1
XJY76x07	19970723	18:45	SRC		A	ESI07BI1	N	SMU 0 1
XJY64x07	19970723	18:47	SRC		A	ESI07BI1	N	SMU 0 1
XJY66x07	19970723	18:49	SRC		A	ESI07BI1	N	SMU 0 1
XJY78x07	19970723	18:51	SRC		A	ESI07BI1	N	SMU 0 1

To determine which PRSUs are applied to both units run a report.

>REPORT DEST ltc 0

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
XCU06X07	19970723	18:46	SRC		A	ELI07BI1	N	LTC 0 0
XCU06X07	19970723	18:46	SRC		A	ELI07BI1	N	LTC 0 1
XMP06X07	19970723	18:46	SRC		A	ELI07BI1	N	LTC 0 0
XMP06X07	19970723	18:46	SRC		A	ELI07BI1	N	LTC 0 1
XZA04X07	19970723	18:45	SRC		A	ELI07BI1	N	LTC 0 0
XZA04X07	19970723	18:47	SRC		A	ELI07BI1	N	LTC 0 1
XZA06X07	19970723	18:49	SRC		A	ELI07BI1	N	LTC 0 0
XZA06X07	19970723	18:49	SRC		A	ELI07BI1	N	LTC 0 1
XZA16X07	19970723	18:49	SRC		A	ELI07BI1	N	LTC 0 0
XZA16X07	19970723	18:49	SRC		A	ELI07BI1	N	LTC 0 1
XCL04X07	19970723	18:47	SRC		A	ELI07BI1	N	LTC 0 0
XCL04X07	19970723	18:47	SRC		A	ELI07BI1	N	LTC 0 1
XCL05X07	19970723	18:47	SRC		A	ELI07BI1	N	LTC 0 0
XCL05X07	19970723	18:47	SRC		A	ELI07BI1	N	LTC 0 1

Another way to get information on PPXLs is to select information from the PRSM database on PPXLs. Use the SELECT command to select the exact information that is needed. In the following example, only one PPXL is datafilled in the switch.

>SELECT destname loadname lfname FROM DESTSET DESTS WHERE +
>'destname = ppxl

```
DEST  LOADNAME  LFNAME
-----
PPXL  ELI07BI1   ELI07BI1_961119
PPXL  ESI07BI1   ESI07BI1_961119
```

The SELECT command can be used to limit information displayed by a select.

```
>SELECT destname loadname lfname FROM DESTSET DESTS WHERE +
>'destid = ppxl esi07bi_961119
```

```
DEST  LOADNAME  LFNAME
-----
PPXL  ESI07BI1   ESI07BI1_961119
```

PPXLs and ONP

After the one-night process (ONP), a nil change is performed on table PMLOADS to update PRSM with PPXL information. If an office encounters patching errors with PPXLs after an ONP, confirm that the PRSM database contains the current set of PPXLs. The current set of PPXLs is the set of PPXLs associated with the loads in the XPMs at the ONP. If the PRSM database does not contain the current set of PPXLs for a load, perform a nil change on the load in table PMLOADS.

SPM Destinations

SPM Loadfile Destinations

SPM permanent PRSUs in an SPM loadfile obtain an applied (A) status against that loadfile destination after the following steps:

1. datafilling the loadfile in table PMLOADS, and
2. executing a DBAUDIT on that loadfile destination.

All PRSUs (both permanent and non-permanent) with the same loadname as the loadfile destination obtain a not needed (NN) status when they are not built into that loadfile. Note that non-permanent PRSUs are not built into SPM loadfiles; therefore, those PRSUs always have the NN status against loadfile destinations.

SPM Peripheral Destinations

SPM permanent PRSUs in an SPM loadfile obtain an applied (A) status against a SPM peripheral destination after the following steps:

1. loading the peripheral with a SPM loadfile, and
2. executing a DBAUDIT on that peripheral destination.

Permanent PRSUs with the same loadname as the peripheral destination obtain a not needed (NN) status when they are no longer applied (via the loadfile) in the peripheral.

Non-permanent PRSUs (patches) with the same loadname as the peripheral get statuses that correspond to the PRSM commands used to manage them (VALIDATE command sets to VA, APPLY command sets to A, and REMOVE command sets to R).

Using the Report command with SPM PRSUs

To confirm that the PRSUs are applied, perform a REPORT on a DEST with the SPM load. Following is an example.

```
>REPORT dest spm 0 cem 0
```

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
SNT000200000	19970718	16:46	GEN		A	CEM0002	N	SPM 0 CEM 0
SNT000200001	19970718	16:46	GEN		A	CEM0002	N	SPM 0 CEM 0
SNT000200002	19970718	16:46	GEN		A	CEM0002	N	SPM 0 CEM 0
SNT000200003	19970718	16:46	GEN		A	CEM0002	N	SPM 0 CEM 0
SNT000200004	19970718	16:46	GEN		A	CEM0002	N	SPM 0 CEM 0
SNT000200005	19970718	16:46	GEN		A	CEM0002	N	SPM 0 CEM 0
SNT0001A0004	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 CEM 0
SNT0001A0041	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 CEM 0
SNT0001C00C0	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 CEM 0
SNT0001C00B3	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 CEM 0
SCB00S13	20000210	17:08	GEN		R	CEM0000	N	SPM 0 CEM 0

The following is another example of using the REPORT command on a DEST with the SPM load:

```
>REPORT dest spm 0 dlc 0
```

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
SNT000200000	19991229	16:46	GEN		A	CEM0002	N	SPM 0 DLC 0
SNT000200001	19991229	16:46	GEN		A	CEM0002	N	SPM 0 DLC 0
SNT000200002	19991230	16:46	GEN		A	CEM0002	N	SPM 0 DLC 0
SNT000200003	19991230	16:46	GEN		A	CEM0002	N	SPM 0 DLC 0
SNT000200004	19991230	16:46	GEN		R	CEM0002	N	SPM 0 DLC 0
SNT000200005	19991230	16:46	GEN		A	CEM0002	N	SPM 0 DLC 0
SNT0001A0004	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 DLC 0
SNT0001A0041	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 DLC 0
SNT0001C00C0	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 DLC 0
SNT0001C00B3	20000104	16:46	GEN		VA	CEM0002	N	SPM 0 DLC 0
DAA66S13	20000209	09:46	DBG		R	DLC0013	N	SPM 0 DLC 0
DEE66S13	20000209	10:01	GEN		R	DLC0013	N	SPM 0 DLC 0

The REPORT command can also be used on SPM destinations that have the same SPM PRSUs applied to the SPM loadfile DEST. Following is an example.

7-10 Statuses

```
>REPORT dest spmload cem0002_000000
```

PRSUID	STATDATE	STATT	CAT	ACT	ST	LOADNAME	H	DESTID
SNT000200000	19970718	16:46	GEN		A	CEM0002	N	CEM0002_000000
SNT000200001	19970718	16:46	GEN		A	CEM0002	N	CEM0002_000000
SNT000200002	19970718	16:46	GEN		A	CEM0002	N	CEM0002_000000
SNT000200003	19970718	16:46	GEN		A	CEM0002	N	CEM0002_000000
SNT000200004	19970718	16:46	GEN		A	CEM0002	N	CEM0002_000000
SNT000200005	19970718	16:46	GEN		A	CEM0002	N	CEM0002_000000

Another way to get information on SPM PRSUs is to select information from the PRSM database. Use the SELECT command to select the exact information that is needed. Following are two examples.

```
>SELECT destid from destset dests where destid ' = spmload
```

```
DESTID
-----
SPMLOAD CEM0002_000000
SPMLOAD CEM0002_000001
SPMLOAD CEM0002_000002
SPMLOAD CEM0002_000003
SPMLOAD CEM0002_000004
SPMLOAD CEM0003_000000
SPMLOAD CEM0004_000000
```

```
>SELECT prsuid destid from prsuset prsus where 'status = a +
```

```
>on spmload fldwidth prsuid 20
```

PRSUID	DESTID
NTI000200000SPMLOAD	CEM0002_000000
NTI000200001SPMLOAD	CEM0002_000000
NTI000200002SPMLOAD	CEM0002_000000
NTI000200003SPMLOAD	CEM0002_000000
NTI000200004SPMLOAD	CEM0002_000000
NTI000200005SPMLOAD	CEM0002_000000

8 Database fields

This chapter describes the selectable fields used by PRSM in the PRSM PRSU and DEST databases.

PRSU fields

The following table lists the names of the PRSU fields available through PRSM. The table provides possible values for each field and a description of the field.

Note: The table does not include blank values. A blank value in a field indicates that the field is not applicable.

Table 8-1 PRSU fields (Sheet 1 of 8)

Name	Values	Description
1stappdate	<yymmdd> (Note 1) <i>or</i> <yymmdd>	The date a PRSU was first applied to any DEST within an office
1stapptime	<hhmm>	The time a PRSU was first applied to any DEST within an office
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format <code>yyyymmdd</code>.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

Table 8-1 PRSU fields (Sheet 2 of 8)

Name	Values	Description
1stremdate	<yymmdd> (Note 1) <i>or</i> <yymmdd>	Indicates the date a PRSU was first removed from any DEST within an office
1stremtime	<hhmm>	Indicates the time a PRSU was first removed from any DEST within an office
1stvaldate	<yymmdd> (Note 1) <i>or</i> <yymmdd)	The date a PRSU was first validated in any DEST within an office
1stvaltime	<hhmm>	The time a PRSU was first validated in any DEST within an office
actdate (Note 2)	<date>	The date that the PRSU was activated or deactivated
actindex	<num>	Numeric index assigned to ACT PRSUs
active (Note 2)	Y, N (Note 3)	Indicates if a PRSU is activated on a particular DEST
acttime (Note 2)	<fme>	The time that the PRSU was activated or deactivated
aor (Note 2 and 4)	Y, N (Note 3)	Indicates if the PRSU is applied on reload in a specific DEST
appdate (Note 2)	<yymmdd> (Note 1) <i>or</i> <yymmdd)	Date which a PRSU was applied in a DEST
apphist (Note 2)	Y, N	Indicates if the PRSU has been applied to a particular DEST
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyymmdd.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

Table 8-1 PRSU fields (Sheet 3 of 8)

Name	Values	Description
apptime (Note 2)	<hhmm>	Time which a PRSU was applied in a destination
autoapp	Y, N	Indicates if the PRSU can be automatically applied
autoinst	Y, N	Indicates if an increment PRSU can be automatically installed
builtin (Note 2)	Y, N	Indicates if the PRSU was built into the load
category		Category assigned to the PRSU
	GEN	General
	EMG	Emergency
	ACT	Activatable
	LTD	Limited
	OBS	Obsolete
	OBE	Obsolete emergency
	OBR (Note 5)	Obsolete replaced
	DBG	Debug
	DNR	Do not reapply
	SRC	Source code patch, for XPMs only
	MAN	Manual or opcode patch, for XPMs only
	???	None
class		Class assigned to the PRSU
	C	Change
	F	Feature
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyyymmdd.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

Table 8-1 PRSU fields (Sheet 4 of 8)

Name	Values	Description
	I	Increment
	S	Service
	-	None
cmd	Y, N	Indicates if PRSU contains CI commands
custapp	Y, N (Note 3)	Indicates if the PRSU is approved for application by AUTOAPP
dependson	<prsuId>	Indicates the PRSUs that the specified PRSU depends on
edition	<xx##>	Edition codes of the modules impacted by the PRSU
filealarm	Y, N	Indicates whether the PRSU meeting the alarmable condition will cause a log or an alarm to be generated. This field is set automatically by PRSM.
fileaudit	Y, N (Note 3)	Determines whether the File Audit generates a log and an alarm for a PRSU which meets an alarmable condition.
freemem (Note 2)		FREEMEM status associated with a PRSU on a DEST
	NA	Not available
	AV	Available
	F	Freed
incrstatus		Increment status of the PRSU
	??	None
	UI	Uninstalled
	OM	Object migration
	CO	Coexistence

Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyymmdd.

Note 2: The value for this field can vary, based on the DEST.

Note 3: This value can be changed with the ASSIGN command.

Note 4: Refer to AOR information in the chapter "Statuses" in this document.

Note 5: OBR category is no longer supported by PRSU creation tools.

Table 8-1 PRSU fields (Sheet 5 of 8)

Name	Values	Description
	CM	Call migration
	CL	Object cleanup
	IN	Installed
	BO	Backed out
	NA	Not applicable
installed	Y, N (Note 3)	Specifies if an increment PRSU is installed
insdate	<date>	Date a PRSU was installed in a DEST
instime	<fme>	Time a PRSU was installed in a DEST
instreqd	Y, N	Indicates if installation is required on an increment PRSU
label	<label>	Label of the modules required by the PRSU
man	Y, N	Indicates if the PRSU is a manual PRSU
matemismatch (Note 2)	Y, N	Indicates if both units have matched PRSUs at the same status. A blank indicates no mate.
matestat (Note 2)		Status of a PRSU in the mate DEST unit. A blank indicates no mate.
	NV	Needs validation
	VA	Validated for application
	A	Applied
	R	Removed
module	<module>	Lists the modules impacted by the PRSU
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyymmdd.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

Table 8-1 PRSU fields (Sheet 6 of 8)

Name	Values	Description
onhold	Y, N (Note 3)	<p>Indicates if a PRSU is on hold. When a dest is on hold, the commands APPLY, REMOVE, FREEMEM, and ASSIGN for activation/deactivation (if the PRSU has an ACT category) cannot act on the PRSU. The command VALIDATE cannot act on \$PATCH PRSU files, but it can act on \$DF PRSU files.</p> <p>If you attempt a to perform one of these commands while the PRSU is on hold, PRSM responds with a message similar to the following message.</p> <pre>*WARNING:Destination LTC 0 0 is on hold and will not be part of a set to process.</pre> <p>All destinations on HOLD. Cannot complete the requested action.</p>
perm	Y, N	Indicates permanent (delivered in the load file) or non-permanent (delivered in a patch file) PRSUs.
precord	<prsuid>	For CM PRSUs, the software layer id prerequisite PRSU. For XPM PRSUs any required PRSU
processor		Target processor assigned with the PRSU
	M68K	CM processor 68k family
	M88K	CM processor 88k family
	ISN	ISN processor
	XPM	XPM processor
	PPC	CM processor PowerPC family
	SPM	SPM processor
	????	None
prsuid	<prsuid>	Name of the PRSU
prsuload	<loadname>	Lists the loadnames that are impacted by the PRSU
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyymmdd.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

Table 8-1 PRSU fields (Sheet 7 of 8)

Name	Values	Description
pstate (Note 2)	prsu_ok missing_needs not_applicable prsu_failure	The state of the PRSU with respect to the destination
release	<num>	The release number of the PRSU
remdate (Note 2)	<yymmdd> (Note 1) <i>or</i> <yymmdd>	The date a PRSU was removed in a DEST
remhist (Note 2)	Y, N	Indicates if the PRSU has been removed from a particular DEST
remtime (Note 2)	<hhmm>	The time a PRSU was removed in a DEST
restart	NONE WARM COLD RELOAD	Indicates the type of restart required for the PRSU to become fully functional No restart, SWACT required Warm restart, SWACT required Cold restart, SWACT required Reload restart required
special	Y, N	Indicates if the PRSU has special application instructions
src	Y, N	Indicates if the PRSU is a source PRSU
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyymmdd.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

Table 8-1 PRSU fields (Sheet 8 of 8)

Name	Values	Description
statalarm	Y, N	Indicates if the PRSU failed the status audit. This field is set automatically by PRSM.
stataudit	Y, N (Note 3)	Indicates if the PRSU is included in the status audit
statdate (Note 2)	<yymmdd> (Note 1) <i>or</i> <yymmdd)	Date a PRSU changed to the current status in a DEST
stattime (Note 2)	<hhmm>	Time a PRSU changed to the current status in a DEST
status (Note 2)	NV VA A R NN	Indicates the status of a PRSU on a particular DEST Needs validation Validated for application Applied Removed Not needed
valdate (Note 2)	<yymmdd> (Note 1) <i>or</i> <yyyymmdd)	Date of PRSU validation in a DEST
valtime (Note 2)	<hhmm>	Time of PRSU validation in a DEST
<p>Note 1: Six character format interprets the century as 19 when year is greater than or equal to 76; otherwise, interprets century as 20. Output is always eight characters in the format yyyymmdd.</p> <p>Note 2: The value for this field can vary, based on the DEST.</p> <p>Note 3: This value can be changed with the ASSIGN command.</p> <p>Note 4: Refer to AOR information in the chapter "Statuses" in this document.</p> <p>Note 5: OBR category is no longer supported by PRSU creation tools.</p>		

DEST fields

The following table lists the names of the DEST fields available through PRSM. The table provides possible values for each field and a description of the field. The destid makes the entry unique.

Table 8-2 DEST fields (Sheet 1 of 2)

Name	Values	Description
actlevel	<num>	Indicates the number of ACT PRSUs activated on a DEST
appllevel	<num>	Indicates the number of PRSUs applied in a DEST
destclass	<num>	PRSM DEST identifier for internal PRSM use
destid	<dest_id>	Complete name of a DEST, including names, devices, and units. The destid field simplifies queries by allowing the user to query on one field rather than three fields. A wildcard (*) can be used with this field.
destname	<num>	Name of the DEST or node
deststat		Maintenance status of DEST
	<blank>	
	ManB	Man busy
	Offl	Offline
	OK	OK
	SysB	System busy
	Unequip	Unequipped
device	<num>	DEST device number or plane
hold (Note)	Y, N	Indicates if the DEST is on hold. When a DEST is on hold, the commands APPLY, REMOVE, FREEMEM, and ASSIGN for activation/deactivation cannot act on any of the PRSUs in the DEST. The command VALIDATE cannot act on \$PATCH PRSU files, but it can act on \$DF PRSU files.
lfname	<loadfilename>	The name of the loadfile with applied PRSUs
loadname	<loadname>	The current load in the DEST
unit	<0, 1, 2, 3>	The DEST unit number or shelf. The unit number is 0 or 1. The ENET shelf number is 0, 1, 2, or 3.
Note: Value that can be changed with the ASSIGN command.		

Table 8-2 DEST fields (Sheet 2 of 2)

Name	Values	Description
unitstate	<act, inact,??>	The DEST state indicator
upgrade_ld (Note)	<loadname>	The upgrade loadname for the DEST. Some DESTs support changing the upgrade_ld loadname in the DEST through the ASSIGN command. However, the assigned loadname must match the LOAD value in the DEST inventory table tuple.
Note: Value that can be changed with the ASSIGN command.		

9 Predefined set definitions

This chapter describes the set definitions that are predefined by PRSM.

Introduction

A set definition is group of definitions stored in a single name. With this single name, a user can support a large group of PMs or PRSUs with a few commands.

A predefined set definition is a set definition created by Nortel Networks and stored in PRSM. These predefined set definitions are permanent and cannot be deleted.

PRSU set definitions

The following table lists the PRSU set definitions predefined by PRSM.

Note: Capital letters, such as "PRSUS WHERE", indicate keywords. Lower case letters, such as "category = act", indicate variables.

Table 9-1 Predefined PRSU set definitions (Sheet 1 of 6)

Name	Set definition	Description
activatable	PRSUS WHERE category = act	Set of all PRSUs that are activatable in the office
activated	PRSUS WHERE category = act active = y ON sos	Set of all PRSUs that have been activated in the office
actpw_notact	PRSUS WHERE category = act actindex >= 256 active = n	Set of PRSUs that are activatable, not activated, and require a password to activate
act_not_app	PRSUS WHERE status = va ON sos PRSUS WHERE status = r ON sos PRSUS WHERE status = nv ON sos & PRSUS WHERE category = act	Set of PRSUs that are activatable, but not applied

9-2 Predefined set definitions

Table 9-1 Predefined PRSU set definitions (Sheet 2 of 6)

Name	Set definition	Description
allincs	PRSUS WHERE class = i	Set of all increment PRSUs in the office
allprsus	no syntax description is applicable to this item	Set of all PRSUs in the office
app_not_inst	PRSUS WHERE status = a class = i installed =n instreqd = y	Set of all increment PRSUs that are applied but not installed
astat	PRSUS WHERE status = a ON alldests	Set of all A (applied) PRSUs in the office
autoapply	PRSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = va ON alldests PRSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = r ON alldests	Set of all PRSUs that will be auto-applied the next time AUTOAPPLY runs
autoinstall	PRSUS WHERE class = i instreq = y autoinst = y	Set of all increment PRSUs that are autoinstallable
cm_active	PRSUS WHERE category = act active = y ON cm	Set of all activatable CM PRSUs that are activated
cm_astat	PRSUS WHERE status = a ON cm	Set of all applied CM PRSUs
cm_autoapp	PRSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = va ON CM PRSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = r ON cm	Set of CM PRSUs that can be auto applied
cm_freeavail	PRSUS WHERE freemem = av ON cm	Set of CM PRSUs available for FREEMEM
cm_freed	PRSUS WHERE freemem = f ON cm	Set of CM PRSUs that have been FREED
cm_notactive	PRSUS WHERE category = act active = n ON cm	Set of CM activatable PRSUs that are not activated
cm_nvstat	PRSUS WHERE status = nv ON cm	Set of CM PRSUs with NV status
cm_rstat	PRSUS WHERE status = r ON cm	Set of CM PRSUs with a removed status
cm_vastat	PRSUS WHERE status = va ON cm	Set of CM PRSUs with a validated status

Table 9-1 Predefined PRSU set definitions (Sheet 3 of 6)

Name	Set definition	Description
coldreq	PRSUS WHERE restart = cold	Set of all PRSUs that need cold restarts to become fully functional
dbg_app	PRSUS WHERE category = dbg status = a ON sos	Set of all CM and ISN PRSUs with category DBG that are applied
dbg_not_app	PRSUS WHERE status = va ON sos PRSUS WHERE status = r ON sos PRSUS WHERE status = nv ON sos & PRSUS WHERE category = dbg	Set of all CM and ISN PRSUs with category DBG that are not applied
dnr_not_app	PRSUS WHERE status = va ON sos PRSUS WHERE status = r ON sos PRSUS WHERE status = nv ON sos & PRSUS WHERE category = dnr	Set of all PRSUs with category DNR that are not applied
emg_not_app	PRSUS WHERE status = va ON sos PRSUS WHERE status = r ON sos PRSUS WHERE status = nv ON sos & PRSUS WHERE category = emg	Set of all PRSUs with category EMG that are not applied
entryinvalid	PRSUS WHERE entrvald = n	PRSUs that are applied but the actual PRSU file does not exist on the system. The PRSU should be obtained and validated to correct the state.
freed	PRSUS WHERE freemem = f ON sos	Set of all PRSUs that have been FREED
freeavail	PRSUS WHERE freemem = av ON sos	Set of all PRSUs available for FREEMEM
gen_not_app	PRSUS WHERE status = va ON sos PRSUS WHERE status = r ON sos PRSUS WHERE status = nv ON sos & PRSUS WHERE category = gen	Set of all PRSUs with category GEN that are not applied
incfreeavail	PRSUS WHERE class = i freemem = av ON cm	CM increments available to have their memory freed
incinst	PRSUS WHERE status = a class = i installed = y	Set of all increment PRSUs that are installed

9-4 Predefined set definitions

Table 9-1 Predefined PRSU set definitions (Sheet 4 of 6)

Name	Set definition	Description
incnotinst	PRSUS WHERE status = a class = i installed = n instreqd = y	Set of all increment PRSUs that require installation but are not installed
instnotreq	PRSUS WHERE class = i instreqd = n	Set of all increment PRSUs that do not require installation
instreq	PRSUS WHERE class = i instreqd = y	Set of all increment PRSUs that require installation
isnfreeavail	PRSUS WHERE freemem = av ON isn	Set of all ISN PRSUs available for FREEMEM
isnnotactive	PRSUS WHERE category = act active = n ON isn	Set of all ISN activatable PRSUs that are not activated
isn_active	PRSUS WHERE category = act active = y ON isn	Set of all ISN activatable PRSUs that are activated
isn_astat	PRSUS WHERE status = a ON isn	Set of all PRSUs applied on ISNs
isn_autoapp	PRSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = va ON isn PRSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = r ON isn	Set of all PRSUs to be auto applied on ISNs
isn_freed	PRSUS WHERE freemem = f ON isn	Set of all PRSUs freed on ISNs
isn_nvstat	PRSUS WHERE status = nv ON isn	Set of all PRSUs with an NV status on ISNs
isn_rstat	PRSUS WHERE status = r ON isn	Set of all PRSUs with an R status on ISNs
isn_vastat	PRSUS WHERE status = va ON isn	Set of all PRSUs with a VA status on ISNs
ltd_not_app	PRSUS WHERE status = va ON sos PRSUS WHERE status = r ON sos PRSUS WHERE status = nv ON sos & PRSUS WHERE category = ltd	Set of all PRSUs with category LTD that are not applied
man_not_app	PRSUS WHERE status = va ON xpm PRSUS WHERE status = r ON xpm PRSUS WHERE status = nv ON xpm & PRSUS WHERE category = man	Set of all PRSUs with category MAN that are not applied

Table 9-1 Predefined PRSU set definitions (Sheet 5 of 6)

Name	Set definition	Description
notactivated	PRSUS WHERE category = act active = n ON sos	Set of all PRSUs with category ACT that have not been activated in the office
nvstat	PRSUS WHERE status = nv ON alldests	Set of all NV (needs validation) PRSUs in the office
obe_not_rem	PRSUS WHERE category = obe status = a ON sos	Set of all PRSUs with a category OBE that are not removed
obsoleted	PRSUS WHERE category = obs PRSUS WHERE category = obe	Set of all obsoleted PRSUs in the office
obs_not_rem	PRSUS WHERE category = obs status = a ON sos	Set of all PRSUs with a category OBS that are not removed
ppxl_astat	PRSUS WHERE status = a ON ppxl	Set of XPM PRSUs applied on all PPXLs
prsus_onhold	PRSUS WHERE onhold = y	Set of all PRSUs that are on hold
reloadreq	PRSUS WHERE restart = reload	Set of all PRSUs that need reload restarts to become fully functional
rem_prsus	PRSUS WHERE category < > dbg & PRSUS WHERE category < > obs & PRSUS WHERE category < > obe & PRSUS WHERE status = r	Set of all normal removed PRSUs
restartreq	PRSUS WHERE restart = warm PRSUS WHERE restart = cold PRSUS WHERE restart = reload	Set of all PRSUs that need any restart to become fully functional
rstat	PRSUS WHERE status = r ON alldests	Set of all R (removed) PRSUs in the office
siteprsus	PRSUS WHERE status = a ON site	Set of all applied SITE PRSUs
spmloadastat	PRSUS where status = a ON spmload	Set of applied SPM PRSUs applied on SPM load
spm_astat	PRSUS WHERE status = a ON spm	Set of SPM PRSUs applied on SPMs

Table 9-1 Predefined PRSU set definitions (Sheet 6 of 6)

Name	Set definition	Description
spm_autoapp	PRUSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = va ON spm PRUSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = r ON spm	Set of all SPM PRSUs to be auto applied
spm_nvstat	PRUSUS WHERE status = nv ON spm	Set of all SPM PRSUs that need to be validated
spm_rstat	PRUSUS WHERE status = r ON spm	Set of all removed SPM PRSUs
spm_vstat	PRUSUS WHERE status = va ON spm	Set of all validated SPM PRSUs
src_not_app	PRUSUS WHERE status = va ON xpm PRUSUS WHERE status = r ON xpm PRUSUS WHERE status = nv ON xpm & PRUSUS WHERE category = src	Set of all PRSUs with category SRC that are not applied on XPMs
unitmismatch	PRUSUS WHERE matemismatch = y ON alldests	Set of PRSUs not applied to both sides of a DEST and its mate unit.
vastat	PRUSUS WHERE status = va ON alldests	Set of all VA (validated) PRSUs in the office
warmreq	PRUSUS WHERE restart = warm	Set of all PRSUs that need warm restarts to become fully functional
xpm_astat	PRUSUS WHERE status = a ON xpm	Set of all applied XPM PRSUs
xpm_autoapp	PRUSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = va ON xpm PRUSUS WHERE autoapp = y custapp = y onhold = n pstate = prsu_ok status = r ON xpm	Set of all XPM PRSUs to be auto applied
xpm_nvstat	PRUSUS WHERE status = nv ON xpm	Set of all XPM PRSUs needing validation
xpm_rstat	PRUSUS WHERE status = r ON xpm	Set of all removed XPM PRSUs
xpm_vastat	PRUSUS WHERE status = va ON xpm	Set of XPM PRSUs with a VA status

DEST set definitions

The following table lists the DEST set definitions predefined by PRSM.

Table 9-2 Predefined DEST set definitions

Name	Purpose	Purpose
allisn	DESTS WHERE destname = isn	Set of all ISN DESTs
allppxl	DESTS WHERE destid = ppxl	Set of all PPXL DESTs
allsos	DESTS WHERE destname = sos	Set of all ISN and CM DESTs
allspm	DESTS WHERE destname = spm	Set of all spm DESTs
allspload	DESTS WHERE destname = spmload	Set of all spmload DESTS
allxpm	DESTS WHERE destname = xpm	Set of all XPM DESTs
dests_onhold	DESTS WHERE hold = y	Set of all DESTs on hold

10 Logs

PRSM provides the user with logs to facilitate maintaining a DMS office software load with software changes.

The following is a list of the PRSM logs:

- PRSM300—General trouble
- PRSM301—PRSM file failure
- PRSM302—File management fail
- PRSM303—PRSM PRSU discrepancy
- PRSM360—File audit failure
- PRSM380—Automated process failure
- PRSM381—Switch sanity failure
- PRSM400—PRSM action summary
- PRSM470—Status audit summary
- PRSM600—General information
- PRSM603—PRSMOPTS option change
- PRSM680—Automated process information
- PRSM681—AUTOPROC CI command
- PRSM682—AUTOOPTS option change
- PRSM683—AUTOPRSU option change

PRSM logs are grouped into logical sets based on log number:

- 300 to 399—Trouble logs
- 400 to 499—Service summary logs
- 600 to 699—Information logs

Trouble logs

Following are descriptions of the PRSM general trouble logs.

PRSM300—General trouble

The PRSM300 log reports general problems that could be service affecting. The log contains lines of trouble information text. Following are examples of PRSM300 log reports.

```
PRSM300 APR18 13:46:58 0800 TBL General Trouble
User Class: CI_USER    User Identity: TEAM20
ERROR: Could not create disconnected user to
execute CI commands for PRSUs.
PRSM user is OPERATOR.
```

```
PRSM300 APR18 13:55:18 0900 TBL General Trouble
User Class: CI_USER    User Identity: TEAM20
ERROR: Could not logout disconnected user that
executed CI commands for PRSUs.
PRSM user is OPERATOR.
```

PRSM301—PRSM file failure

A PRSM301 log is generated for failures associated with locating or reading PRSU files. PRSM301 contains lines of information identifying the exact nature of the failure. Following is an example of a PRSM301 log report.

```
PRSM301 FEB24 08:48:22 1287 TBL PRSM File Failure
601: Opcode cmd in wrong order in PRSU file.
Retrieve a new copy of the PRSU file.
PRSU:JVA12BB0
File:JVA12BB0$PATCH
Device:SFDEV
```

Refer to the chapter 11, "PRSM301 log reason interpretation" on page -1 in this document for a list of reasons, explanations, and actions for PRSM301.

PRSM302—File management failure

A PRSM302 log is generated when there is a problem with file management. These problems could occur if there is a problem such as a failure to open or read a file. Following are examples of PRSM302 log reports.

```
PRSM302 MAY01 10:27:23 7700 TBL  PRSM File Control Problem
Command: prsu file search
Action: scan all files
Device:SFDEV
Device error
*** There may be other logs, please check. ***
```

```
PRSM302 MAY02 15:42:39 3400 TBL  PRSM File Control Problem
Command: VALIDATE
Action: get file information
PRSU: GNA07CB9
Illegal file system operation requested
*** There may be other logs, please check. ***
```

```
PRSM302 JAN08 00:01:34 9400 TBL  PRSM File Control Problem
Command: prsu file compare
Action: close a file
Filename:JED31BB9$PATCH
File not open
*** There may be other logs, please check. ***
```

Note: Filename, device name, or PRSU do not appear in a PRSM302 log if that information is not available.

PRSM303—PRSM PRSU discrepancy

A PRSM303 log is generated when a patching discrepancy is found between the PRSM database and a peripheral. The discrepancy is discovered by executing either the PRSM DBAUDIT or the AUTOPROC STATAUD START command.

```
PRSM303 JUN09 23:25:49 6400 TBL  PRSM PRSU DISCREPANCY User
Class: STATUS_AUDIT_USER      User Identity: STATUS_AUDIT
PRSU discrepancy found between the PRSM database and the
destination.
The PRSM database has been updated to match the destination
database.
DEST:  SPM  0  CEM  0
```

```
-----
PRSUs in PRSM database but not in destination:
P00BLS10, P01BLS10, P02BLS11
```

```
-----
PRSUs in destination but not in PRSM database:
P03BLS10, P04BLS10, P05BLS11
```

Note 1: This log is not generated on a post-ONP DBAUDIT.

Note 2: The "..." in the PRSM303 log is used when more than 25 PRSUs meet the alarmable condition. In this case, only the first 25 prsuids are displayed.

PRSM360—File audit failure

A PRSM360 log reports results from the PRSM file audit. The log is generated if the audit finds a missing SPM or XPM PRSU file or an unequal copy of a \$PATCH file for a PRSU. The log contains up to 10 PRSUIDs and indicates the last device where the PRSU was found or, in the case where unequal \$PATCH files were found, all devices associated with the unequal \$PATCH files are listed. Following is an example of the PRSM360 log report.

```
PRSM360 FEB12 23:09:17 2000 TBL File Audit Failure
Files Not Found Last device
(no missing XPM PRSU files)
Unequal Files Devices
-----
BAL00BB8$PATCH SFDEV S00DSCRATCH
```

PRSM380—Automated process failure

The PRSM380 log is generated when a problem is encountered during execution of the automated processes. Examples of when the log is generated include not being able to drop synchronization on the CM because the switch is already out of sync and not being able to establish a connection with the PRSM Request Router to execute PRSM maintenance commands. Following are examples of PRSM380 log reports.

```
PRSM380 OCT1 01:03:17 8900 TRBL AUTOMATED PROCESS FAILURE
Automated Process: AUTOAPP
Process aborted: Switch is out-of-sync.
```

```
PRSM380 OCT1 01:03:17 8900 TRBL AUTOMATED PROCESS FAILURE
Automated Process: AUTOAPP
AUTOAPPLY aborted: Autodump is in progress.
```

PRSM381—Switch sanity failure

The PRSM381 log is generated when switch sanity fails based on the logs defined in table AUTOMON. The log can be generated during the PREMON log monitoring period before the Autoapply and Autoinstall processes are executed or during the POSTMON log monitoring period after the Autoapply and Autoinstall processes are executed. Following is an example of a PRSM381 log report.

```
PRSM381 OCT1 01:03:17 8900 TRBL SWITCH SANITY FAILURE
Switch Sanity limit exceeded before applying/installing
PRSUs.
```

Log	Allowed	Exceeded	Log	Allowed	Exceeded
CM115:	1	4	CM119:	24	40
SWER:	5	6			

Service summary logs

Following are descriptions of the PRSM service summary logs.

PRSM400—PRSM action summary

Each execution of an applicable PRSM command produces one or more logs. The total number of logs created is determined by the total number of actions initiated by the command.

Each instance of the log contains information for a maximum of 15 actions. For example, consider an APPLY command with a PRSU set containing 3 PRSUs (one host, one ISN, and one XPM) and a DEST set specifying 59 destinations (1 host, 26 ISNs, and 32 XPMs).

- The host PRSU is applicable to the host destination.
- The ISN PRSU is applicable to 22 of the 26 ISN destinations.
- The XPM PRSU is applicable to 30 of the 32 XPM destinations.

This provides a total of 53 separate application actions and requires the generation of 4 logs.

When this log contains less than fifteen actions, the log is generated when the user quits PRSM.

The following commands are included in the PRSM400 log: VALIDATE, APPLY, REMOVE, FREEMEM, DBAUDIT, and ASSIGN for the activation of act patches; and ASSIGN for the installation of increment PRSUs. Following is an example of a PRSM400 log report.

```

TL_ALL06AL      PRSM400 DEC11 10:30:58 8200 INFO COMMAND SUMMARY
  User Class: CI_USER                      User Identity: OPERATOR
                                     Destination Command Pass/ Time
  PRSU id                               (DEST)      Method  Fail  Complete
  -----                               -
CAL22CB8                               NONE       APPLY   Fail  09:52:32
CAL22CB8                               NONE       APPLY   Fail  09:53:12
BAS07AL                               SITE       APPLY   Pass  10:30:16
CAL22CB7                               CM         VALIDATE Pass  10:30:26
CAL22CB7                               CM         APPLY   Pass  10:30:39
CAL22CB7                               CM         REMOVE  Pass  10:30:47
    
```

PRSM401—Assignment summary

A PRSM401 log reports a summary of the PRSUs that have been activated or deactivated with the ASSIGN command. The command ASSIGN ACTIVE Y activates a PRSU. The command ASSIGN ACTIVE N deactivates a PRSU.

The switch generates the log after the log contains 15 PRSUs or the user quits PRSM. Following is an example of the PRSM401 log report.

```

BASE_ALL08AQ PRSM401 JAN10 10:34:33 1900 INFO ASSIGN SUMMARY
  User Class: CI_USER      UserIdentity: VMAP1
    Destination
PRSU id      (DEST)      Assign      Action      Pass/ Time
-----      -
PLR01BB9    CM          ACTIVE      Y          Pass 10:24:59
PRL01BB9    CM          ACTIVE      N          Pass 10:25:02
    
```

PRSM470—Status audit summary

A PRSM470 log reports results from the PRSM status audit and is generated every night after the Status Audit completes. It contains a summary of how many PRSUs were found in each condition and a breakdown of which PRSU is in each alarmable condition. Following is an example of the PRSM470 log report.

Note 1: The "*" in the PRSM470 log is a field separator.

Note 2: The "... " in the PRSM470 log is used when more than ten PRSUs meet the alarmable condition. In this case, only the first ten prsuids are displayed.

Note 3: If the ENABLED field for an alarm condition in table PRSMALRM is set to N, the alarm condition will not be listed in PRSM470.

Note 4: A copy of the indicated PRSU must be obtained and validated in the switch to correct the database.

```

PRSM470 MAR10 01:02:41 5600 INFO Status Audit Summary
  PRSU Condition      Number of PRSUs
  -----
  ACT_NOT_APP          0
  NOTACTIVATED         0
  ACTPW_NOTACT         0
  DBG_APP              1
  DBG_NOT_APP          0
  DNR_NOT_APP          0
  EMG_NOT_APP          0
  GEN_NOT_APP          2
  LTD_NOT_APP          0
  MAN_NOT_APP          0
  SRC_NOT_APP          1
  OBS_NOT_REM          0
  OBE_NOT_REM          0
  REM_PRSUS            0
  INCNOTINST           0
  UNITMISMATCH         0
  ENTRYINVALID         1
  PRSU Condition Details
  -----
  ACT_NOT_APP
  *
  NOTACTIVATED
    
```

```

*
ACTPW_NOTACT
*
DBG_APP
DKC52CB5
*
DBG_NOT_APP
*
DNR_NOT_APP
*
EMG_NOT_APP
*
GEN_NOT_APP
MBR30CB5
LNC15IB5
*
LTD_NOT_APP
*
MAN_NOT_APP
*
SRC_NOT_APP
XMF06X05
*
OBS_NOT_REM
*
OBE_NOT_REM
*
REM_PRSUS
*
INCNOTINST
*
UNITMISMATCH
*
ENTRYINVALID

```

Information logs

Following are descriptions of the PRSM information logs.

PRSM600—General information

A PRSM600 log is generated for general information that may be out of the ordinary. PRSM600 is a log with up to five lines of information. Following is an example of the PRSM600 log reports.

```

PRSM600 APR27 11:29:56 1700 INFO General Info
PRSM CM Statuses have been synced by operator.

```

PRSM603—PRSMOPTS option change

A PRSM603 log is generated whenever an option is changed in table PRSMOPTS. Following is an example of the PRSM603 log report.

```

PRSM603 MAY16 15:19:47 8300 INFO PRSMOPTS Option Change
DESTMETH changed from BROADCAST to NO_BROADCAST.

```

PRSM680—Automated process information

The PRSM680 log displays general automated process information. For example, the PRSM680 displays the start time for a scheduled Autoapply process. Following are examples of PRSM680 log reports.

```
PRSM680 OCT1 01:03:17 8900 INFO AUTOMATED PROCESS INFO
Automated Process: AUTOAPP
Process started.
PRSM680 OCT1 01:03:17 8900 INFO AUTOMATED PROCESS INFO
Automated Process: AUTOAPP
Process has been manually delayed.
```

PRSM681—AUTOPROCESS CI command

The PRSM681 log is generated when the AUTOPROC command is executed for the start, stop, or delay options. The PRSM681 log displays the automated process, the command parameter, and the user that entered the command. Following are examples of PRSM681 log reports.

```
PRSM681 OCT1 01:03:17 8900 INFO AUTOPROCESS CI COMMAND
AUTOPROC AUTOAPP STOP issued by ADMIN.
PRSM681 OCT1 01:03:18 8900 INFO AUTOPROCESS CI COMMAND
AUTOPROC FILEAUD START issued by ADMIN.
```

PRSM682—AUTOOPTS option change

The PRSM682 log is generated when any of the datafill options are changed in the AUTOOPTS table. Following are examples of PRSM682 log reports.

```
PRSM682 OCT1 12:15:27 6900 INFO AUTOOPTS OPTION CHANGE
MAXATMPT changed from $ to 4 by ADMIN.
PRSM682 OCT1 12:15:27 6900 INFO AUTOOPTS OPTION CHANGE
APPROVAL changed from N to Y by ADMIN.
```

PRSM683—AUTOPRSU option change

The PRSM683 log is generated when any of the datafillable options are changed in the AUTOPRSU table. Following is an example of the PRSM683 log report.

```
PRSM683 OCT1 12:15:28 8900 INFO AUTOPRSU OPTION CHANGE
Automated Process: AUTOAPP
MTWTFSS changed from NNNNNNNN to YYYYYYYY by ADMIN
```

11 PRSM301 log reason interpretation

PRSU errors

When a corrupt PRSU file is found, a PRSM301 log is generated. The following table shows the integer that appears in the log, the response reflected on the MAP screen when an error occurs, an explanation of the integer value, and suggested corrective actions.

Within this table references are made to different types of PRSM records. To better understand the table information, these records are defined as follows.

- J-record is used for storing the PRSUID of the increment PRSU and applies to ICCs only.
- M-record indicates which modules must be present on the switch and applies to CM and ISN nodes only.

V-record identifies the version issue of the PRSU file and is used with ICCs only.

Table 11-1 PRSM301 log reason interpretation (Sheet 1 of 10)

Reason	Response	Explanation	Action
1	<code>prsm_no_method_for_cleanup</code>	Attempt to clean up storage after parsing a PRSU failed.	Contact your next level of support.
2	<code>prsm_no_method_for_parsing</code>	Attempt to parse a PRSU file failed.	Contact your next level of support.
3	<code>prsm_mem_alloc_failed</code>	Could not allocate storage while attempting to parse a PRSU file.	Contact your next level of support.
4	<code>prsm_oop_error</code>	Attempt to call an OOP method while attempting to parse a PRSU file failed.	Contact your next level of support.

Table 11-1 PRSM301 log reason interpretation (Sheet 2 of 10)

Reason	Response	Explanation	Action
6	grp_within_bundle	A group PRSU was found within a PRSU bundle while attempting to parse a PRSU file.	Retrieve a new copy of the PRSU file.
450	lfs_conv_cant_alloc_store_for_pkg_prsus	Could not allocate storage while parsing a PRSU package.	Contact your next level of support.
451	lfs_conv_cant_alloc_store_for_grp_needed_prsus	Could not allocate storage while parsing a group PRSU, specifically needed for the PRSUs.	Contact your next level of support.
452	lfs_conv_unable_to_create_prsuid	While attempting to parse a PRSU file, could not create an entry in the PRSU map. This occurs for the PRSU being parsed, its patch (P) records, and for needed PRSUs for group PRSUs.	Contact your next level of support.
453	lfs_conv_prsu_does_not_exist	Attempted to retrieve a nonexistent PRSU.	Retrieve a new copy of the PRSU file.
487	lfs_invalid_install_reqd	The install required boolean in the A-record for the PRSU file did not contain a valid boolean value.	Retrieve a new copy of the PRSU file.
488	lfs_cant_conv_incrdelt_ds_bytes	There was a problem parsing the DS store size from an incrdelt record in the PRSU file.	Retrieve a new copy of the PRSU file.
489	lfs_cant_conv_incrdelt_ps_bytes	There was a problem parsing the PS store size from an incrdelt record in the PRSU file.	Retrieve a new copy of the PRSU file.
490	lfs_cant_conv_incrload_ps_bytes	There was a problem parsing the PS store size from an incrlload record in the PRSU file.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 3 of 10)

Reason	Response	Explanation	Action
491	lfs_cant_conv_ incrload_ds_bytes	There was a problem parsing the DS store size from an incrload record in the PRSU file.	Retrieve a new copy of the PRSU file.
492	lfs_duplicate_ loader_options	There were duplicate loader options found in an incrload record in the PRSU file.	Contact your next level of support.
493	lfs_to_many_ loader_options	There were more than the allowable number (currently 3) of loader options found in an incrload record in the PRSU file.	Contact your next level of support.
494	lfs_invalid_ jrecord_in_file	A J-record was found in a PRSU file that should not contain a J-record.	Contact your next level of support.
495	lfs_invalid_ vrecord_in_file	A V-record was found in a PRSU file that should not contain a V-record.	Contact your next level of support.
496	lfs_cant_add_ label_to_dir	While attempting to add a new module label to the label directory, an error occurred.	Contact your next level of support.
497	lfs_incrdelt_no_ assoc_mrecord	An incrdelt record that did not have a corresponding M-record was found in the PRSU file.	Contact your next level of support.
498	lfs_invalid_ loader_option	An unsupported loader option was found in an incrload record in the PRSU file.	Contact your next level of support.
499	lfs_invalid_safe_ to_install	The safe to install boolean in the A-record for the PRSU file did not contain a valid boolean value.	Retrieve a new copy of the PRSU file.
500	lfs_invalid_safe_ to_apply	Retrieved an invalid character from safe_to_apply field.	Retrieve a new copy of the PRSU file.
501	lfs_invalid_ special_app	Retrieved an invalid character from the application field of the PRSU file.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 4 of 10)

Reason	Response	Explanation	Action
502	lfs_invalid_authorized	Retrieved an invalid character from the authorized field of the PRSU file.	Retrieve a new copy of the PRSU file.
503	lfs_invalid_cicont	Expected to find a CI continuation character while parsing a PRSU file. Retrieved an invalid character.	Retrieve a new copy of the PRSU file.
506	lfs_invalid_command_rec	Invalid command in the command section of the PRSU file.	Retrieve a new copy of the PRSU file.
507	lfs_invalid_section_head	Invalid section header in PRSU file.	Retrieve a new copy of the PRSU file.
508	lfs_garbage_at_end_of_ci	Characters at the end of a CIEND command in PRSU command section.	Retrieve a new copy of the PRSU file.
509	lfs_ciend_with_no_ci_cmd	CIEND command follows a non-CI command action in the command section of the PRSU file.	Retrieve a new copy of the PRSU file.
510	lfs_actions_nil_with_ciend	A CIEND command has no corresponding command in the PRSUfile command section.	Retrieve a new copy of the PRSU file.
511	lfs_invalid_adm_delimiter	Invalid record in administration section of PRSU file.	Retrieve a new copy of the PRSU file.
512	failed_to_alloc_store_for_text_records	Could not allocate storage for the text portion of the special applications instructions of the file.	Contact your next level of support.
513	lfs_cant_convert_act	Could not convert activation index from character form to integer form from the PRSU file.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 5 of 10)

Reason	Response	Explanation	Action
515	lsf_cant_convert_bcsnum	Could not convert the BCS number from character form to integer form from the PRSU file.	Retrieve a new copy of the PRSU file.
516	lfs_invalid_feature_or_change	Invalid character in classification record in the administration section of PRSU file.	Retrieve a new copy of the PRSU file.
517	lfs_invalid_restart_field	Invalid restart type in restart field of the administration section of the PRSU file.	Retrieve a new copy of the PRSU file.
518	lfs_invalid_processor_field	Invalid processor type in the processor field of the administration section of the PRSU file.	Retrieve a new copy of the PRSU file.
519	lfs_invalid_category_field	Invalid characters in category field of the administration section of the PRSU file.	Retrieve a new copy of the PRSU file.
520	lfs_unable_to_determine_class	Based on the file extension, sections in the file, and file type, a file class could not be determined.	Contact your next level of support.
522	lfs_not_able_to_write_code	The separate code files contained within a PRSU file could not be written to SFDEV.	Check table DSLIMIT and verify there is enough SFDEV storage available. Contact your next level of support.
523	lfs_unable_to_open_code_file	Unable to open working code file on SFDEV while parsing the PRSU file.	Contact your next level of support.
524	lfs_unable_to_close_code_file	Unable to close working code file on SFDEV while parsing the PRSU file.	Contact your next level of support.
525	lfs_never_found_eof_marker	While attempting to parse a file, PRSM did not find an end of file marker.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 6 of 10)

Reason	Response	Explanation	Action
527	lfs_file_lrecl_ too_long	PRSM attempted to open a file that appeared to be valid and found the record length was too long.	Retrieve a new copy of the file. Verify all file parameters are properly set before transferring the file to the switch.
528	lfs_checksums_ do_not_match	The calculated checksum does not match the checksum stored in the parsed PRSU file.	Retrieve a new copy of the PRSU file.
529	lfs_missing_end_ ADM_label	PRSM did not find an administration section end label while parsing a PRSU file.	Retrieve a new copy of the PRSU file.
530	lfs_missing_end_ CMD_label	PRSM did not find a command section end label while parsing a PRSU file.	Retrieve a new copy of the PRSU file.
531	lfs_missing_end_ CODE_label	PRSM did not find a code section end label while parsing a PRSU file.	Retrieve a new copy of the PRSU file.
532	lfs_missing_end_ ISU_label	PRSM did not find an ISU section end label while parsing a PRSU file.	Retrieve a new copy of the PRSU file.
533	missing_required_ section_in_patch	The parsed PRSM file does not contain all the required sections.	Retrieve a new copy of the PRSU file.
534	invalid_option_ section_in_patch	An invalid optional section was found in the parsed PRSU file.	Retrieve a new copy of the PRSU file.
535	lfs_cirem_ without_ciapp	A CIREM statement without a CIAPP statement was found in the PRSU file command section.	Retrieve a new copy of the PRSU file.
537	lfs_no_section_ specified_to_ parse	PSRM was unable to parse a PRSU file because it was not given a section to verify.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 7 of 10)

Reason	Response	Explanation	Action
538	lsf_spatch_no_ assoc_mod_record	An SPATCH record was found in the command section of the PRSU file with no corresponding module record.	Retrieve a new copy of the PRSU file.
539	lfs_cant_convert_ version	The V-record in the prsu file is corrupt.	Retrieve a new copy of the PRSU file.
540	lfs_grp_section_ empty	An empty group section was found while parsing a PRSU file.	Retrieve a new copy of the PRSU file.
541	lfs_grp_record_ id_too_long	A group was found that had a record id that was too long (greater than 1 character).	Retrieve a new copy of the PRSU file.
542	lfs_grp_record_ has_too_many_ tokens	A group record contains extraneous characters in the parsed file.	Retrieve a new copy of the PRSU file.
543	lfs_grp_record_ not_0_record	The first record of the group section is not the required 0 record in the parsed file.	Retrieve a new copy of the PRSU file.
544	lfs_grp_records_ after_rep_patch	Records were found after the replacement record in the group section of the parsed file.	Retrieve a new copy of the PRSU file.
545	lfs_grp_ duplicate_0_ records	A second 0 record was found in the group section of the parsed file.	Retrieve a new copy of the PRSU file.
546	lfs_grp_invalid_ record_after_0_ record	An invalid record was found after the 0 record in the group section of the parsed file.	Retrieve a new copy of the PRSU file.
547	lfs_grp_failed_ to_alloc_X_ record_store	PRSM could not allocate data store for an X record in the group section of the parsed file.	Retrieve a new copy of the PRSU file.
548	lfs_grp_more_ than_one_grp_ section	PRSM found a second group section in the parsed file.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 8 of 10)

Reason	Response	Explanation	Action
549	lfs_grp_patch_portion_of_record_bad	The patch portion of a group record in the group section of the parsed file was invalid or empty.	Retrieve a new copy of the PRSU file.
550	lfs_missing_end_GRP_label	Could not find an end label for the group section of a parsed file.	Retrieve a new copy of the PRSU file.
551	lfs_grp_section_out_of_order	A group section in the parsed file was not in its correct location.	Retrieve a new copy of the PRSU file.
560	lfs_pkg_section_empty	PRSM has found a package in the PRSU file with no PRSUs listed in the package section.	Retrieve a new copy of the PRSU file.
561	lfs_missing_end_PKG_label	PRSM has found a package in the PRSU file with no end of package label found.	Retrieve a new copy of the PRSU file.
562	lfs_pkg_section_in_wrong_location	More than one package section was found in the parsed file.	Retrieve a new copy of the PRSU file.
563	lfs_pkg_patch_name_nil	A record in the package section of the parsed file does not contain a valid PRSU id.	Retrieve a new copy of the PRSU file.
564	lfs_pkg_invalid_chars_after_patch_name	Extra characters were found at the end of a record in the package section of the parsed file.	Retrieve a new copy of the PRSU file.
565	lfs_pkg_failed_to_alloc_pkg_store	Temporary storage could not be allocated while parsing the package section of the file.	Contact your next level of support.
566	lfs_pkg_patch_name_too_long	A PRSU id in a record in the package section of the parsed file is too long.	Retrieve a new copy of the PRSU file.
567	lfs_pkg_patchid_not_in_package_file	No corresponding PRSU file found for a PRSU id listed in the package section of the file.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 9 of 10)

Reason	Response	Explanation	Action
568	lfs_pkg_system_contradiction	A package section in the parsed file contains corrupted information.	Retrieve a new copy of the PRSU file.
569	lfs_pkg_file_specified_is_not_a_package	PRSM thinks it should be parsing a file, but the given file is not a PRSU package.	Retrieve a new copy of the PRSU file.
570	lfs_pkg_eof_before_begin_package	The end of the parsed PRSU file was reached before the package section of the file was found.	Retrieve a new copy of the PRSU file.
575	lfs_adm_more_than_one_adm_section	The parsed file contains more than one administration section.	Retrieve a new copy of the PRSU file.
576	lfs_adm_section_out_of_order	The administration section of the parsed file was found to be out of order with respect to the other sections.	Retrieve a new copy of the PRSU file.
577	lfs_cmd_more_than_one_command_section	The parsed file contains more than one command section.	Retrieve a new copy of the PRSU file.
578	lfs_cmd_section_out_of_order	The command section of the parsed file was found to be out of order with respect to the other sections.	Retrieve a new copy of the PRSU file.
579	lfs_code_section_out_of_order	The code section in the parsed PRSU file was not found in the proper order with respect to the other sections.	Retrieve a new copy of the PRSU file.
580	lfs_chk_more_than_one_checksum_section	The parsed PRSU file contains more than one checksum section.	Retrieve a new copy of the PRSU file.
581	lfs_chk_section_out_of_order	The checksum section in the parsed PRSU file was not found in the proper order with respect to the other sections.	Retrieve a new copy of the PRSU file.
582	lfs_eof_more_than_one_end_of_file_section	The parsed PRSU file contains more than one end of file section.	Retrieve a new copy of the PRSU file.

Table 11-1 PRSM301 log reason interpretation (Sheet 10 of 10)

Reason	Response	Explanation	Action
583	lfs_eof_section_ out_of_order	For the given parsed PRSU file we expected to find the end of file, but found an unexpected section.	Retrieve a new copy of the PRSU file.
584	lfs_error_ opening_file	The PRSU being parsed failed to open. The device containing the file may be out of service.	Check for other logs and check the device.
600	opcode_unknown_ opcode_cmd	An unknown opcode command was found in the command section of the parsed PRSU file.	Retrieve a new copy of the PRSU file.
601	opcode_cmds_in_ wrong_order	An opcode command was found in the wrong order in the command section of the parsed PRSU file.	Retrieve a new copy of the PRSU file.
602	opcode_no_end_ statement	PRSM did not find an END statement for an opcode patch in the command section of the parsed PRSU file.	Retrieve a new copy of the PRSU file.
N/A	Not applicable	PRSU file not found for PRSU during DBAUDIT.	Execute the VALIDATE command with a valid PRSU file to make the PRSM database accurate.

12 Interface to PRSM

Introduction

Many systems interface with PRSM. This chapter describes what PRSM provides to the ISN autoimage system.

ISN Autoimage support

PRSM applies, removes, and audits PRSUs on a DMS node without manual intervention. When a node fails, the recovered node's software does not contain PRSUs if an image has not been taken.

ISN Autoimaging periodically checks DMS nodes for PRSU activities. In the event that any of these activities have occurred, ISN Autoimaging begins the process necessary to perform an image of affected nodes.

ISN autoimage provides an interface between system software and PRSM. The interface enables the autoimaging infrastructure to request and receive a list of nodes that have had patching activity and require imaging.

PRSM acts as an image requestor and provides loadnames and a list of nodes for each loadname when queried by the autoimaging facility. Autoimaging is performed after PRSM updates are made or after PRSM updates are removed. Based on specific criteria, the autoimaging infrastructure creates a list of nodes eligible for imaging.

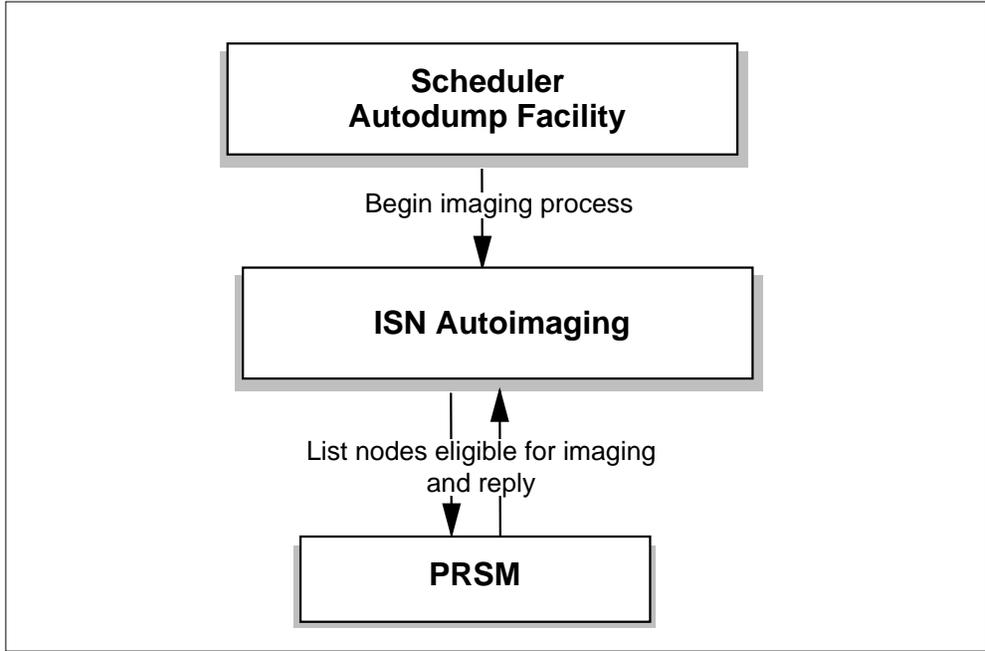
To determine which nodes will have images done, the software does a comparison and review of nodes with like nodenames. Each unique loadname undergoes a comparison.

Based on the following criteria, a node is, or is not, selected as a candidate for the imaging list.

- A DEST where HOLD = Y is not considered for imaging.
- A review of the APPLEVEL and ACTLEVEL fields is made to determine if changes have occurred since the last image was done.
- Nodes with the most patches applied, or the highest APPLEVEL become imaging candidates.

Nodes with the highest ACTLEVEL are considered. When nodes with like nodenames have the same APPLEVELE, ACTLEVEL field determines which nodes are eligible for imaging.

Figure 12-1 ISN autoimaging process



Note: For more information about the ISN Autoimaging feature refer to the *Feature Description Manual*, 297-8001-801.

DMS-100 Family
**Post-Release Software Manager
(PRSM)**
Reference Guide

Product Documentation-Dept 3423
Nortel Networks
P.O. Box 13010
RTP, NC 27709-3010
1-877-662-5669
Electronic mail: cits@nortelnetworks.com

Copyright © 1995, 1996, 1997, 1998, 1999, 2000 Nortel Networks,
All Rights Reserved

NORTEL NETWORKS CONFIDENTIAL: The information contained herein is the property of Nortel Networks and is strictly confidential. Except as expressly authorized in writing by Nortel Networks, the holder shall keep all information contained herein confidential, shall disclose the information only to its employees with a need to know, and shall protect the information, in whole or in part, from disclosure and dissemination to third parties with the same degree of care it uses to protect its own confidential information, but with no less than reasonable care. Except as expressly authorized in writing by Nortel Networks, the holder is granted no rights to use the information contained herein.

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design and components as progress in engineering and manufacturing may warrant.

DMS, MAP, NORTEL, NORTEL NETWORKS, the NORTEL NETWORKS LOGO, the GLOBEMARK, HOW THE WORLD SHARES IDEAS, NORTHERN TELECOM, NT, and SUPERNODE are trademarks of Northern Telecom.

Publication number: 297-8991-540
Product release: BASE14 and up
Document release: Standard 10.01
Date: February 2000
Published in U.S.A.