NORTEL NETWORKS

Preside Multiservice Data Manager

Server Reference

Guide

241-6001-310

Preside Multiservice Data Manager

Server Reference

Guide

Publication: 241-6001-310
Document status: Standard
Document version: 15.1RSUP
Document date: August 2004

Printed in Canada

Publication history

August 2004

15.1 RSUP Standard
Commercial availability except for MPE support which will be available in a future release.

Contents

Chapter 7
DPN Management Data Router (DMDR) 81

Chapter 8
DPN NCS Communications Manager (NCSMGR) 97

Chapter 38
Network Model Coordinator (DNMNMC) 303

Chapter 39
Network Model Editing server (EDSERVER) 307

Chapter 40
Network Model Server (NMSERVER) 311

Chapter 44 Passport 4400 Restore Provider (PRSTPP4400) 341

Chapter 45 Passport 4460 Backup Provider (PBCKPP4460) 347

Chapter 46 Passport 4460 Restore Provider (PRSTPP4460) 353

Chapter 47 Passport Backup Provider (PBCKPP) 359

Chapter 48
Passport Command Access Server (PPAccessServer) 363

Chapter 49
Passport Communications Manager (FDTM) 367

Chapter 50
Passport Configuration Model Server (PCMS) 381

Chapter 51
Passport Management Data Router (FMDR) 389

Chapter 52
Passport Nodal Provisioning Configuration Server (PCSERVER) 415

Chapter 53
Passport Restore Provider (PRSTPP) 419

Chapter 54
Fault Device Access Agent (PSVAGENT) 423

Chapter 55
Performance Measurement Stream Processor (PMSP) 435

Chapter 56
Real Time Alarm Collector (RTACCOL) 451

Chapter 57 Restore Controller (NSCTLRST) 457

Chapter 58 Security Audit Log Collector (SALC) 461

Chapter 59 SNMP Data Collection Daemon (GENDCD) 475

Chapter 67 Session Servers 545

Chapter 68 Servers controlled by HP OpenView 553

About this document

The following topics are discussed in this section:

- “Who should read this document and why” (page 31)
- “What you need to know” (page 31)
- “What’s new in this document” (page 35)
- “Text conventions” (page 36)
- “Related documents” (page 38)

Who should read this document and why

This document contains reference information for Preside Multiservice Data Manager servers.

This document is intended for system administrators who specialize in managing networks.

What you need to know

Users of this document require the following knowledge and skills:

- working knowledge of UNIX, the Solaris operating environment;
- Nortel Networks products and deployment;
- knowledge of Preside Multiservice Data Manager and its user interface.

How this document is organized

The 241-6001-310 *Preside MDM Server Reference Guide* contains the following sections:

- “Server Daemon (SVMDMN)” (page 39)

- “Multi-nodal Name Server (MNSD)” (page 47)
- “Backup Controller (NSCTLBCK)” (page 55)
- “Context Server (CTXSVR)” (page 65)
- “Customer Database Server (CDBSERVER)” (page 69)
- “DPN DBNL Auto-disabling Daemon (DBNLWatch)” (page 71)
- “DPN Management Data Router (DMDR)” (page 81)
- “DPN NCS Communications Manager (NCSMGR)” (page 97)
- “DPN PM File Access Server (PFAS)” (page 107)
- “Data Manager Agent (DMA)” (page 113)
- “Data Synchronization Server (DATASYNCSERVER)” (page 133)
- “Data Viewer Agent (PMAGENT)” (page 139)
- “Data Viewer Data Collection Daemon (PMDCD)” (page 143)
- “End-to-end Server (ETESERVER)” (page 147)
- “Fault Device Access Agent (PSVAGENT)” (page 423)
- “General Management Data Router (GMDR)” (page 151)
- “Host Group Directory Server (HGDS)” (page 177)
- “Injected Management Data Router (IMDR)” (page 189)
- “Log Collector (OAMC)” (page 199)
- “GMDR Agent (GMDRAGENT)” (page 207)
- “NM Agent (NMAGENT)” (page 209)
- “RTAC agent (RTCAGENT)” (page 211)
- “MDP DPN File Collector (MDPCOL)” (page 213)
- “MDP DPN File Manager (MDPDPNMGR)” (page 215)
- “MDP Disk Manager (MDPDISKMGR)” (page 219)
- “MDP File Mover Manager (MDPFMMGR)” (page 223)
- “MDP MPE 9500 Collector Manager (MPEMCMGR)” (page 235)
- “MDP MPE 9500 File Manager (MDPMPEMGR)” (page 241)

- “MDP Passport Data Model Manager (MDPDMM)” (page 231)
- “MDP Passport File Manager (MDPPPMGR)” (page 245)
- “MDP File Prober Manager (MDPFPMGR)” (page 227)
- “MDP Statistics Retrieval System (MDPSRS)” (page 249)
- “Multi-nodal Name Server Agent (MNSDAGENT)” (page 253)
- “Network Configuration Database Server (NCDSVR)” (page 257)
- “Network Data Access Mediator (NDAM)” (page 263)
- “MPE 9500 Communications Manager (NDTM)” (page 279)
- “MPE 9500 Management Data Router server (NMDR)” (page 283)
- “MPE Nodal Provisioning Configuration Server (NCSERVER)” (page 299)
- “Network Model Coordinator (DNMNMC)” (page 303)
- “Network Model Editing server (EDSERVER)” (page 307)
- “Network Model Server (NMSERVER)” (page 311)
- “Network Model Surveillance Updater (SURNUP)” (page 317)
- “Nodal Provisioning Configuration Manager (CONFIGMAN)” (page 331)
- “Passport 4400 Backup Provider (PBCKPP4400)” (page 335)
- “Passport 4400 Restore Provider (PRSTPP4400)” (page 341)
- “Passport 4460 Backup Provider (PBCKPP4460)” (page 347)
- “Passport 4460 Restore Provider (PRSTPP4460)” (page 353)
- “Passport Backup Provider (PBCKPP)” (page 359)
- “Passport Command Access Server (PPAccessServer)” (page 363)
- “Passport Communications Manager (FDTM)” (page 367)
- “Passport Configuration Model Server (PCMS)” (page 381)
- “Passport Management Data Router (FMDR)” (page 389)
- “Passport Nodal Provisioning Configuration Server (PCSERVER)” (page 415)

- “Passport Restore Provider (PRSTPP)” (page 419)
- “Performance Measurement Stream Processor (PMSP)” (page 435)
- “Real Time Alarm Collector (RTACCOL)” (page 451)
- “Restore Controller (NSCTLRST)” (page 457)
- “Security Audit Log Collector (SALC)” (page 461)
- “SNMP Data Collection Daemon (GENDCD)” (page 475)
- “SNMP IP Discovery server (IPDSVR)” (page 505)
- “SNMP Management Data Router (SMDR)” (page 509)
- “SNMP Proxy Agent (SPA) on MPE 9500” (page 521)
- “SNMP Trap Server Daemon (TSVR)” (page 529)
- “Secure FTP Daemon (launchSecureFTPD)” (page 533)
- “VPN Monitor Extractor (VPNMonitorExtractor)” (page 535)
- “VPN Monitor Server (VPNMonitorServer)” (page 541)
- “Session Servers” (page 545)
- “Servers controlled by HP OpenView” (page 553)
- “Server ports” (page 585)
- “Server names, executables, and suggested names” (page 591)

What’s new in this document

This document has been updated to include the following features:

- “MDM OAM and Security Audit Logging” (page 35)
- “MDM Database Synchronization Support” (page 35)
- “MDP MPE 9500 File Manager (MDPMPEMGR)” (page 35)
- “MDP MPE 9500 Collector Manager (MPEMCMGR)” (page 35)
- “MPE 9500 Management Data Router server (NMDR)” (page 36)
- “MPE 9500 Communications Manager (NDTM)” (page 36)
- “Nodal Provisioning Enhancements” (page 36)
- “SNMP Proxy Agent (SPA) on MPE 9500” (page 36)

MDM OAM and Security Audit Logging

This feature includes the following additions and enhancements:

- “Security Audit Log Collector (SALC)” (page 461)
- changes to the “Log Collector (OAMC)” (page 199).

MDM Database Synchronization Support

This feature includes the following additions and enhancements:

- “Backup Controller logging” (page 58)
- “Database Synchronization Controller Logging” (page 136)
- “Restore Controller logging” (page 459)

MDP MPE 9500 File Manager (MDPMPEMGR)

The Management Data Provider MPE 9500 File Manager (MDPMPEMGR) is a new server. It manages the conversion of raw data files to BDF format (when applicable) and the transfer of spooled files to down stream devices.

MDP MPE 9500 Collector Manager (MPEMCMGR)

The Management Data Provider Collector Manager (MPEMCMGR) is a new server. It manages the collection of spooled data from an MPE 9500.

MPE 9500 Management Data Router server (NMDR)

The NMDR server routes alarm event reports from Nortel Networks Multiservice Provider Edge 9500 (MPE 9500) devices to a GMDR server.

MPE 9500 Communications Manager (NDTM)

The NDTM server performs the following functions:

- creates and manages Nortel Networks Multiservice Provider Edge 9500 (MPE 9500) data translation (NDTR) processes. The NDTR processes allow the workstation to communicate with a MPE 9500 device.

MPE Configuration Management

This feature adds MPE support to configuration, network reporting, and backup and restore tools. The section "MPE Nodal Provisioning Configuration Server (NCSERVER)" (page 299) was added.

Nodal Provisioning Enhancements

This feature includes the following provisioning enhancements:

- the ability to retrieve a view from the backup site rather than download it from the Passport device, see "Startup command" (page 417).

SNMP Proxy Agent (SPA) on MPE 9500

The SNMP Proxy Agent (SPA) is available on the Nortel Networks Multiservice Provider Edge 9500 (MPE 9500) device. SPA provides a single point of SNMP access to several MPE 9500 devices through an MDM server.The following section was added: "SNMP Proxy Agent (SPA) on MPE 9500" (page 521).

Text conventions

This document uses the following text conventions:

- nonproportional spaced plain type

  Nonproportional spaced plain type represents system generated text or text that appears on your screen.

- nonproportional spaced bold type

  Nonproportional spaced bold type represents words that you should type or that you should select on the screen.

- italics

  Statements that appear in italics in a procedure explain the results of a particular step and appear immediately following the step.

  Words that appear in italics in text are for naming.

- [optional_parameter]

  Words in square brackets represent optional parameters. The command can be entered with or without the words in the square brackets.

- <general_term>

  Words in angle brackets represent variables which are to be replaced with specific values.

- UPPERCASE, lowercase

  Uppercase and lowercase letters that appear in UNIX commands and parameters must be matched exactly. The system matches upper and lowercase characters differently.

- |

  This symbol separates items from which you may select one; for example, ON|OFF indicates that you may specify ON or OFF. If you do not make a choice, a default of ON is assumed.

- ...

  Three dots in a command indicate that the parameter may be repeated more than once in succession.

The term absolute pathname refers to the full specification of a path starting from the root directory. Absolute pathnames always begin with the slash (/) symbol. A relative pathname takes the current directory as its starting point, and starts with any alphanumeric character (other than /).

Related documents

See the following documents for related information:

- 241-6001-011 *Preside MDM Fault Management User Guide*
- 241-6001-015 *Preside MDM Network Model Administrator Guide*
- 241-6001-203 *Preside MDM Alarm and Status API Reference Guide*
- 241-6001-303 *Preside MDM Administrator Guide*
- 241-6001-304 *Preside MDM Configuration Management for DPN Administration*
- 241-6001-308 *Preside MDM Network Configuration Database for DPN Administrator Guide*

Chapter 1
Server Daemon (SVMDMN)

This section contains information on the Server daemon. See the following topics for more information:

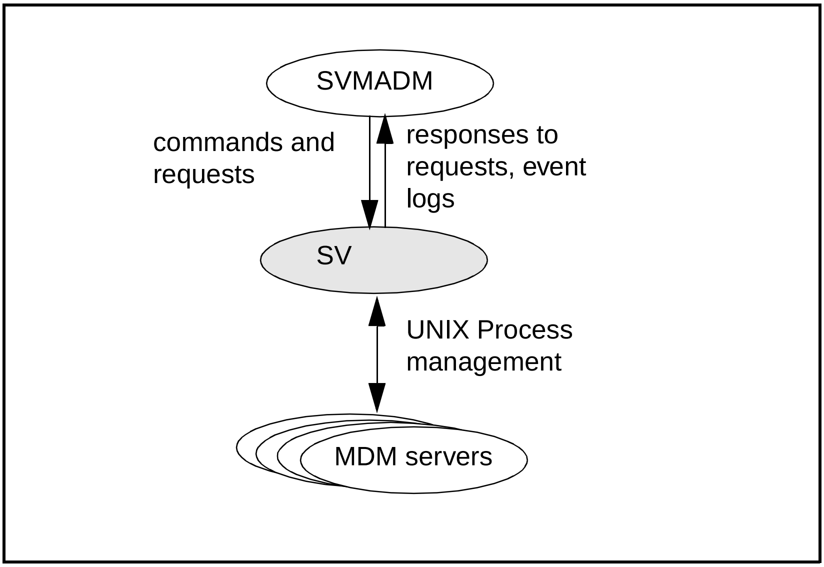
- “About the Server daemon” (page 39)
- “Managing the Server daemon” (page 41)
- “Exit codes” (page 42)
- “Error messages” (page 42)

About the Server daemon

The server daemon (SVMDMN) works with the Server Administration tool and the UNIX operating system to monitor and manage all of the other Preside Multiservice Data Manager (MDM) servers. The SVMDMN server performs the following functions:

- starts MDM servers when the workstation is rebooted
- restarts MDM servers when they exit abnormally
- communicates with the Server Administration tool to enable user interaction

Figure 1
SVMDMN data flow diagram



Security audit events

The SVMDMN server generates a signal to the OAMC server to generate security audit log events. The events logged are listed below; events that generate alarms are indicated. See “Log Collector (OAMC)” (page 199) for more information.

- MDM license (alarm)
- STARTED_EVENT (alarm)
- STOPPED_EVENT (alarm)
- ADDED_EVENT
- CHANGED_EVENT (alarm)
- DELETED_EVENT (alarm)
- MOVED_EVENT
- EXITED_EVENT (alarm)
- QUIT_EVENT (alarm)

- FAILED_EVENT (alarm)
- ADMIN_EVENT
- NOADMIN_EVENT
- BOOT_EVENT (alarm)
- SHUTDOWN_EVENT (alarm)
- PASSWD_EVENT

The SVMDMN server removes components in the fault stack by providing the raw state attribute. For example, when servers are renamed or removed from the server lists, the SVMDMN server generates logs that instruct the OAMC server to declare that the components do not exist.

Managing the Server daemon

The SVMDMN server operates automatically. The SVMDMN server is started automatically by the Solaris operating system when the workstation is rebooted. Other Preside Multiservice Data Manager servers can be monitored, started, and stopped through the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 41)
- “Startup command” (page 41)

Suggested name in Server Administration

The SVMDMN server does not contain an entry in /opt/MagellanNMS/cfg/SVMList.cfg. This file is the configuration file for the SVMDMN server.

Startup command

The startup command for the SVMDMN server has the following syntax:

```
/opt/MagellanNMS/bin/svmdmn [-noapi] [-noipc] \
[-nolog] [-nopoll] [-cfg <path>] [-log <path>] \
[-err <path>] [-poll <i>] [-hosts <n>] \
[-debug <n>] \
[-h]
```

where:

`-nolog` disables logging to disk

`-nopoll` disables polling for defunct children

`-cfg <path>` specifies a non-standard server manager configuration file

`-log <path>` specifies a non-standard server manager log file

`-err <path>` specifies a non-standard server manager error file

`-poll <i>` specifies the polling interval to reap dead child processes

`-hosts <n>` specifies an admin tool support limit

Exit codes

Exit codes for the SVMDMN server are shown in the following table.

Table 1
Exit codes for the SVMDN server

Exit code	Description
50	Help invoked
50	Told to terminate
254	Failed to register service (MNSD not coming up)
255	Could not open config file
255	Failed to initialize IPC

Error messages

Error messages for the SVMDMN server are shown in the following table.

Table 2
Error messages for the SVMDMN server

Error message	Meaning and action
build_server_list(): Fatal: Failed to open config file for reading	Fatal, could not read configuration file. Check /opt/MagellanNMS/cfg/SVMList.cfg
register_api_service(): Failed to initialize IPC	Fatal, could not initialize IPC environment. Is MNSD starting? Check the logs for MNSD errors.
Fatal, could not register service (MNSD not coming up?)	Check the logs for MNSD errors.
register_api_service(): Failed to register API service	Check the logs for MNSD errors.
path_is_valid(): Command length exceeds 1024 bytes - rejected	UNIX has a command line length limit. Specify a shorter command line.
path_is_valid(): Command path exceeds 255 bytes - rejected	UNIX has a path length limit. Specify a shorter path in the command.
path_is_valid(): Cannot start command path - rejected	Executable image specified by the command cannot be found. Correct the command.
path_is_valid(): Cannot execute command path - rejected	The image specified by the command is not executable. Either correct the file's permissions or specify a different command.
set_time_stamp(): Failed to get time of day	Try again. If the problem persists, contact your System Administrator.
log_to_file(): Failed to open log file for append	Verify permissions for log file. Also verify that there is available disk space. If problem persists, contact your System Administrator.
notify_clients(): Failed to init event PDU	Try again. If the problem persists, contact your System Administrator.
generate_dpn_alarm(): Failed to generate ACTIVATE CLEAR 30100000 MDM log	Try again. If the problem persists, contact your System Administrator.
(Sheet 1 of 4)	

Table 2 (Continued)
Error messages for the SVMDMN server

Error message	Meaning and action
generate_dpn_alarm(): Failed to generate OOS SET 30100000 MDM log	Try again. If the problem persists, contact your System Administrator.
api_handler(): Message received when -noapi set	Disregard this message.
api_handler(): Too many administrators	Close some Server Administration tool windows.
api_handler(): Unexpected command in pdu header	If problem persists, contact your System Administrator.
api_handler(): IPC message recv'd from endpoint not in client list	If problem persists, contact your System Administrator.
api_handler(): Lost connection from unknown client	If problem persists, contact your System Administrator.
api_handler(): Unexpected ipc status encountered	If problem persists, contact your System Administrator.
pdu_get_handler(): Unknown get command encountered	If problem persists, contact your System Administrator.
dump_all_servers(): Failed to init pdu struct for dump_all_servers	Try again. If problem persists, contact your System Administrator.
dump_server(): Invalid server index (out of range)	If problem persists, contact your System Administrator.
dump_server(): Failed to initialize PDU struct - dump_server()	If problem persists, contact your System Administrator.
dump_logs(): Failed to start log file - does not exist or no read perms	Check file permissions for log file and available disk space. If all appears O.K., contact your System Administrator.
dump_logs(): Failed to open log file for reading	Check file permissions for log file and available disk space. If all appears O.K., contact your System Administrator.
(Sheet 2 of 4)	

Table 2 (Continued)
Error messages for the SVMDMN server

Error message	Meaning and action
dump_logs(): Failed to seek to position in log file	If problem persists, contact your System Administrator.
dump_logs(): Failed to init pdu struct - dump_logs()	If problem persists, contact your System Administrator.
pdu_set_handler(): Unknown set command encountered	If problem persists, contact your System Administrator.
set_notify(): Request for events when already registered	If problem persists, contact your System Administrator.
set_notify(): Request for no more events when not registered	If problem persists, contact your System Administrator.
set_state(): Server title not found	If problem persists, contact your System Administrator.
set_state(): Request to stop server that is not running	If problem persists, contact your System Administrator.
set_state(): Request to start server that is already running	If problem persists, contact your System Administrator.
set_state(): Unexpected state specified	If problem persists, contact your System Administrator.
start_server(): Failed to execv new server	If problem persists, contact your System Administrator.
start_server(): Failed to fork new process	If problem persists, contact your System Administrator.
add_server(): Server already exists	Give new server a different name and/or command.
add_server(): Failed to start file	New command cannot be found. Correct the command.
add_server(): Bad execute permissions	New command is not executable. Change permissions or change command.
(Sheet 3 of 4)	

Table 2 (Continued)
Error messages for the SVMDMN server

Error message	Meaning and action
add_server(): Failed to save list to disk	Verify write permission to .cfg file. Verify disk space availability. If problem persists, contact your System Administrator.
change_server(): Specified server name not found	If problem persists, contact your System Administrator.
change_server(): Cannot change a permanent server's name	If problem persists, contact your System Administrator.
change_server(): Server name already exists	Pick a unique server name and try again.
change_server(): Cannot change path of permanent entry	If problem persists, contact your System Administrator.
change_server(): Failed to start new path	Command cannot be found. Correct the command.
change_server(): Bad execute permissions	Command is not accessible. Change permissions or change the command.
change_server(): Failed to save changed list to disk	Verify write permission to .cfg file. Verify disk space availability. If problem persists, contact your System Administrator.
delete_server(): Specified server title not found	If problem persists, contact your System Administrator.
delete_server(): Cannot delete a permanent server	If problem persists, contact your System Administrator.
delete_server(): Cannot delete a running server	Stop the server first, then delete the server.
delete_server(): Failed to save modified list to disk	Verify write permission to .cfg file. Verify disk space availability. If problem persists, contact your System Administrator.
write_config_file(): Failed to open the config file for writing	Verify write permission to .cfg file. Verify disk space availability. If problem persists, contact your System Administrator.
(Sheet 4 of 4)	

Chapter 2
Multi-nodal Name Server (MNSD)

This section contains the following information on the Multi-nodal Name server (MNSD):

About the MNSD

There are two types of Multi-nodal Name (MNSD) servers that provide the following capabilities:

- level 1 MNS server

  provides a place where software processes that are running on the same workstation to register so that they can communicate with one another. The MNSD is a server that processes register their name on so that other processes can look them up to establish inter-process communication.

- level 2 MNS server

  provides a few applications (Server Administration, Service Selection) with a list of available host names to select from.

Figure 2
MNSD Level 1data flow diagram

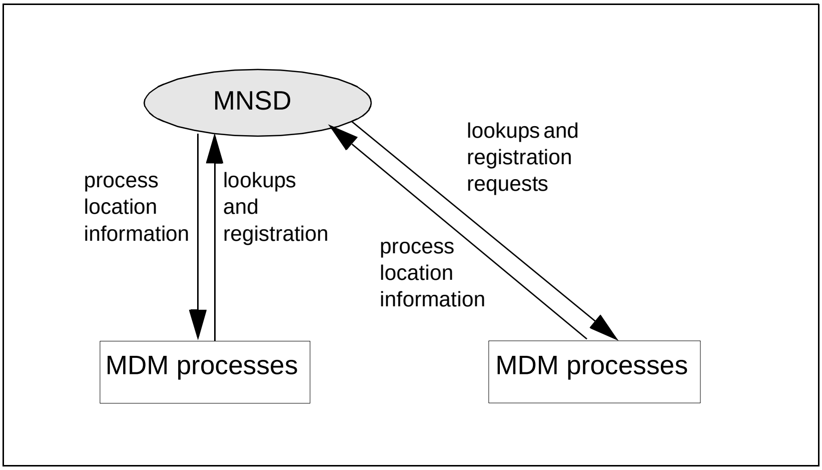
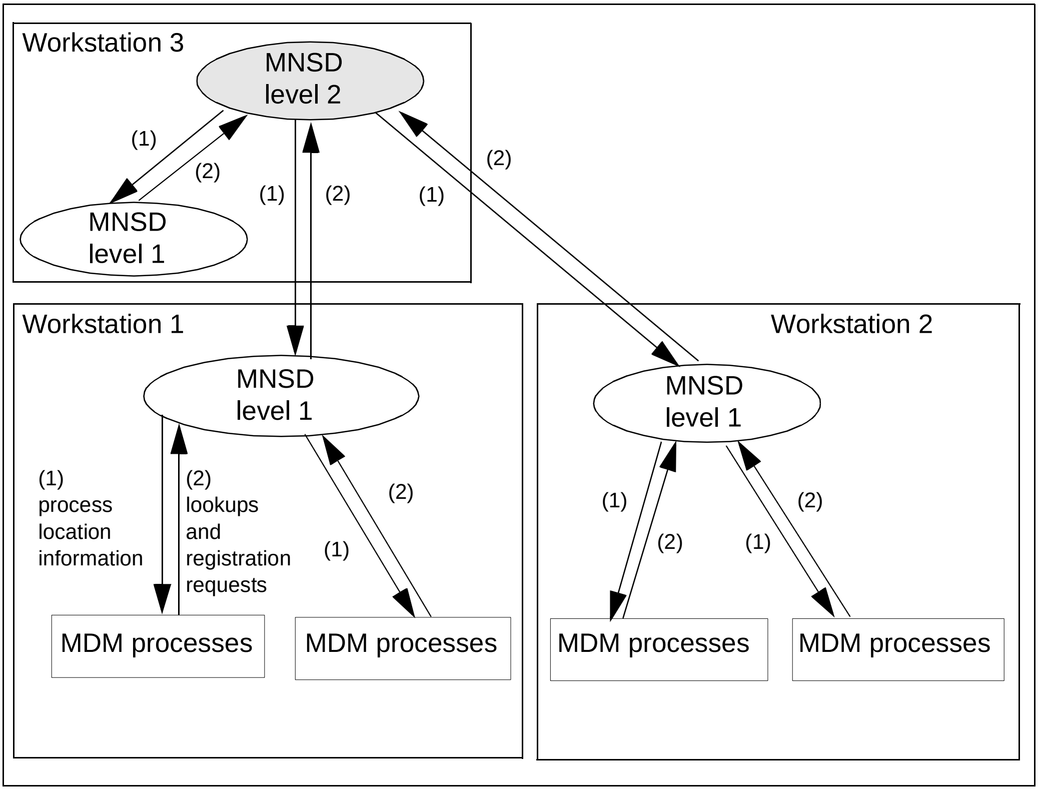

MNSD
process location information
lookups and registration
lookups and registration requests
process location information
MDM processes
MDM processes

Figure 3
MNSD Level 2 data flow diagram



Managing the MNSD server

Level 1 servers are implicitly started by the Server Administration tool. They should never be manually started.

For level 2 servers, use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. To configure the Multi-nodal Naming Service domains, see 241-6001-303 *Preside MDM Administrator Guide* for more information.

For more information, see the following:

- “Suggested name in Service Administration” (page 50)
- “Startup command” (page 50)

Suggested name in Service Administration

The recommended name for the MNSD level 2 server is MNSD Level 2.

Configuring MNSD with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start an MNSD server has the following syntax:

```
/opt/MagellanNMS/bin/mnsd [<level> <hostname>...]
```

where:

`<level>` specifies the MNSD server level. If you omit this parameter, level 1 is used by default. For a level 2 server, enter 2. You should always use 2.

`<hostname>...` are the host names of the workstations. For level 2 or above, the host names are specified for each additional workstation that contains processes to communicate with the level 1 processes.

For an explanation of how to set up MNS domains with level 2 MNS servers, see the section on configuring Multi-nodal Naming Service domains in 241-6001-303 *Preside MDM Administrator Guide*.

Configuration

To allow deployment across a communication firewall, you need to override the Preside Multiservice Data Manager (MDM) dynamic port allocation and predetermine the TCP/UDP port number to be used by MDM processes. For more information on configuring TCP/UDP port numbers, see 241-6001-303

Preside MDM Administrator Guide. For a list of the port numbers used by MDM servers, see “Server ports” (page 585). To predetermine TCP/UDP port numbers, use the following methods:

- Configure a range of TCP/UDP port numbers that MDM processes will be allowed to bind. See “Configuring a range of TCP/UDP ports” (page 51).
- Configure the TCP/UDP port numbers that a specific MDM server should bind. See “Configuring named-service TCP/UDP ports” (page 52).

The second method is preferable because it provides the most direct control, particularly across firewalls where communications occur with only a limited number of well-known MDM services (for example, GMDR for alarm access). You can use both methods simultaneously. If you do so, you do not need to constrain the port numbers configured by name to the range configuration.

> ***Note:*** There is no guarantee that the port numbers and ranges configured using these methods will only be allocated by the intended MDM processes (if an MDM process at all). If an MDM process cannot allocate a corresponding name-assigned port number (for example, if the port number is already in use), the process terminates.

UDP port numbers 5502 and 5503 are reserved for the MNSD daemon. Always allow communication to and from these ports through the firewall if MDM processes on either side need to communicate.

Port numbers configured by Operator Client agents cannot be configured using these methods. If you do not use the default ports, specify the ports on the agent command lines.

Configuring a range of TCP/UDP ports

Port range configuration applies to both TCP and UDP. To activate port range configuration, create the file /opt/MagellanNMS/cfg/private/IPCPortRange.cfg. Any lines in this file that start with a # sign (comments) or are empty are ignored. The file must contain a line with the following format:

```
<port range lower limit> <port range upper limit>
```

where:

`<port range lower limit>` and `<port range upper limit>` are the lower and upper range limits. These limits must be large enough to allow all Preside Multiservice Data Manager (MDM) processes to allocate the ports they need. As a general rule, assume 3 ports per server and 2 ports per client/utilities to be run at the same time on the workstation.

You can configure the firewall to allow communications with ports in a specified range. This range must have values above 1024 as the 1–1024 port range is reserved for standard well-known IP services such as FTP and Telnet (see `man services` for more information). For example, the following contents for the port-mapping file restricts the TCP/UDP service port numbers used by MDM processes from 11200 to 11699 (500 ports):

```
# allowed TCP/UDP port range for MDM processes
11200 11700
```

The firewall can then be configured to allow communications to and from this port range. Remember to allow communications with MNSD at port numbers 5502 and 5503.

Configuring named-service TCP/UDP ports

The named-service configuration is similar to the port range configuration. To activate named-service configuration, create the file /opt/MagellanNMS/cfg/private/IPCNameMap.cfg. Any lines in this file that start with a # sign (comments) or are empty are ignored. The file must contain a line with the following format:

```
<service name> <port number>
```

where:

`<service name>` is the name of the service.

`<port number>` is the number of the port.

As a guide, a prototype file, /opt/MagellanNMS/lib/IPCNameMap.cfg, lists the supported Preside Multiservice Data Manager (MDM) service names in comments.

> ***Note:*** Since the *IPCNameMap.cfg* file is scanned every time a new service is created, for efficiency reasons, it is recommend that you do not copy the entire prototype file to the */opt/MagellanNMS/cfg/private* directory. Instead, create the file with only those entries you want to map. Read and analyze the prototype file carefully before you configure your own port mappings. This file outlines server inter-dependencies that must be obeyed to allow the MDM communications to be carried out properly through the firewall.

You can configure the firewall to allow communications with ports mapped in the file. The specified port numbers must have values above 1024 as the 1–1024 port range is reserved for standard well-known IP services such as FTP and Telnet (see `man services` for more information). For example, the following contents for the name-mapping file restricts the TCP/UDP service port numbers used by the GMDR server (Surveillance server, Alarm&Status API) to the value 11201:

```
# Surveillance/Alarm&Status API service
GMDR 11201
```

The firewall can then be configured to allow communications to and from this port. Remember to allow communications with MNSD at UDP port numbers 5502 and 5503.

> ***Note:*** Some MDM services cannot be mapped to a TCP/UDP port number by name. For example, the FDTR service names are constructed at runtime and cannot be explicitly configured in advance in a name-map file. Only the port-range mechanism applies to these services. Consequently, some capabilities, such as Network Access, cannot be supported through a firewall with the name-mapping mechanism alone (use the port range mechanism as well). Refer to the prototype *IPCNameMap.cfg* file for the list of supported services.

Exit codes

Exit codes for the MNSD server are shown in the following table.

Table 3
Exit codes for the MNSD server

Exit code	Description
50	The server exits under the following conditions: - invalid command line - could not initialize the IPC environment - could not open MNSD database

Error messages

Error messages for the MNSD server are shown in the following table.

Table 4
Error messages for the MNSD server

Error message	Meaning and action
mnsd: bad scope specified	Fatal, invalid command line option. Revise the server configuration with the Server Administration tool.
mnsd <level>: could not ipc_init()	Fatal, could not initialize the IPC environment. Make sure the MNSD level 1 is running.
mnsd <level>: could not init object database	Fatal, the MNSD database cannot be opened. Check the permission of the /opt/MagellanNMS/cfg/private/mns* files and the directory.
mnsd <level>: could not ipc_register() or mnsd <level>: could not register gateway service	Fatal, could not register MNSD service name. MNSD level 1 may not be running or another MNSD of that level may be already running.
mnsd <level>: No legal hostnames given	A level 2 or above MNSD is missing hostnames on the command line. Revise the server configuration with the Server tool.

Chapter 3 Backup Controller (NSCTLBCK)

This section contains information on the Backup Controller. This server can be configured to provide basic backup capabilities. It can also be configured to provide extended backup capabilities for Passport 6000, and Passport 7400, 15000, 20000. In this case, it is referred to as the Backup Server. Only one server can be configured. See the following topics for more information:

- “Backup Controller” (page 55)
- “Backup Controller as Backup Server” (page 59)

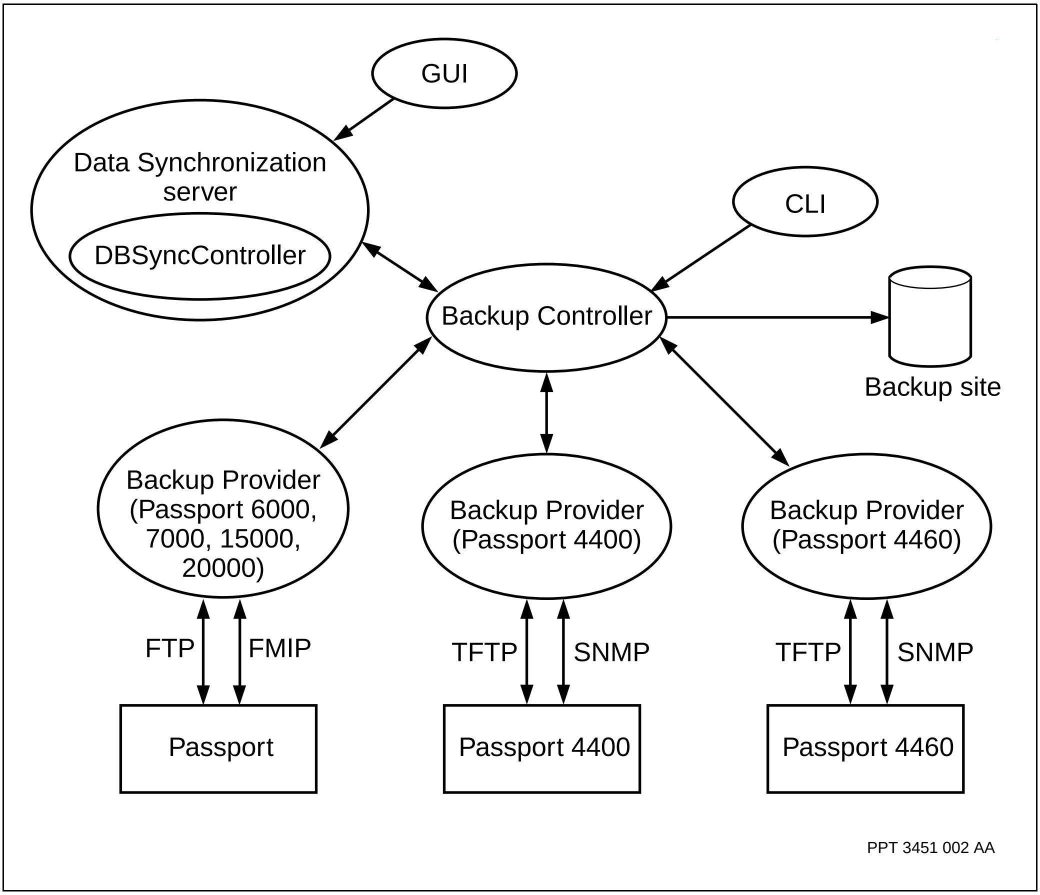
Backup Controller

The Backup Controller receives requests from Passport/SNMP Backup tools (GUI and CLI) and connects to the appropriate Backup Provider. See the diagram “Backup Controller data flow diagram” (page 56) for an illustration of data flow.

See 241-6001-807 *Preside MDM Network Backup and Restore* for more information.

See the following section “Managing the Backup Controller” (page 56) for more information on the Backup Controller.

Figure 4
Backup Controller data flow diagram


GUI
Data Synchronization server
DBSyncController
CLI
Backup Controller
Backup site
Backup Provider (Passport 6000, 7000, 15000, 20000)
Backup Provider (Passport 4400)
Backup Provider (Passport 4460)
FTP
FMIP
TFTP
SNMP
TFTP
SNMP
Passport
Passport 4400
Passport 4460
PPT 3451 002 AA

Managing the Backup Controller

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 57)

- "Backup Controller startup command" (page 57)

Suggested name in Server Administration

The recommended name is Backup Controller.

Configuring the Backup Controller with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Backup Controller startup command

The startup command for the Backup Controller is as follows:

```
/opt/MagellanNMS/bin/nsctlbck [-p <port_no>]
  [-c <remote_mapping_file>]
  [- ll <level>]
  [-h]
```

Use the following table to substitute command parameters:

Parameter	Definition
`-p <port_no>`	is the port number the Backup Controller uses. The port number is dynamically assigned but you can also specify it as port number 5000. To launch the Passport/SNMP Service Data Backup interface, you must specify port number 5000.
`-c <remote_mapping_file>`	is the name of the remote mapping file. The default remote mapping file is /opt/MagellanNMS/cfg/Controller.cfg This file only needs to be changed if the Backup Provider is running on a different workstation than the Backup Controller.
(Sheet 1 of 2)	

Parameter	Definition
`-ll <level>`	sets the logging level used by the Backup Controller: error, warning, notice, info, debug, or trace
`-h`	displays command line usage
(Sheet 2 of 2)	

Backup Controller logging

The OAMC server collects logs from the Backup Controller. These logs can also be sent directly to a log file. See the diagram “Backup Server architecture” (page 61). The Backup Controller logs are stored in /opt/MagellanNMS/data/log/bckRst/backup.dlog.

Logging is automatically enabled as a default. The Backup Controller has the following logging levels: `error, warning, notice, info, debug, trace`. The default is `notice`. The initial logging level is set with the command line option `-ll <level>`, for example, nsctlbck `-ll debug`. This starts the Backup Controller with the logging level of `debug` and also includes `trace`. Similarly, if the level `error` is set, it includes all the other logging levels. See “Backup Controller startup command” (page 57) for more information.

Backup Controller as Backup Server

The -notification option on the command line of the server is what changes the Backup Controller into a Backup Server. When you run the Backup Controller as the Backup Server, all the functionality of the Backup Controller, described in “Backup Controller (NSCTLBCK)” (page 55), is available. For more information, refer to 241-6001-807 *Preside MDM Network Backup and Restore*.

The -notification option turns on the alarm-triggered backup function. The -notification option is required by the Backup Server to notify the Data Synchronization Server, so if the Backup Server is not turned on, automatic synchronization of the Administration Database cannot occur.

The Backup Server retrieves and stores the view and journal files to be available for node “restore” operations, and also for database synchronization.

The Backup Server retrieves and backs up the following:

- journal log files as soon as they are activated on the node, as well as the committed view associated with them
- the current view, if it exists
- the AV list of the current view

Therefore, in the event of a major disruption that requires a node to be replaced, the Restore tool would be able to trigger a software download, restore the last committed view and its associated journal log files, and activate to restore the latest current view on the replacement node. To be properly restored, the replacement node would need to have the same name as the original. For additional information on node recovery, see 241-6001-807 *Preside MDM Network Backup and Restore*.

For database synchronization, the Backup Server provides the following specific roles:

- manages the retrieval and storage of the views and journal files for synchronizing. The Backup Server receives alarms from GMDR, queues them, and performs the backup.

- notifies the Data Synchronization Server each time a backup happens to trigger the updates to the Administration Database. The notifications contain the details of the device's current view, journals, timestamps and state. This information is stored to disk so that the Data Synchronization Server can determine the current state of each device upon start-up.

For additional information on database synchronization, see 241-6001-400 *Preside MDM Administration Database User Guide*.

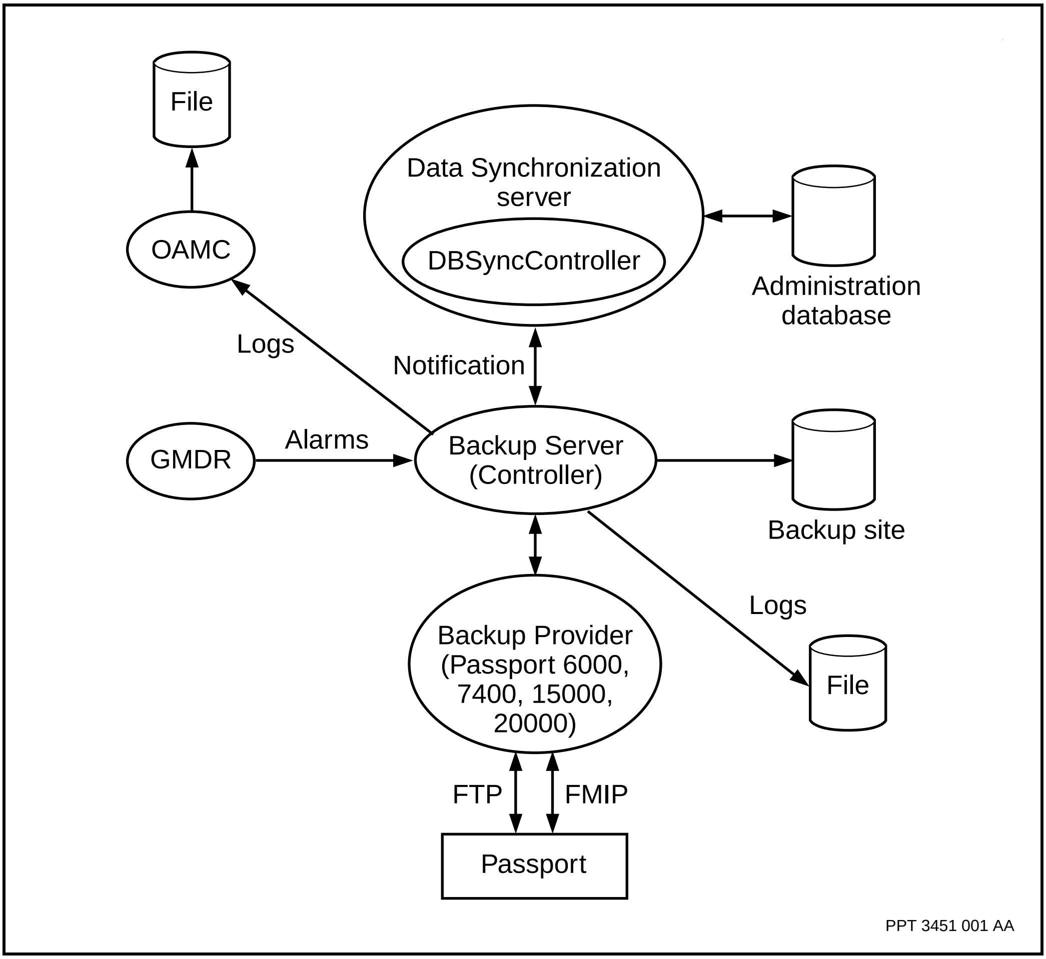
The Backup Server can be triggered to perform a backup in the following ways:

- **on alarm with journaling supported**
  For nodes that support journaling, a journal log file is created during each activation and contains a delta of the configuration changes between the current and edit views.

  The alarm generated after a confirm prov, commit prov, reset of the switch, or workstation reconnect to the node triggers the Backup Server to start a backup for those nodes that belong to groups that are specified in the DataSync.cfg file to be backed up on alarm.
- **on alarm for complete view (with journaling not supported)**
  Nodes that do not support journaling only send out a confirm prov alarm to trigger the Backup Server to backup recently created view files.
- **on demand through a scheduled cron job**
  Backups can be done on demand through the command line. You can define a cron job for scheduled backups.

The figure “Backup Server architecture” (page 61) shows the backup process.

Figure 5
Backup Server architecture



Configuration

The DataSync.cfg file is associated with the Backup Server. This file specifies the following:

- groups that you want to backup, user ID, and encrypted password.
- if backup on alarm is supported
- if you want to synchronize the Administration Database

- DBSyncController Name lets you specify the DBSyncController, which is responsible for the node.

For information on how to configure the Backup Server, refer to 241-6001-807 *Preside MDM Network Backup and Restore*.

Managing the Backup Server

Use the Server Administration tool to enter or to edit the startup command and to start, stop, and configure this server to start automatically when the workstation is rebooted. Any changes you make to the startup command or to the configuration file become active whenever the daemon is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 62)
- “Startup command” (page 62)

Suggested name in Server Administration

The recommended name for the server is Backup Controller.

Configuring the Backup Server with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The following is the startup command for the Backup Server.

```
/opt/MagellanNMS/bin/nsctlbck -notification
  [-p <port_no>]
  [-c <remote_mapping_file>]
  [-gmdrhost <hostname>]
  [-nbofbck <#>]
  [-DB_Synch_port <port_no>]
  [-h]
```

Use the following table to substitute command parameters:

Parameter	Definition
`-p <port_no>`	is the port number the Backup Server uses. The port number is dynamically assigned but you can also specify it as port number 5000.
`-c <remote_mapping_file>`	is the name of the remote mapping file. The default remote mapping file is /opt/MagellanNMS/cfg/Controller.cfg This file only needs to be changed if the Backup Provider is running on a different workstation than the Backup Controller.
-`gmdrhost <hostname>`	is the host where GMDR is running. The default is localhost.
`-nbofbck <#>`	is the maximum number of parallel backups. The default is 5.
-`DB_Synch_port <port_no>`	is the port for communication with the DBSyncController. The default is 5050.
-h	displays command line usage

Backup Server interdependencies

The Backup Server has the following interdependencies:

- GMDR server. The Backup Server receives alarms from the GMDR server. If the GMDR runs on a different host than the Backup Server, you need to specify the GMDR host in the Backup Server command.
- Passport Backup Provider. You need to start the Passport Backup Provider before the Backup Controller.
- Data Sync Server. The Backup Controller notifies the Data Sync Server each time a backup occurs, to trigger the updates to the database.
- Backup Site. The Backup Server retrieves and backs up journal log files as soon as they are created on the node, as well as the committed view associated with them, and the AV list of the current view and compares this information to the contents of the disk at the Backup Site.

Backup Server exit codes

Exit codes for the Backup Server are shown in “Exit codes for the Backup Server” (page 64).

Table 5
Exit codes for the Backup Server

Exit code	Description
21	Lost connection to GMDR

Backup Server error messages

Log files are stored at the following location:

/opt/MagellanNMS/data/log/bckRst/backup.dlog

Chapter 4
Context Server (CTXSVR)

This section contains information on the Context server (CTXSVR). See the following topics for more information:

About the CTXSVR server

The CTXSVR server provides a way for processes running on the workstation to communicate with each other by putting values into context, or by getting values that have been previously put into context. The CTXSVR server maintains the (key, content) pairs and responds to requests to put and get the values.

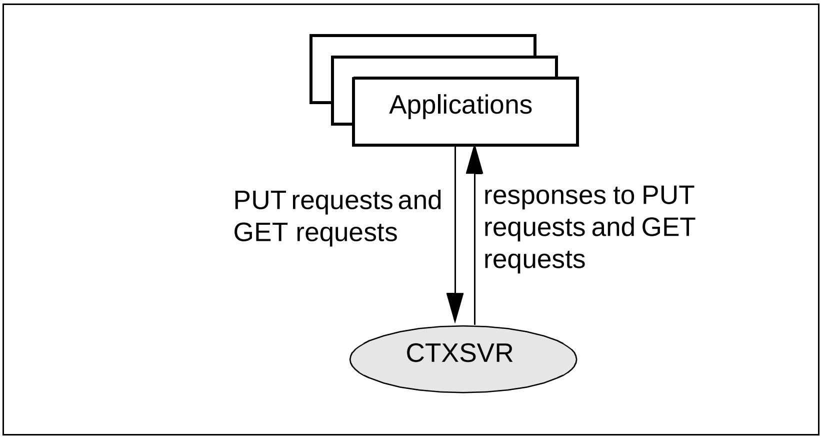
There are two sets of contexts: USER and WORKSTATION. The workstation contexts are a set of (KEY, VALUE) pairs that are available to all the applications on the workstation; for example, default service selections. The user contexts are a set of (KEY, VALUE) pairs per active user sessions which are available to the user sessions applications. For example, hot context. A value is “in context” when it is the VALUE of a particular (KEY, VALUE), pair.

A set of functions is provided for sending requests to the CTXSVR server and decoding the responses.

The context server actually maintains two sets of contexts:

- the workstation context, which is available to all applications on the workstation, and is usually persistent across workstation reboots (for example, default service selections)
- the user session contexts (one per active user session), which is only available to the user session's applications (for example, Hot Component context)

Figure 6
CTXSVR data flow diagram


Applications
PUT requests and GET requests
responses to PUT requests and GET requests
CTXSVR

Managing the CTXSVR server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the CTXSVR. Any changes you make to the startup command or options take effect when the CTXSVR is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- "Suggested name in Server Administration" (page 67)
- "Startup command" (page 67)

Suggested name in Server Administration

The recommended name for the CTXSVR is Context Server.

Configuring CTXSVR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the CTXSVR server has the following syntax:

```
/opt/MagellanNMS/bin/ctxsvr
```

Interdependencies

There are no interdependencies.

Exit codes

The CTXSVR server exits when the Context Server’s end point is not created. Exit codes for the CTXSVR server are shown in the following table.

Table 6
Exit codes for the CTXSVR

Exit code	Description
244	Could not initialize IPC system
48	Could not register service name (already running?)

Error messages

Error messages for the CTXSVR are shown in the following table.

Table 7
Error messages for the CTXSVR

Error message	Meaning and action
CTXSVR: Unable to write workstation context variable entry to disk	Non-fatal, could not write entries to disk. Check the access mode on /opt/MagellanNMS/cfg/private/CTXSvr.cfg

Chapter 5
Customer Database Server (CDBSERVER)

This section contains information about the Customer Database Server (CDBSERVER). See the following sections for information about this server:

- About the CDBSERVER (page 69)
- Managing the CDBSERVER (page 69)
- Interdependencies (page 70)

About the CDBSERVER

The Customer Data Server (CDBSERVER) provides users with access to a database that contains custom information for a customer's network. A customer creates this database.

The database can contain any information that the customer wishes to maintain. Typically a customer uses the database to store information about specific components in the network, such as phone numbers, circuit numbers, and port numbers. An example of the information stored in the database, is the number of a port on an interface card on a node, and the telephone number, physical address, and e-mail address of the customer whose equipment is connected to that port.

Managing the CDBSERVER

See the following topics for information about managing the CDBSERVER:

- Configuration (page 70)
- Suggested name in Server Administration (page 70)
- Start-up command (page 70)

Configuration

Configuring the CDBSERVER is part of a larger task: Creating a Customer Database. For procedures to set up and maintain a Customer Database, see 241-6001-804 *Preside MDM Workstation Utilities User Guide*.

Suggested name in Server Administration

The recommended name for CDBSERVER to enter in the Server Administration tool is Cust Data Server.

Configuring the CDBSERVER with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the CDBSERVER server has the following syntax:

```
/opt/MagellanNMS/bin/cdbserver <database>
```

where:

`<database>` is the full path name of the customer database you wish to create

example:

```
/opt/MagellanNMS/bin/cdbserver /usr/db/
Apr_29_CustData
```

Interdependencies

None.

Chapter 6
DPN DBNL Auto-disabling Daemon (DBNLWatch)

This section contains information on the DBNL auto-disabling daemon (DBNLWatch). See the following topics for more information:

About DBNLWatch

In networks that only contain Passport nodes, this daemon is not required, and should not be started.

The DBNLWatch auto-disabling daemon monitors alarms from the DPN switches in the network. When it detects the presence of a DBNL activation alarm or a DBNL heartbeat alarm indicating that a DBNL has been activated, it sets up a watch on the DBNL. Using a set of utilities (dbnlfindam, dbnlcheck, dbnldisable, and dbnlenable) DBNLWatch obtains information about changes in the status of the primary network link and the DBNL and writes them to a log file. Optionally, it deactivates the DBNL when the primary link returns to service and remains in service for a specified time period.

It also provides responses to queries about DBNLs being watched from utility dbnlapi.

Managing the DBNLWatch

Use the Server Administration tool to enter or to edit the startup command and to start, stop, and to configure this daemon to start automatically when the workstation is rebooted. Any changes you make to the startup command or to the configuration file become active whenever the daemon is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide* for instructions.

For more information see the following:

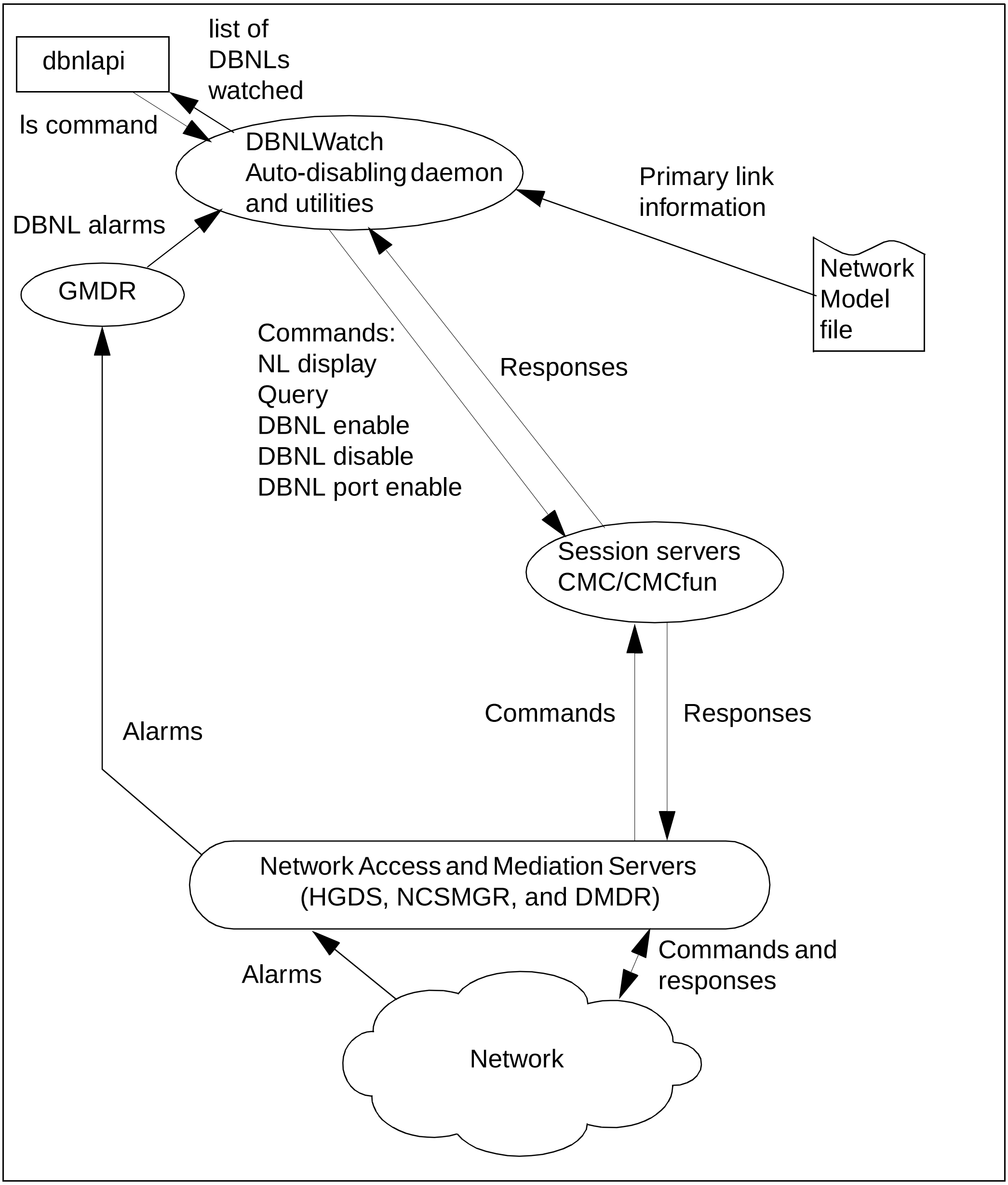
- “Suggested name in Server Administration” (page 72)
- “Startup command” (page 74)

Suggested name in Server Administration

The recommended name for the Auto-disabling server daemon is
DBNL Watch.

Configuring DBNLWatch with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Figure 7
DBNL auto-disabling server daemon data flow diagram



- “Suggested name in Server Administration” (page 72)
- “Startup command” (page 74)

Startup command

The startup command for the DBNL auto-disabling daemon is as follows:

```
/opt/MagellanNMS/bin/dbnlwatch [-v] [-n] \
[-host <GMDR host name>] [-serv <GMDR service name>] \
[-log <log file prefix>] \
[-cfg <configuration filename>]
```

where:

`-v` specifies the verbose option, and provides detailed information in the log files output by DBNLWatch.

`-n` specifies that no log files are to be output. If this parameter is omitted, log files are output.

`-host <host name>` is the name of the host that is running the GMDR server and is to supply DBNL alarm information. This parameter is only required if the GMDR server is not running on the same workstation as DBNLWatch. If this parameter is omitted, the default localhost (this workstation) is used.

`-serv <service name>` is the service name for the GMDR server. This parameter is only required if the service name of the server is not GMDR. If this parameter is omitted, the default GMDR is used.

`-log <log file prefix>` specifies an alternate stem to the name of the file in which logs are output. If this parameter is omitted, the default stem /opt/MagellanNMS/data/DBNLWatch.log is used.

`-cfg <configuration filename` specifies the full pathname of a configuration file containing parameters for DBNLWatch. If this parameter is omitted, the default /opt/MagellanNMS/cfg/DBNLWatch.cfg is used.

Interdependencies

DBNLWatch requires that

- the network access servers (HGDS and NCSMGR) be configured and running
- the mediation servers that provide alarm information (DMDR and GMDR) be configured and running
- the Network Model has been properly configured, is committed (startup), and contains information about DPN modules node names and their nams_ids

DBNLWatch does not require the use of another user's session servers (CMC and CMCFun) because it starts its own session servers.

Configuration

The following file is involved in configuring the DBNLWatch:

/opt/MagellanNMS/cfg/DBNLWatch.cfg

This file contains parameters for the following items:

- settings for the timers and counters used by DBNLWatch and the dbnlfindam, dbnlcheck, dbnldisable, and dbnlenable utilities
- login information for the primary OA and backup OA through which DBNLWatch sends commands to the NCS

Once the file has been configured, DBNLWatch must be started using the Server Manager Administration tool to make the configuration active.

The parameters in the file are shown in the following table:

Table 8
Parameters in file DBNLWatch.cfg

Name	Description	Defaults
CheckTimer	interval in seconds between each check to determine if the primary network link is up or down	30 seconds
CheckTries	number of successive checks showing that the primary network link is up before the DBNL can be declared unnecessary and disabled	5 checks
DisableTimer	interval in seconds between attempts to disable a DBNL	30 seconds
EnableWait	period of time in seconds that DBNLWatch waits after disabling the DBNL before it attempts to re-enable the DBNL port	5 seconds
EnableTries	number of attempts at disabling the DBNL before DBNLWatch goes back to checking if the primary network link is up	5 attempts
EnableTimer	interval in seconds between attempts to enable a DBNL port before DBNLWatch goes back to checking if the primary network link is up	30 seconds
DisableTries	number of attempts at enabling the DBNL port before DBNLWatch goes back to checking to see if the primary link is up	5 attempts
MaxWatchTime	maximum length of time in hours that a DBNL can be watched DBNLWatch ceases to watch the DBNL, as measured from the arrival of the DBNL enable alarm or the most recent DBNL heartbeat alarm. ***Note:*** A new watch will start on the DBNL if DBNLWatch receives a DBNL heartbeat alarm.	48 hours
(Sheet 1 of 2)		

Table 8 (Continued)
Parameters in file DBNLWatch.cfg

Name	Description	Defaults
MonitorOnly	Boolean (0 or 1) indicating whether DBNLs can be monitored and disabled or just monitored. This only applies to DBNLs that are enabled because of AM or AM cluster isolation.	0 (monitoring and disabling is allowed)
MonitorBWOD	Boolean (0 or 1) indicating whether DBNLs that are activated by the NCS bandwidth on demand (BWOD) feature are tracked or ignored	0 (ignored)
PrimaryOAAuth	the OA name, NCS_capability_id, and password required to log on to the primary OA in the NCS and send commands to NCS	None
BackupOAAuth	the OA name, NCS_capability_id, and password required to log on to the backup OA in the NCS and send commands to NCS. This is the OA that is used to access the NCS if the primary OA fails. If no backup OA is configured in the NCS comment out the BackupOAAuth entry in the file.	None
(Sheet 2 of 2)		

Exit codes

Exit codes for the DBNLWatch are shown in the following table.

Table 9
Exit codes for the DBNLWatch

Exit code	Description
1	GMDR communication failure
2	Command Access Session server problems
(Sheet 1 of 2)	

Table 9 (Continued)
Exit codes for the DBNLWatch

Exit code	Description
53	Could not post service (already running)
56	Configuration file error
(Sheet 2 of 2)	

Error messages

When an error occurs, messages are displayed in the System Log Display and written to the OAM log. Error messages for the DBNLWatch are shown in the following table.

Table 10
Error messages for the DBNLWatch

Error message	Meaning and action
DBNLWatch - Invalid configuration file DBNLWatch - Invalid Primary authentication in configuration file DBNLWatch - Invalid Backup OA Authentication in configuration file DBNLWatch - Configuration does not specify a primary OA authentication	The configuration file contains an error. Edit the configuration file, correct the error, then restart DBNLWatch.
DBNLWatch - Could not register query service	DBNLWatch is probably already running on this workstation.
DBNLWatch - Could not connect to the GDMR server.	There is a communication problem with the GMDR server. Use the Server Administration tool to verify that the GMDR server is up and running and is reachable.
(Sheet 1 of 2)	

**Table 10 (Continued)
Error messages for the DBNLWatch**

Error message	Meaning and action
DBNLWatch - Failed to register with the GMDR server DBNLWatch - Failed to create Alarm Sieve DBNLWatch - Error with Alarm Sieve to the GMDR server DBNLWatch - Lost connection to the GMDR server	
DBNLWatch - Failed to launch private session servers DBNLWatch - Lost one of the session servers	There is a problem with one of the session servers CMC or CMCFun started by DBNLWatch. Restart DBNLWatch.
(Sheet 2 of 2)	

Chapter 7
DPN Management Data Router (DMDR)

This section contains information on the DPN Management Data Router (DMDR). See the following topics for more information:

About the DMDR server

In networks that only contain Passport nodes, this server is not used and should not be started.

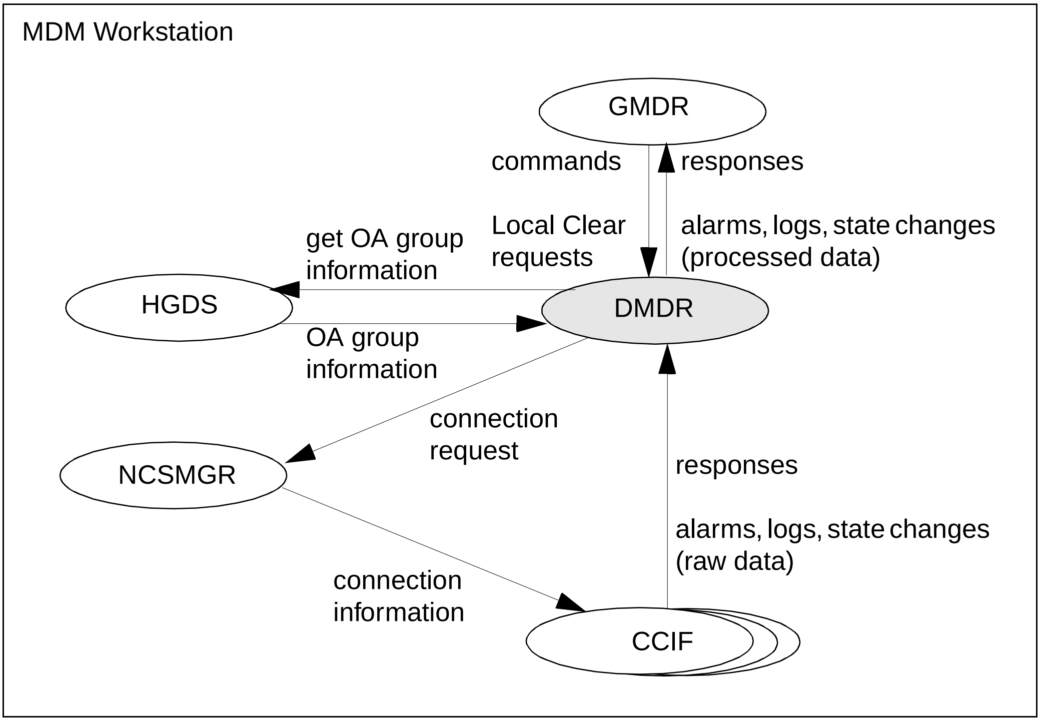
The DMDR server processes raw data received from the CCIF process, calculates the state of the DPN components monitored by NCS OAs, and forwards this processed information to its GMDR client servers.

At startup, the DMDR server sends requests to the NCSMGR server, to be connected to the OAs in its OA group.

At runtime, the DMDR server converts surveillance data to internal protocol (API format), filters information based on its clients’ CNMID, handles Local Clear requests received from the GMDR server, and creates proxy alarms when the state of components is changed by a status record. Optionally, the DMDR server fully records the active alarms in memory so that a new client can be given an up-to-date active alarm list (AAL) when it registers.

The DMDR server can also optionally record selected types of status records in memory to support clients' GET requests.

Figure 8
DMDR data flow diagram



Managing the DMDR server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for DMDR. Any changes you make to the startup command or options take effect when DMDR is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

See the following for more information:

- “Suggested name in Server Administration” (page 83)
- “Startup command” (page 83)

Suggested name in Server Administration

The recommended name for the DMDR server is DMDR_<OA_name>. An example of an entry for a DMDR server is:

```
DMDR_GROUPA
```

where:

`GROUPA` is the OA group name used by the DMDR server.

Configuring DMDR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start a DMDR server has the following syntax:

```
/opt/MagellanNMS/bin/dmdr \
-c <capability ID> \
-g <OA group name> \
-p <password> \
[-a] \
[-b <msg congestion threshold>] \
[-d] \
[-e <exceptions file name>] \
[-f]
[-l <reconnection delay length>] \
[-n <NCSMGR host name>] \
[-t <status record type>]...\
[-x] \
[-B] \
[-C <CNMID file name>] \
[-L]
```

```
   [-P <executable location>] \
   [-m]
   [-h]
```

where:

`-c <capability ID>` specifies the capability ID used to connect to OAs

`-g <OA group name>` specifies the name of the DMDR OA group

`-p <password>` specifies the password used to connect to OAs

`-a` suppresses full recording of active alarms in memory by the DMDR server. Active alarms are fully recorded if this parameter is omitted

`-b <msg congestion threshold>` specifies the number of congested replies before the client is cut off. If msg congestion threshold is not specified, the default value of 1000 is used.

`-d` puts the DMDR server into debugging mode. This option is not intended for use in the field.

`-e <exceptions file name>` specifies the absolute path of a file specifying the alarm exceptions database contents. The DMDR server uses the file /opt/MagellanNMS/cfg/DMDRAlarmExcep.cfg if this parameter is omitted.

`-f` clears any SET alarm indicating that a component that is out of service when a status record is received indicating that the component is back in service but no corresponding CLEAR alarm has been received.

`-l <reconnection delay length>` specifies, in seconds, the length of time the DMDR server waits before checking for lost modules when it reconnects to an OA. A value of 30 seconds is used by default.

`-n <NCSMGR and HGDS host name>` specifies a host name on which the NCSMGR and HGDS servers are running. This parameter is not recommended for use in the field. By default, the local host name is used.

`-t <status record type>` specifies a type of status record that must be stored in memory by the DMDR server. This parameter can be repeated for different status record types. The possible values are CM (common subsystem), PROC (PE status), NL (Network link), TRK (trunk), X25 (X.25 gateway), X75 (X.75 gateway) or all (all of the above). Status records are not stored if this parameter is omitted.

`-x` suppresses client authentication. This parameter is not intended for use in the field. By default, client authentication is performed for each client.

`-B` specifies that babbler alarms are forwarded to clients. This parameter is optional. If it is present, the `-a` option cannot be used. If it is not present, babbler alarms are discarded by default.

`-C <CNMID file name>` specifies the absolute path of a file specifying the CNMID of components allocated to a Virtual Private Network (VPN). The DMDR server uses the file /opt/MagellanNMS/cfg/DMDRCnmid.cfg if this parameter is omitted. This file can be built using the utility /opt/MagellanNMS/bin/dmdrcid. For information on this utility, see “Configuring file /opt/MagellanNMS/cfg/DMDRCnmid.cfg” (page 89).

`-L` turns off population of DMDR’s internal database with information about links. If this option is not specified, the DMDR server defaults to generating link information.

`-P <executable location>` specifies where the file containing DMDR executable code is located. This parameter is not intended for use in the field. By default, the file /opt/MagellanNMS/bin/dmdr contains DMDR executable code.

`-m` discards control down, office, and PE summary status records from Legacy Data Modules (LDMs) and Legacy Expansion Modules (LEMs) if the MPA in which they are located does not exist in the DMDR server’s internal database.

`-h` displays the help information.

LDMs and LEMs are cards that fit into Passport 4400 series devices and are equipped with ports for services like X.25, ITI, FR, TR, SNA. LDMs and LEMs generate four types of status records: call control down status records, office status records, PE summary status records, and PE status records. Only PE status records carry a field that identifies the component that originated the record, the others do not. If you specify the -m option, DMDR discards all records except PE status records. When DMDR receives a PE status record DMDR uses the PE type contained in the PE status record to create a component identifier for the LDM or LEM in its internal database. Once DMDR has created the component identifier in its internal database, the DMDR server keeps all types of status records that it receives.

Interdependencies

The DMDR server relies on the HGDS and NCSMGR servers.

Configuration

The following files are involved in configuring the DMDR server:

- /opt/MagellanNMS/cfg/HGDS.cfg

  contains a list of the groups of DPN OAs, the members in each group, and the communications parameters required to access each OA

- /opt/MagellanNMS/cfg/DMDRAlarmExcep.cfg

  contains action codes that identify the exceptions (special processing) required by exception alarms

- /opt/MagellanNMS/cfg/DMDRCnmid.cfg

  contains lines of data that specify the CNMID for VPN components

Configuring file /opt/MagellanNMS/cfg/HGDS.cfg

The Host Group Directory Information file /opt/MagellanNMS/cfg/HGDS.cfg contains a list of the groups of DPN OAs, the members in each group, and the communications parameters required to access each OA. The communications parameters are OA name, DNA, CUG index, packet size, X.75 use, and RPOA number.

> ***Note:*** This file also contains information about Passport groups. However, the DMDR server only uses fields that apply to groups of DPN OAs.

Before any DMDR server is started from the Server Manager Administration tool or, if after a reboot, before system initialization, file /opt/MagellanNMS/cfg/HGDS.cfg must reside on each workstation running the DMDR server. The file must be updated on all workstations on which it resides if any changes are made to the network element addresses or grouping. For a description of this file, see “HGDS server information file” (page 179).

Configuring file /opt/MagellanNMS/cfg/DMDRAlarmExcep.cfg

The alarm exceptions file, /opt/MagellanNMS/cfg/DMDRAlarmExcep.cfg, contains action codes that identify the exceptions required by exception alarms. The file /opt/MagellanNMS/cfg/DMDRAlarmExcep.cfg enables the DMDR server to manage these exceptions as required.

Each line in an alarm exceptions file has the following syntax:

```
<type> <fault code> <action> <comment>
```

where:

`type` specifies the alarm type; one of SET, CLR, MSG or COM. A line with the type COM is treated as a comment, therefore is ignored.

`fault code` is a string 8 characters long or less, formed of hexadecimal digits, the single character wildcard (?), or the end-of-string wildcard (*). If the string is fewer than 8 characters long, the string must be terminated by an end-of-string wildcard.

The single character wildcard (?) matches any single character in the alarm fault code.

The end-of-string wildcard (*) matches any sequence of characters up to the end of the alarm fault code. It can only occur at the end of the character string.

If the fault code of an alarm matches several entries, the entry that matches the longest initial portion of the alarm fault code without wildcards is used.

`action` is a numerical code defining the action for the DMDR server to take. For a list of the codes and the action applied for each code, see the table “Codes and actions applied by the DMDR server” (page 88).

`comment` is any text occurring on the line after the action code. It is treated as a comment, therefore is ignored. Lines beginning with an exclamation mark (!) are also treated as comments and are ignored.

The alarm exceptions file must contain four entries with action codes 20, 21 and 22 present in the file originally installed; otherwise, the corresponding alarms is not be processed correctly. The user can add entries with action codes 1, 30, 31, 32 and 33 to customize processing of some alarms.

The following is an example of an Alarm Exception file:

```
CLR    10164020  20     DBNL deactivation
CLR    10164021  21     DBNL activation
CLR    FFFFFFFF  22     NCS Undefine
CLR    FFFFFFFF  22     NCS Undefine
```

Table 11
Codes and actions applied by the DMDR server

Code	Action	Impact
0	no exception	normal processing
1	discard the alarm	can be applied to any alarm type. It will have no impact on component states and will not be forwarded to DMDR clients.
20	special processing for the Dial Backup Network Link (DBNL) deactivation alarm 1016 4020	Codes 20 to 22 are reserved for special purposes alarms.
21	special processing for the DBNL activation alarm 1016 4021	
22	special processing for the NCS Undefine alarms 04FF FFFF and 07FF FFFF	
(Sheet 1 of 2)		

Table 11 (Continued)
Codes and actions applied by the DMDR server

Code	Action	Impact
30	set the severity of this alarm to CRITICAL	Codes 30 to 32 only apply to SET alarms. The alarm severity is normally derived from the NCS severity attribute specified by the switch. Action codes 30 to 33 specify this severity directly.
31	set the severity of this alarm to MAJOR	
32	set the severity of this alarm to MINOR	
33	set the severity of this alarm to WARNING	
(Sheet 2 of 2)		

Configuring file /opt/MagellanNMS/cfg/DMDRCnmid.cfg

File /opt/MagellanNMS/cfg/DMDRCnmid.cfg specifies the CNMID (Customer Network Management Identifier) for each component

The syntax of each line in the file /opt/MagellanNMS/cfg/DMDRCnmid.cfg is:

```
::<Component name>:<Component CNMID>:
```

For example, the following lines preload CNMID information for the ports of PM MOD01 PE 10 PI 12:

```
::PM MODO1 PE 10 PI 12 PO 1:36:
::PM MODO1 PE 10 PI 12 PO 2:25:
::PM MODO1 PE 10 PI 12 PO 3:3:
::PM MODO1 PE 10 PI 12 PO 4:6:
```

File /opt/MagellanNMS/cfg/DMDRCnmid.cfg needs to be configured if a component is reactivated, or if a DMDR server is restarted. To configure file /opt/MagellanNMS/cfg/DMDRCnmid.cfg, the following script can be executed to extract the information from the NRS database:

```
/opt/MagellanNMS/bin/dmdrcid \
[-d <provisioning date>] \
[-k <provisioning key>] \
[-o <filename>]
```

where:

`-d <provisioning date>` is a six-digit numeric date string in the form of YYMMDD. This option selects provisioning files with an activation date closest to the date specified without exceeding it.

`-k <provisioning key>` is an alphanumeric string of up to 6 characters, that specifies a key. The provisioning files selected by this option are, for each module, the most recent file matching this key.

The -k and -d options are mutually exclusive; if both are used, an error message is printed and the script terminates. If neither the -k nor the -d option is used, the file with the lexically highest file name is selected for each module.

`-o <filename>` specifies an output file. An output file is required when the user wants to write the script results to another filename. If an output file is not specified, the script results are written to file /opt/MagellanNMS/cfg/DMDRCnmid.cfg by default.

Exit codes

Exit codes for the DMDR server are shown in the following table.

Table 12
Exit codes for the DMDR server

Exit code	Description
1	Failed for a reason that is explained in the logs
51	Lack of memory
55	Bad command line arguments
59	IPC unit failed, DMDR is trying to register with IPC (interprocess communication) and it doesn't work.

Error messages

When an error occurs, messages are displayed in the System Log Display and written to the OAM log. The error messages for the DMDR server are shown in the following table:

Table 13
Error messages for the DMDR server

Error message	Meaning and action
DMDR -- Invalid reconnection delay length argument	Invalid specification of the reconnection delay length on the command line: DMDR prints help text for command line and exits.
DMDR -- Invalid command line argument	Invalid parameter on the command line: DMDR prints help text for command line and exits.
DMDR -- Missing mandatory argument	Missing mandatory parameter on the command line: DMDR prints its command line help message and exits.
DMDR -- Manager failed to connect to NCS Manager; error: <error code>	DMDR connection manager failed to connect to NCS Manager and exits.
DMDR -- Lost connection to NCS communication manager; exiting	DMDR lost connection to the NCS Manager and exits.
DMDR -- Handler failed to connect to NCS Manager; error: <error code>	A DMDR connection handler failed to connect to NCS Manager; connection to the associated OA will not be re-attempted.
DMDR -- Bad NCS message	No action on this NCS Manager internal error.
DMDR -- Failed to create DMDR network service; error: <error code>	DMDR failed to create the service endpoint for CCIFs and exits.
DMDR -- Bad CCIF message	The CCIF message with an invalid status is discarded.
DMDR -- Unexpected message from CCIF	DMDR is trying to connect to an OA and the communication manager receives an unexpected message; no action, the specific communication handler will also be notified of the problem and take the proper action.
(Sheet 1 of 6)	

Table 13 (Continued)
Error messages for the DMDR server

Error message	Meaning and action
DMDR -- OA connection failed: cannot send NCS message; OA: <OA name>	DMDR is trying to connect to an OA and cannot send the request to NCS Manager; connection to this OA will be retried later.
DMDR -- <diagnostic message>	NCS manager cannot establish connection to requested OA; connection to this OA will be retried later.
DMDR -- Lost connection to OA <OA name>	Connection to an OA was lost and will be retried later.
DMDR -- connected to OA <OA name>	DMDR has established communication with this OA.
DMDR -- incompatible CNMIDs, current: <cnmid> received from OA <OA name>: <cnmid>	DMDR capability ID and password are authenticated differently by this OA; DMDR takes no action but this discrepancy should be fixed.
DMDR -- Failed to connect to HGDS	DMDR exits because it cannot connect to HGDS to query for the names of the OAs it must connect to.
DMDR -- Lost HGDS connection	DMDR exits because it cannot ensure that it has received all the OA names.
DMDR -- Error response from HGDS	DMDR exits on the error response from the Host Group Directory Server.
DMDR -- HGDS error	DMDR exits on the internal error reported by the Host Group Directory Server.
DMDR -- Received no destination OA names from HGDS	The OA group used is empty; DMDR exits.
DMDR -- Failed to create DMDR authentication endpoint; error: <error code>	DMDR failed to create the endpoint for a client authentication; the authentication and the client registration fail and must be retried.
(Sheet 2 of 6)	

Table 13 (Continued)
Error messages for the DMDR server

Error message	Meaning and action
DMDR -- Authentication failed: no OA connection available	DMDR is not currently connected to any OA and cannot authenticate a client; the authentication and the client registration fail and must be retried.
DMDR -- Authentication failed: cannot send NCS message	DMDR cannot send an authentication request; the authentication and the client registration fail and must be retried.
NCS message - length too small <length>	Rejected incoming message is too small to be either an alarm or a status record.
NCS alarm record - oversize alarm received	Rejected incoming alarm is larger that the maximum size.
NCS alarm record - bad component ID	Rejected incoming alarm has an invalid component ID.
NCS alarm record - bad FAULT CODE	Rejected incoming alarm has an invalid fault code.
NCS alarm record - bad event type	Rejected incoming alarm has an invalid event type.
NCS alarm record - bad severity	Rejected incoming alarm has an invalid severity.
NCS status record - length too small	Rejected incoming status record is too small.
NCS status record - type <record type> : missing device mnemonic	The device mnemonic is missing from an incoming status record; rejected.
NCS status record - not enough space for record;	A status record is only partially contained in an incoming gross status record; rejected.
NCS status record - unsupported category	Rejected incoming status record has an invalid category.
NCS status record - <record type>: bad record length	Rejected incoming status record has an invalid length.
(Sheet 3 of 6)	

Table 13 (Continued)
Error messages for the DMDR server

Error message	Meaning and action
DMDR -- cannot open file <file name>	DMDR failed to open the alarm exceptions specification file; the alarm exceptions database will be left empty.
DMDR -- Bad character in fault code from excep file: <code>	The fault code read contains an invalid character; exception specification rejected.
DMDR -- Action code not allowed for MSG alarm: <line contents>	The alarm exceptions specification file contains this invalid line: line ignored.
DMDR -- Action code not allowed for CLR alarm: <line contents>	The alarm exceptions specification file contains this invalid line: line ignored.
DMDR -- Action code not allowed for SET alarm: <line contents>	The alarm exceptions specification file contains this invalid line: line ignored.
DMDR -- Illegal alarm type: <line contents>	The alarm exceptions specification file contains this invalid line: line ignored.
DMDR -- '*' embedded in fault code: <line contents>	The alarm exceptions specification file contains this invalid line: line ignored.
DMDR -- Bad character in fault code from alarm: <character>	An incoming alarm contains an invalid character; should not happen in the alarm exception manager since alarm fault code are validated when the alarm is received.
DMDR -- DBNL activation action on non-link component	An alarm for a component that is not a link has a fault code that the alarm exceptions database associates with the DBNL activation action: the alarm is handled as an ordinary CLR alarm.
DMDR -- DBNL deactivation action on non-link component	An alarm for a component that is not a link has a fault code that the alarm exceptions database associates with the DBNL deactivation action: the alarm is handled as an ordinary CLR alarm.
(Sheet 4 of 6)	

Table 13 (Continued)
Error messages for the DMDR server

Error message	Meaning and action
DMDR -- UNDEFINE alarm on a non-module component	An alarm for a component that is not a module has a fault code that the alarm exceptions database associates with the NCS UNDEFINE action: the alarm is handled as an ordinary CLR alarm.
DMDR -- Invalid status record type argument.	Invalid specification of a status record type to store; ignored.
DMDR -- Options -A and -B are incompatible.	Incompatible options are set by the command line; -a is ignored.
DMDR -- Handler failed to connect to NCS Manager; error: <error code>	A DMDR connection handler failed to connect to NCS Manager; connection to the associated OA will not be re-attempted.
DMDR -- cannot send connection request to OA <OA name>	NCS manager cannot establish connection to requested OA; connection to this OA will be retried later.
NCS alarm record - bad component ID <compid> FAULT_CODE = <fault code>	Rejected incoming alarm has an invalid component ID.
NCS alarm record - bad event type <type> FAULT_CODE = <fault code>	Rejected incoming alarm has an invalid event type.
NCS alarm record - bad FAULT CODE <fault code>	Rejected incoming alarm has an invalid fault code.
NCS alarm record - bad severity <sev> FAULT_CODE = <fault code>	Rejected incoming alarm has an invalid severity.
NCS alarm record - oversize alarm received <size> FAULT_CODE = <fault code>	Rejected incoming alarm is larger that the maximum size.
NCS message - length too small <length>	Rejected incoming message is too small to be either an alarm or a status record.
(Sheet 5 of 6)	

Table 13 (Continued)
Error messages for the DMDR server

Error message	Meaning and action
NCS status record - [NETLINK]: old record length <length>	Rejected incoming network link status record has a pre-G33 length.
NCS status record - [TRUNK]: old record length <length>	Rejected incoming trunk status record has a pre-G33 length.
DMDR -- bad CNMID in file <file name> -- <cnmid>	Invalid CNMID value in a line in the CNMID file; line ignored.
DMDR -- invalid COMP ID in file <file name> -- <compid>	Invalid component id in a line in the CNMID file; line ignored.
DMDR -- invalid line in file <file name> -- <line>	Invalid line in the CNMID file; line ignored.
DMDR -- Fault Code too long: <code>	The alarm exceptions specification file contains a line with an invalid fault code; line ignored.
(Sheet 6 of 6)	

Chapter 8
DPN NCS Communications Manager (NCSMGR)

This section contains information on the DPN NCS Communications Manager (NCSMGR). See the following topics for more information:

About the NCSMGR

In networks that only contain Passport nodes, this server is not used and should be turned off.

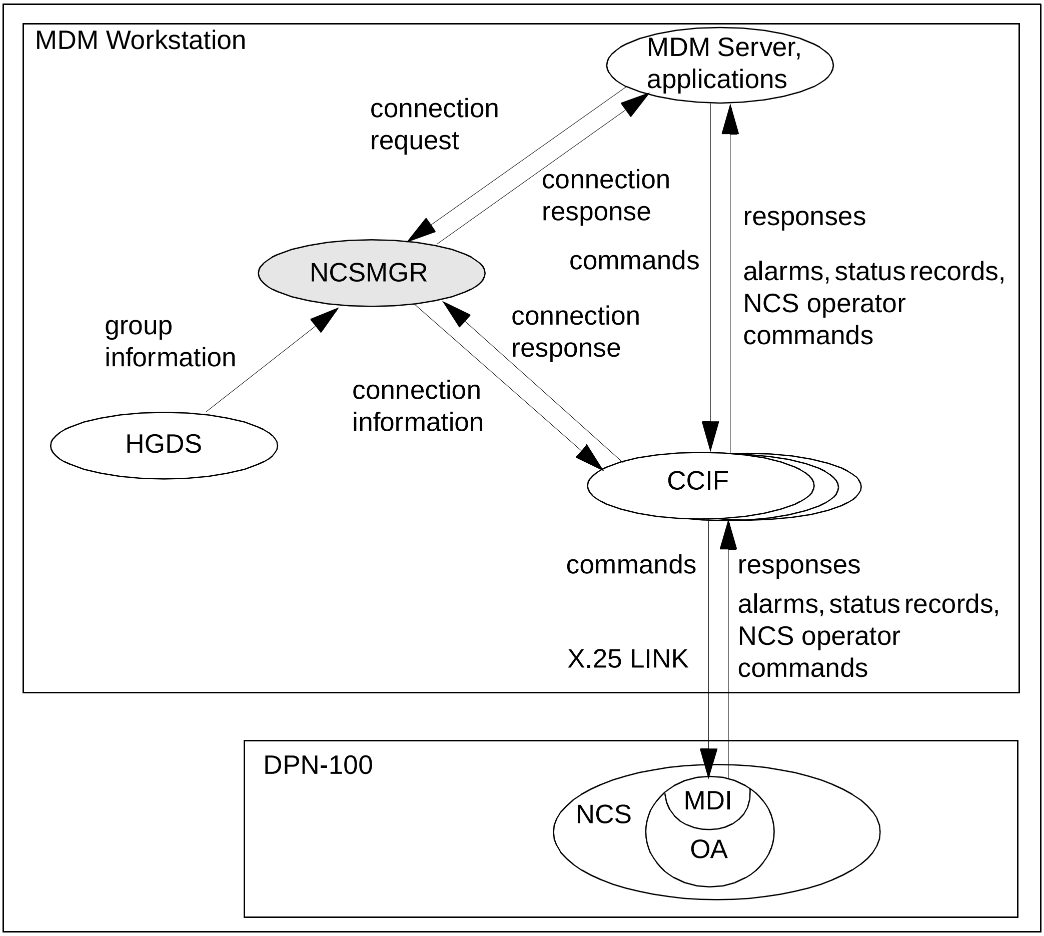
The NCSMGR server is responsible for setting up and maintaining connections to Workstation Management Data Interfaces (WS-MDI) on the OAs on DPN switches that run the Network Control System (NCS).

The NCSMGR server creates the communication channel interface (CCIF) process that is used to manage a virtual circuit. The CCIF process maintains the connection to NCS established over the virtual circuit on the X.25 link to the DPN network.

At startup, the NCSMGR server receives connection requests from other Preside Multiservice Data Manager s (MDM) servers and applications, to set up virtual circuits to specific NCS OAs. The NCSMGR server creates a CCIF process for each request. The CCIF process sets up the virtual circuit on the X.25 link to the network. The CCIF process returns the connection response to the NCSMGR server. The NCSMGR server forwards this data to the client.

At runtime, MDM servers and applications communicate directly with the CCIF process for direct data transfer.

Figure 9
NCSMGR data flow diagram



Managing the NCSMGR server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

For more information, see the following:

- “Suggested name in Server Administration” (page 100)
- “Startup command” (page 100)

Suggested name in Server Administration

The recommended name for the DPN NCS Communications Manager is DPN NCS Comms Mgr.

Configuring NCSMGR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the NCSMGR server has the following syntax:

```
/opt/MagellanNMS/bin/ncsmgr [-prefix <prefix_string>]
```

where:

`-prefix` specifies a prefix of up to eight characters that NCSMGR/CCIF uses to identify the MDI client connection. If you do not use the -prefix option, Preside Multiservice Data Manager identifies itself uniquely to an NCS OA when creating new MDI connections by using a string built from the first eight characters of the host name, plus the link and channel numbers. However in some cases the host name is not unique for the first eight characters and this can cause the NCS connection to fail. The -prefix command allows you to substitute a unique string in place of the host name.

This command starts the NCSMGR server with the following hard defaults:

- Maximum number of CCIFs allowed (0 .. 90). Default: 90
- Maximum number of dest table entries (0 .. 256). Default: 100
- CCIF executable name to exec. Default: /opt/MagellanNMS/bin/ncsccif
- Transaction pseudo-end period (1 .. 60). Default: 3
- Maximum transaction number (8 .. 256). Default: 64

To run the NCSMGR server with different default values, the following startup command should be used:

```
/opt/MagellanNMS/bin/ncsmgr
/opt/MagellanNMS/cfg/NCSCom.cfg
```

See “Configuration” (page 101) for details on file /opt/MagellanNMS/cfg/NCSCom.cfg.

Interdependencies

The NCSMGR server relies on the HGDS server.

Configuration

The following files are involved in configuring the NCSMGR server:

- /opt/MagellanNMS/cfg/HGDS.cfg

  contains a list of the groups of DPN OAs, the members in each group, and the communications parameters required to access the NCS in each DPN OA. See “HGDS server information file” (page 179).

  Note: This file also contains information about Passport groups. However, the NCSMGR server only uses fields that apply to groups of DPN OAs.

- /opt/MagellanNMS/cfg/NCSLink.cfg

  contains service parameters used for setting up SVCs over X.25 links to the network. See “Configuring file /opt/MagellanNMS/cfg/NCSLink.cfg” (page 102).

- /opt/MagellanNMS/cfg/NCSCom.cfg

  contains runtime parameters used by NCSMGR server. The parameter values in this file override the hard default values used by NCSMGR server. This file is only activated when the file name is included in the startup command. See “Configuring file NCSCom.cfg” (page 103).

Configuring file /opt/MagellanNMS/cfg/NCSLink.cfg

File /opt/MagellanNMS/cfg/NCSLink.cfg contains service parameters used for setting up SVCs over X.25 links to the network. The contents of this file requires modification under either of the following conditions:

- The first X.25 link that Preside Multiservice Data Manager (MDM) software begins searching for to set up an SVC connection for network access is anything other than link 1. The default version of the NCSLink.cfg file provided with MDM software sets the search startup link number to link 1.

  Link numbers are assigned to ports when setting up interfaces with Sun's X.25 Administration tool. To view the current link-to-port assignments, log in as root, enter /opt/SUNWconn/bin/x25tool & to launch the X.25 Administration tool, then from the main window, select Create/Modify with the Links menu button.

- More than two X.25 links have been configured for network access on the workstation. The default version of this file provided with the software searches a maximum of 2 links for an X.25 link on which to set up an SVC. There is no conflict if there are 2 links or less, only if there are more than 2.

Obey the following rules when configuring this file:

- Never insert information about a Frame Relay link into this file.
- You must enter the parameters in the order in which they are shown in "File format" (page 102). Start the entry and separate each field with a colon (:).
- When making modifications, do not modify the file provided with MDM software. Instead copy the file to directory /opt/MagellanNMS/cfg then modify the copied file. When searching for the file, the software first looks in /opt/MagellanNMS/cfg and uses the file, if it finds it. It the software does not find the file, it then looks in directory /opt/MagellanNMS//lib/cfg.

File format

File /opt/MagellanNMS/cfg/NCSLink.cfg consists of one statement in the form:

```
:p1:p2:
```

where:

`p1` is the logical link number to use first for X.25 VC setup. You assigned link numbers when setting up ports and interfaces with Sun’s X.25 Administration tool. The link number that you specify must correspond to a port for a low-speed interface (ZSH0, ZSH1) or a high-speed interface (HIH0, HIH1...) and it must be an X.25 interface; it cannot be an Ethernet interface or a Frame Relay Interface.

`p2` is the maximum number of X.25 links available. When attempting to set up a virtual circuit, the software tries to set up an SVC on one of the available x25 links in a round-robin fashion starting with the link specified by P1.

Example

The NCSLink.cfg file supplied with Preside Multiservice Data Manager software /opt/MagellanNMS/lib/cfg/NCSLink.cfg has the following statement in it:

```
:1:2:
```

This statement causes MDM software to start at link 1 when attempting to set up an SVC on an X.25 connection to the network, and will search a maximum of two links in round-robin fashion to try and set up an SVC.

Configuring file NCSCom.cfg

File /opt/MagellanNMS/cfg/NCSCom.cfg contains key parameters needed to allow the Preside Multiservice Data Manager workstation to communicate with NCS.

> ***Note:*** For new installations that only have one X.25 link to the DPN network, it should not be necessary to add an entry to file NCSCom.cfg. This file is designed to contain a single entry that consists of the five following parameters:
>
> *:<p1>:<p2>:<p3>:<:p4>:<p5>:*
>
> If the file does not contain an entry, the software pretends that there is an entry with the following values:

<p1> = 90, <p2> = 64, <p3> = /opt/MagellanNMS/bin/ncsccif, <p4> = 3, <p5> = 64

For new installations that only have one X.25 link to the DPN network, these default values should be sufficient. After installation is complete and the network expands, it may become necessary to modify the values in this file.

Adding parameters to the NCSCom.cfg file

The format is:

```
:p1:p2:p3:p4:p5:
```

where:

`p1` is the maximum number of CCIFs allowed. A communications channel interface (CCIF) is a software module that manages a single communications channel to NCS. A CCIF is required for each active virtual circuit. Set this value to a number greater than or equal to the maximum number of simultaneous virtual circuits required by the DPN Network Operator toolset from this integrated workstation to NCS. A value of 0 means no virtual circuits may be set up to NCS.

`p2` is the maximum number of destination table entries. The destination table contains an entry for each DPN operation center that is to be accessed by this Preside Multiservice Data Manager workstation. Set the value to a number greater than or equal to the maximum number of operations agent entries that will be placed in file /opt/MagellanNMS/cfg/HGDS.cfg.

`p3` is the CCIF filename. This name may be changed at software installation to agree with the UNIX directory and pathname where the CCIF module is installed.

`p4` is the transaction-quiet flush period. This is the time in seconds before a transaction is flushed. Some commands, such as the NM TEST command, give a periodic response. All of the data for a particular response may arrive at the workstation, but are not necessarily be visible to the user since the transaction is not ended. This timer indicates that if no new data has arrived in X seconds, the data which has already arrived is to be displayed on the screen.

`p5` is the maximum transaction number. This value should be set greater than or equal to the maximum number of simultaneous transactions which may be opened to NCS.

You can add your own comments at the end of the entry. Type # followed by the comment text.

> ***Note:*** The order in which parameters are entered into fields is critical. The entry begins and ends with a colon (:) and each field in the entry is separated from the next field by a colon. You can specify a default value by entering two colons (::), which is equivalent to leaving a field empty. For example:
>
> *:4:100::5:40:*
>
> is the same as:
>
> *:4:100:/opt/MagellanNMS/bin/ncsccif:5:40:*

Exit codes

Exit codes for the NCSMGR server are shown in the following table.

Table 14
Exit codes for the NCSMGR server

Exit code	Description
1	The server exits under the following conditions: - out of memory - failed to use IPC service - invalid command line arguments - invalid configuration parameter - given CCIF (ncsccif) file not executable - destination table full - invalid or duplicate destination definition entry
(Sheet 1 of 2)	

Table 14 (Continued)
Exit codes for the NCSMGR server

Exit code	Description
35	Failed to run ncsccif
62	Terminated due to licensing problem (see MDM Logs). Look at the System Log Display tool for logs about licensing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.
(Sheet 2 of 2)	

Error messages

Error messages for the NCSMGR server are shown in the following table

Table 15
Error messages for the NCSMGR server

Error message	Meaning and action
NCSMGR - Unable to establish a licensing context	The NCSMGR server licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
NCSMGR - License request failed : <reason>	A run-time license cannot be allocated to the NCSMGR server for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running NCSMGR.
NCSMGR - License warning: <reason>	The license required to run NCSMGR is about to expire. Contact Nortel Networks immediately.

Chapter 9
DPN PM File Access Server (PFAS)

This section contains information on the DPN PM File Access Server (PFAS). See the following topics for more information:

About the PFAS server

In networks that only contain Passport nodes, this server is not used and should not be started.

The PFAS performs the following main functions:

- manages uploading and downloading of files between the Preside Multiservice Data Manager workstation and DPN packet modules (PMs)
- manages backup and restoration of MCFs from a backup directory on a disk on the workstation
- establishes X.25 VC connections to the PA ICON of a packet module to manage uploads and/or downloads of files

There are two versions of the PFAS server:

- a standard PFAS server, which is used for retrieving and modifying service data files on DPN modules
- a software download server; a PFAS server, which is set up to download software to DPN modules

Managing the PFAS server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See in 241-6001-303 *Preside MDM Administrator Guide* for more information for instructions to use the Server Administration tool.

For more information, see the following:

- "Suggested name in Server Administration" (page 108)
- "Startup command" (page 108)

Suggested name in Server Administration

The recommended name for the standard PFAS server is DPN PFAS. The recommended name for the software download server is DPN PFAS SW Download.

Configuring PFAS with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the PFAS server has the following syntax:

```
/opt/MagellanNMS/bin/pfas
```

Note: If you are going to be performing software downloading, start a second instance of the PFAS server using the following command:

```
/opt/MagellanNMS/bin/pfas -n swdld
```

Configuration

Configuring the PFAS server involves provisioning a PA ICON on the DPN-100 switch to which the PFAS server communicates and setting parameters in file /opt/MagellanNMS/cfg/PFA.cfg. For the instructions to perform these configuration tasks, see the section on provisioning file access server customization in 241-6001-304 *Preside MDM Configuration Management for DPN Administration.*

Interdependencies

The PFAS server relies on the NCSMGR server. The PFAS server must be running on the same workstation as the NCSMGR server.

Exit codes

When a server fails for any of the reasons listed, the exit code is displayed in the message area of the Server Administration tool. The reason for the failure is displayed in the System Log Display and logged in the OAM log. Exit codes for the PFAS server are shown in the following table.

Table 16
Exit codes for the PFAS server

Exit code	Description
50	Failed to register with MNSD, failed to register service name. This exit code notifies the server process starting PFAS to stop automatic restart of PFAS.
62	Terminated due to licencing problem (see MDM Logs). Look at the System Log Display tool for logs about licencing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.

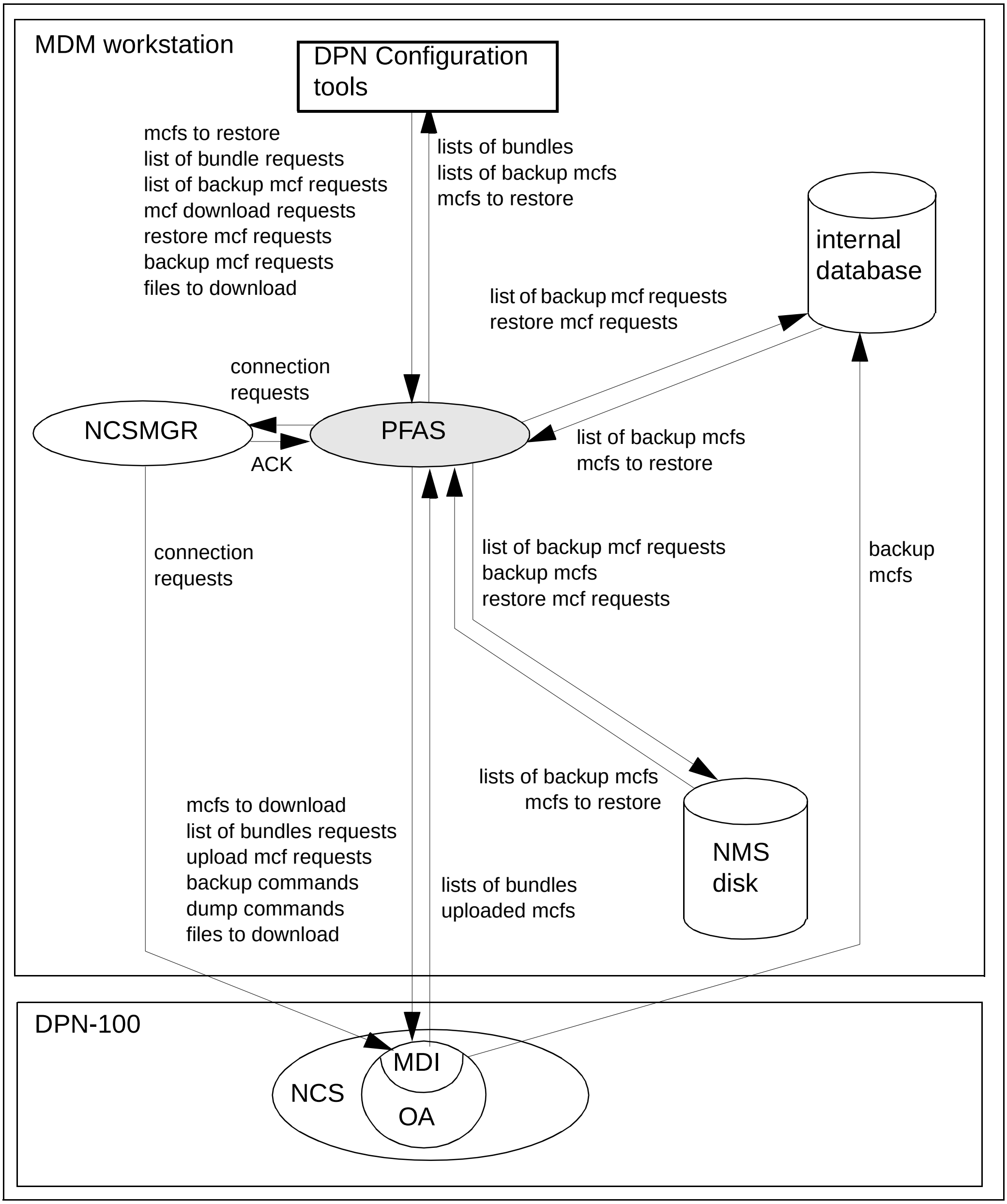
Error messages

Error messages are generated if the PFAS server fails to obtain a Preside Multiservice Data Manager license. Error messages for the PFAS server are shown in the following table.

Table 17
Error messages for the PFAS server

Error message	Meaning and action
PFAS - Unable to establish a licensing context	The PFAS server licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
PFAS - License request failed : <reason>	A run-time license cannot be allocated to the PFAS server for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running PFAS
PFAS- License warning: <reason>	The license required to run PFAS is about to expire. Contact Nortel Networks immediately.

Figure 10
PFAS data flow diagram



Chapter 10
Data Manager Agent (DMA)

This section contains information on the Data Manager Agent (DMA). See the following topics for more information:

- “About the Data Manager Agent” (page 113)
- “Configuration” (page 121)
- “Managing the DMA server” (page 127)
- “Interdependencies” (page 129)
- “Exit codes” (page 129)
- “Error messages” (page 130)

About the Data Manager Agent

Depending on the startup parameters you use when configuring the server, the DMA server can be used to perform the following functions:

- “Workstation surveillance using NCS status probing” (page 113)
- “Global alarm clearing for DPN” (page 116)
- “Global alarm clearing for Passport” (page 119)

Workstation surveillance using NCS status probing

For this function, the DMA server sends the status probing interval to the NCS in the call-setup packets that the DMA server uses to set up the SVC to the NCS over the X.25 link. At a scheduled interval, the NCS polls the workstation. If the NCS does not receive a reply from the workstation, it raises an alarm, which it distributes throughout the NCS.

Workstation surveillance using NCS status probing applies to DPN-only and DPN/Passport networks.

The probing sequence takes place as follows:

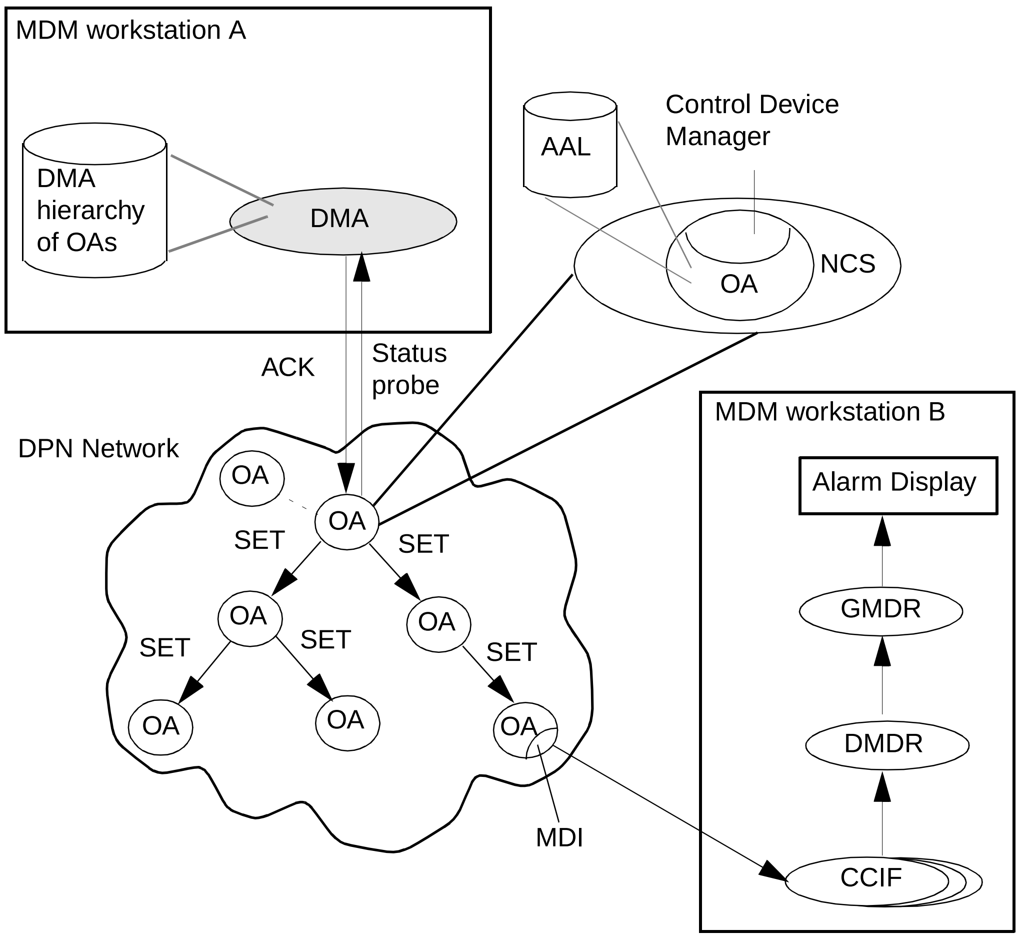
1 The NCS issues a status probe to the Preside Multiservice Data Manager (MDM) workstation periodically, depending upon what the probing interval is set to. The probing interval is set in the startup command parameters of the DMA server. For information on setting the probing interval, see “Startup command” (page 128).

2 The DMA server replies to each NCS status probe with an acknowledgment (ACK) to the OA. The NCS generates an alarm if there is no reply before the next status probe is issued.

3 The alarm is routed through the NCS hierarchy, to ensure that all workstations connecting to the OAs on the routing hierarchy will receive the alarm.

4 The workstation alarm can then be displayed by MDM surveillance tools, if required by the user. In the example shown in the figure “DMA data flow diagram for workstation surveillance using NCS status probing” (page 115), the Alarm Display in workstation B displays the workstation alarm.

Figure 11
DMA data flow diagram for workstation surveillance using NCS status probing



Global alarm clearing for DPN

Global alarm clearing applies to DPN-only or DPN/Passport networks.

For this function, you enter a global alarm clear request from the Alarm Display and the alarm is cleared throughout the Network Control System. Global alarm clearing takes place as follows:

1. The user issues a global alarm clear request. In the example shown in the figure “DMA data flow diagram for DPN global alarm clearing” (page 118), the user at workstation A issues a global alarm clear request from the Alarm Display.
2. The global alarm clear request is forwarded to the GMDR server in workstation A.
3. The global alarm clear request is forwarded to other GMDR servers in the GMDR hierarchy. In the example shown in the figure “DMA data flow diagram for DPN global alarm clearing” (page 118), the global alarm clear request is forwarded to the GMDR server in workstation B. The GMDR server forwards the request to the DMA server.
4. The DMA server determines the hierarchy of OAs from its internal database.
5. The DMA server sends the global alarm clear request through the top level OA in the managed region, to the OAs in the lowest layer of the OA hierarchy. The connection from the workstation is a Management Data Interface (MDI) on the top level OA for the managed region.
6. Some of the OAs have an Active Alarm Database (AAL), depending on the configuration of the OAs.
7. The OAs on the lowest layer of the DPN network remove alarms from their AALs, if they have an AAL. They also send an ACK back to the DMA server and a CLR to the next OA up the hierarchy.
8. The ACK informs the DMA server that the OA found an alarm in the AAL, and the CLR tells the next OA up the hierarchy to remove the alarm from its AAL.
9. If the DMA server receives an ACK from at least one bottom layer OA, it assumes that the global alarm clearing function is successful.

   On receiving a CLR from a lower OA, an OA clears the alarm in its own AAL if it has an AAL, and sends a CLR to the next OA up the hierarchy.

10 One or more of the following fault conditions may occur:

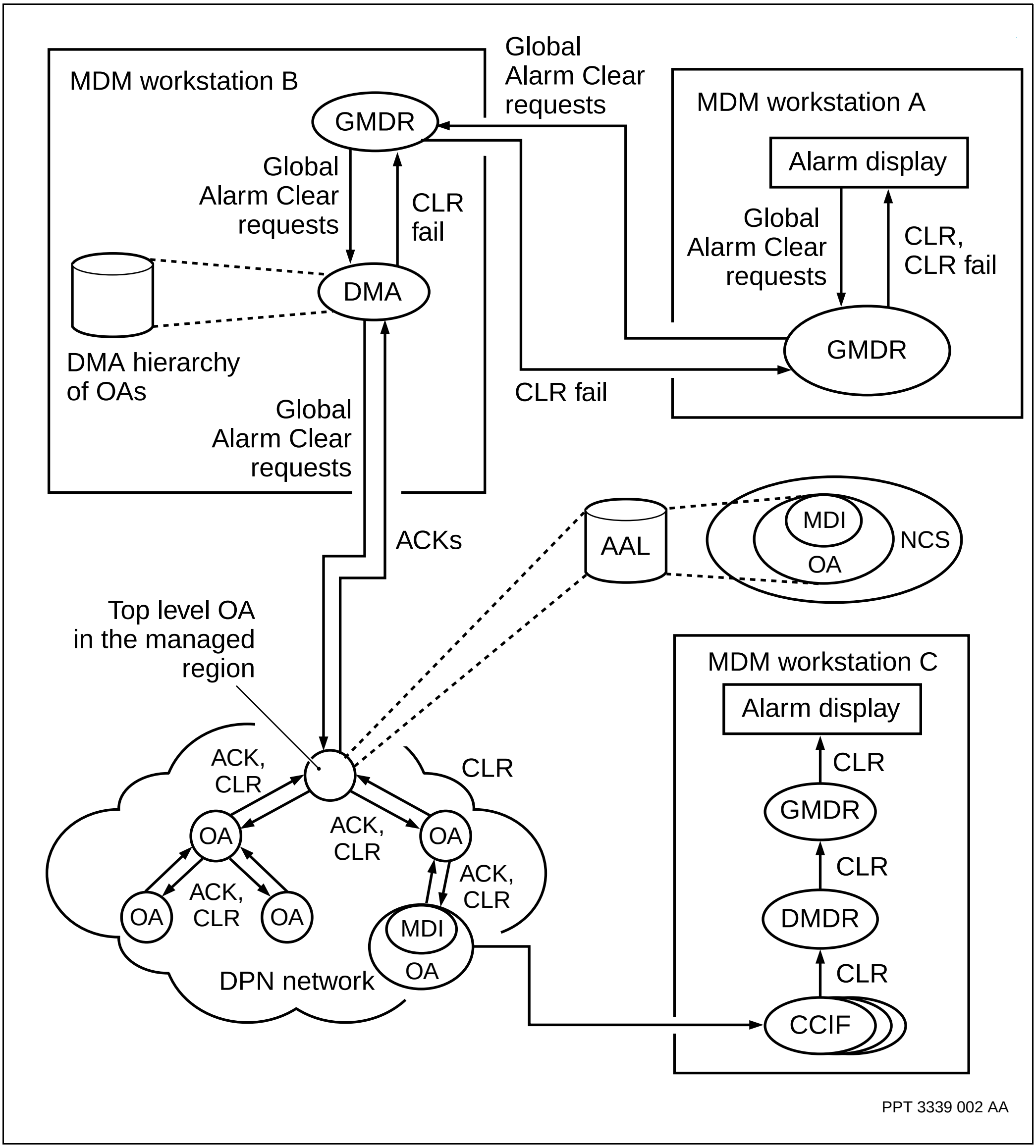
If the DMA server does not receive an ACK from any lowest level OA, then step 3 through step 7 are performed for the next-lowest level of OAs. If the DMA server receives an ACK from at least one OA, the DMA server assumes that the global alarm clearing function is successful.

Subsequently, if the DMA server does not receive an ACK from any next-lowest level OA, then step 3 through step 7 are performed for the layer of OAs above the next-lowest level, if there is such layer. If the DMA server receives an ACK from at least one OA, the DMA server assumes that the global alarm clearing function is successful.

If the DMA server does not receive an ACK from any OA in any layer, the DMA server assumes that the global alarm clearing function is not successful. Through the GMDR servers, and through the Alarm Display, the DMA server notifies the user that global alarm clearing was not successful.

Information to identify the hierarchy of OAs is stored in the DMA database. If the user changes the OA structure, the DMA server must be restarted to rediscover the hierarchy of OAs where the alarms are removed from the AAL.

Figure 12
DMA data flow diagram for DPN global alarm clearing



Global alarm clearing for Passport

Alarms can be globally cleared in a Passport network. To clear alarms, a global alarm clear request is made from the Alarm Display or the Component Information Viewer and is cleared throughout the network. Global alarm clearing takes place as follows:

1 The user issues a global alarm clear request. In the example shown in “DMA data flow diagram for Passport global alarm clearing” (page 120), the user at workstation A issues a global alarm request from the Alarm Display (or Component Information viewer or ManClear macro).

2 The global alarm clear request is forwarded to the GMDR server in workstation A.

3 The global alarm clear request is forwarded to other GMDR servers in the GMDR hierarchy and DMA servers. In the example shown in “DMA data flow diagram for Passport global alarm clearing” (page 120), the global alarm clear request is forwarded to the GMDR server in workstation B. The GMDR server forwards the request to the DMA server.

4 The DMA server authenticates with the group specified through its configuration file.

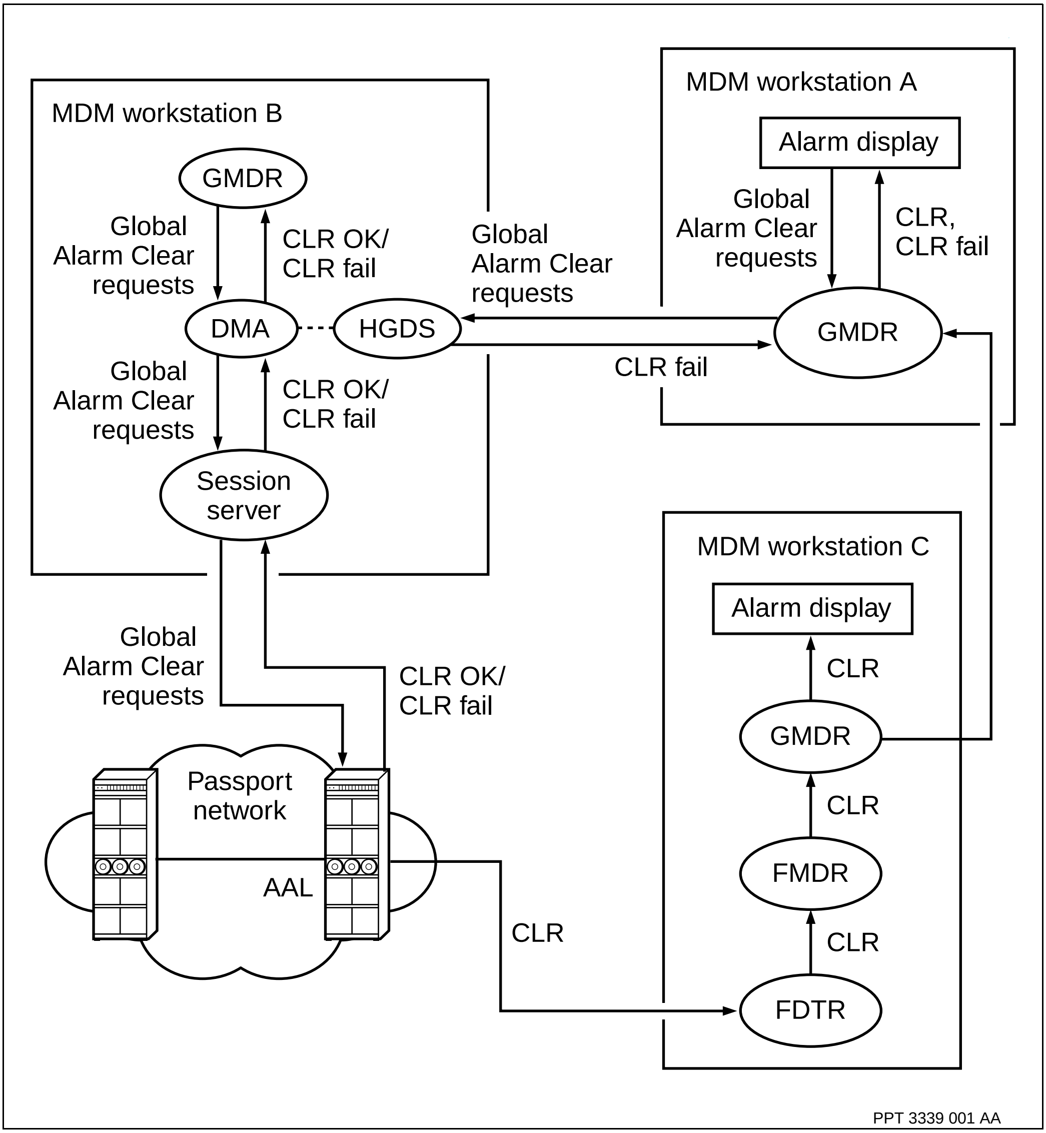
5 The DMA server sends the global alarm clear request to the targeted node. The connection from the workstation to the node uses a FMIP session. The request is encoded in ASCII over FMIP.

6 The targeted node removes the alarm from its AAL, if it has an AAL set up. The node sends back an OK response in the case of success and a CLR alarm is issued on all NMIS sessions, which include FMIP sessions.

7 In the case of a failure, through the GMDR servers and through the top level application (see “DMA data flow diagram for Passport global alarm clearing” (page 120)), the DMA server notifies the user that global alarm clearing was not successful with specific reasons.

8 If successful, the CLR alarm comes in the system through the FDTR/FMDR connected to that node. The alarm is cleared out of FMDR(s) database, and then it is forwarded to all GMDR(s) and cleared out of GMDR(s) databases and reflected in the top level applications.

Figure 13
DMA data flow diagram for Passport global alarm clearing



Configuration

This section describes the configuration files associated with the DMA server. These files are as follows:

- “File /opt/MagellanNMS/cfg/DmaOA.cfg” (page 122)

  This file is used for workstation surveillance using NCS status probing.

- “File /opt/MagellanNMS/cfg/DmaClrOA.cfg” (page 123)

  This file is used for DPN global alarm clearing.

- “File /opt/MagellanNMS/cfg/DmaClrPP.cfg” (page 126)

  This file is used for Passport global alarm clearing.

Configuring these files is insufficient to allow the DMA server to perform all of its functions. Additional configuration tasks need to be performed. Also, the DMA server supports more functions than just those associated with these configuration files. For procedures to set up the full set of functions associated with the DMA server, see the following:

- Preside Multiservice Data Manager workstation surveillance using NCS status probing

  To set up server alarm distribution through NCS and workstation surveillance using NCS status probing, see 241-6001-303 *Preside MDM Administrator Guide*.

- Global alarm clearing for DPN and Passport

  To set up global alarm clearing, see 241-6001-303 *Preside MDM Administrator Guide*.

File /opt/MagellanNMS/cfg/DmaOA.cfg

File /opt/MagellanNMS/cfg/DmaOA.cfg is used for workstation surveillance using NCS status probing.

More than just configuring this file is required to set up workstation surveillance using NCS status probing. Configuring these functions includes the following main steps:

- editing file /opt/MagellanNMS/cfg/DmaOA.cfg to add the information that the DMA server needs to connect to the NCS
- starting the DMA server with arguments in its startup command to have the NCS probe the workstation.

For the instructions to configure these functions, see the section on setting up workstation surveillance using NCS status probing in 241-6001-303 *Preside MDM Administrator Guide*.

File DmaOA.cfg contains key parameters needed to allow the DMA server to communicate with the Control Device Manager in the NCS. Specifically, the parameters permit a virtual circuit (VC) to be established between the Preside Multiservice Data Manager workstation and the Control Device Manager.

File /opt/MagellanNMS/cfg/DmaOA.cfg contains only one entry. Additional entries are ignored.

The format for an entry in this file is as follows:

```
:DDD ... D:OO ...O:AAA ... A:CC:PPP:X:R:RPOA:
```

where:

`D` is the Destination mnemonic. Maximum 12 characters

`O` is the mnemonic of the NCS OA (OA name) containing the destination’s Control Device Manager. This mnemonic must match the information entered into the Name field of an OA Member that is defined in file /opt/MagellanNMS/cfg/HGDS.cfg. Maximum 12 characters.

`A` is the DNA of the Control Device Manager. Maximum 16 characters

> ***Note:*** This is not the same as the DNA of the MDI access DNA.

`C` is the CUG index of the Control Device Manager. Maximum 2 digits

`P` is the packet size on the VC. (Use 128, 256, or 512). Maximum 3 digits

`X` specifies whether the call is to be routed over X.75. Can be Y or N.
If N, then R and RPOA are ignored.

`R` specifies whether the calls are to be routed over the X.75 facilities of a Remote Private Operating Agency (RPOA). Can be Y or N. If N, the RPOA is ignored.

`RPOA` is the RPOA identifier code. 4 (BCD) digits.

> ***Note:*** The order in which parameters are entered into the fields is critical. The entry must begin and end with a colon (:) and each field in the entry must be separated from the next field by a colon. There are no default values for the parameters.

The following is an example of a file entry. For this example, there should also be an OA Member defined in file /opt/MagellanNMS/cfg/HGDS.cfg with a Name field containing the entry CORENCS.

```
:CORENCSIF:CORENCS:3021015008:01:512:N:
```

> ***Note:*** This example only contains six parameters because the call does not have to be routed over X.75.

File /opt/MagellanNMS/cfg/DmaClrOA.cfg

File /opt/MagellanNMS/cfg/DmaClrOA.cfg is used for DPN global alarm clearing.

More just than configuring this file is required to set up global alarm clearing for DPN. Configuring global alarm clearing involves the following main steps:

- adding an entry to file opt/MagellanNMS/cfg/DmaClrOA.cfg that provides Preside Multiservice Data Manager (MDM) software with the information required to log in to the top level OA in the region managed through this workstation.

  The NCS Capability ID and password used to log in must have access privileges that are sufficient to permit global alarm clearing.

- using the Server Administration tool to start the DMA server with the -c option
- using the GMDR Administration tool to set up the GMDR server to access the DMA server as one of is subservers

For the instructions to set up global alarm clearing for DPN, see 241-6001-303 *Preside MDM Administrator Guide*.

File DmaClrOA.cfg contains the configuration parameters required to establish a virtual circuit to the top level OA in the managed region for alarm clearing requests. This file should only contain one entry. Only the first entry is recognized, any others are ignored.

The format for an entry in this file is

```
:DDDDDDDDDDDD:IIIIIIIIIIII:PPPPPPPPPPPP:
```

where:

`D` is the OA Destination mnemonic. The OA Destination mnemonic corresponds to the OAMember field defined for the top level OA in the managed area as defined in file /opt/MagellanNMS/cfg/HGDS.cfg. In file HGDS.cfg, the OA Member contains the name of the Management Data Interface (MDI) on the OA. The MDM workstation connects to this OA to send global alarm clearing request messages to NCS. This mnemonic must match the OA Member for the top level OA entered in file /opt/MagellanNMS/cfg/HGDS.cfg. Maximum 12 characters. See also “The OA definition section” (page 181).

`I` is the NCS capability id. Maximum 12 characters. The id must have the following capability, level, and impact:

NAMS	Network	Service
	OA/Device	None
	Application/Line	None
Switching	Network	None
	Device	None
	Line	None

`P` is a password that has the NCS capability id. Maximum 12 characters.

The following is an example of a file entry. For this example, there should also be an OA Member named CORENCS in file /opt/MagellanNMS/cfg/HGDS.cfg.

```
:CORENCSIF:CORENCS:axylt:
```

File /opt/MagellanNMS/cfg/DmaClrPP.cfg

File /opt/MagellanNMS/cfg/DmaClrPP.cfg is used for global alarm clearing.

Setting up global alarm clearing for Passport networks involves configuring this file and following these main steps:

- adding one or more entries to file opt/MagellanNMS/cfg/DmaClrPP.cfg to provide the DMA server with the information required to connect to each group.

  It is highly recommended to add one group per FMDR configured. For each group of nodes monitored by a FMDR, that same group should be added to the DmaClrPP.cfg file.

  The group user ID and password used to log in must have access privileges that are sufficient to permit global alarm clearing. At a minimum, the user ID must have systemAdministration privilege, a customer ID of 0, and a scope of device or higher.

- using the Server Administration tool to start the DMA server with the -f option to allow global clear.

  Optionally, the command line option -t can be configured to customize the inactivity period timer, and the command line option -n can be used to disable automatic connection attempts to IMDR.

- using the GMDR Administration tool to set up the GMDR server to access the DMA server as one of is subservers.

  Typically, one DMA server should be configured along with the FDTM server.

For the instructions on how to set up global alarm clearing for Passport, see 241-6001-303 *Preside MDM Administrator Guide*.

The /opt/MagellanNMS/cfg/DmaClrPP.cfg contains the parameters to configure global alarm clearing. The format for entries to this file are in the following format:

```
:GroupName:UserID:Password:
```

`where:`

`GroupName` is the group name. The group name corresponds to the FGroup field definition with its included members as defined in file /opt/MagellanNMS/cfg/HGDS.cfg. The DMA server connects to all groups indicated in this file to send global alarm clearing request messages to the targeted node. Maximum 12 characters.

`UserId` is the group user ID. At a minimum, the user ID must have systemAdministration impact and scope of device or higher and a customer ID of 0. Maximum 8 characters**.**

`Password` is a password that corresponds to the user ID. Maximum 8 characters.

The following is an example of a file entry. For this example, there should also be a group entry named ALL in file /opt/MagellanNMS/cfg/HGDS.cfg.

```
:ALL:user:password:
```

As soon as a syntax error is found in the file, it is displayed in the Preside Multiservice Data Manager System Log Display and DMA exits.

Once the file is read by the DMA server, each password is removed and an encrypted one is added in the forth field. The above example would become:

```
:ALL:user::72eilRnWj7{s{A6hgg7:
```

Managing the DMA server

Use the Server Administration tool to enter or edit the startup command, and to start, stop, or set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide* for more information.

See the following information:

- “Suggested server name in Server Administration” (page 128)
- “Startup command” (page 128)

Suggested server name in Server Administration

The recommended name for the DMA server is DMA.

Configuring DMA with the Server Administration tool requires you enter to the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command syntax for the DMA server is as follows:

```
/opt/MagellanNMS/bin/dma
[-d [<filename>]] [-c [<filename>]] \
[-l <link number>] [-p <pr_time_int>] [-f]
[-t <interval>] [-a <interval>] [-n]
```

where:

`-n` disables automatic connection attempts to IMDR

`-f` enables global clearing. The file /opt/MagellanNMS/cfg/DmaClrPP.cfg must be present and filled.

`-t <interval>` This option is associated with global clearing. It indicates the period of idle time (in minutes) before DMA disconnects all opened connections. The default is 10 minutes. The possible range is [1-99]; otherwise, the interval is restored to default values.

`-a <interval>` This option is associated with global clearing. It indicates the waiting period (in minutes) before DMA re-attempts a connection to a group in case of failure. The default is 5 minutes. The possible range is[1-99]; otherwise, the interval is restored to default values.

`-d [<filename>]` specifies the name of a file that contains the parameters needed to establish a connection to an OA for workstation surveillance. If you specify the -d option without a file name, the default file /opt/MagellanNMS/cfg/DmaOA.cfg is used.

`-c [<filename>]` specifies the name of the file that contains the parameters needed establish a VC to the top level OA in the managed region for global alarm clearing. If you enter the -c option without a filename, the default file /opt/MagellanNMS/cfg/DmaClrOA.cfg is used.

`-l <link number>` is either 0 or 1. If you do not specify this parameter, the DMA server uses link 0. You can force the DMA server to use link 1 instead.

`-p <probing interval>` specifies that status probing is to be performed for workstation surveillance using NCS status probing. The <probing interval> is the interval in minutes at which NCS probes the workstation and it must be an integer with a minimum value of 1. If you do not specify the <probing interval> parameter, the default NCS status probe time interval of five minutes is used.

Interdependencies

On workstations that use the DNP global alarm clearing feature or that send their workstation status to the DPN network (workstation surveillance), NCSMGR servers are also required in addition to the DMA server. Otherwise, the GMDR server may be used as a clearing server by other GMDR servers.

On workstations that use the Passport global alarm clearing feature, HGDS and FDTM are also required in addition to the DMA server; otherwise, the GMDR server may be used as a clearing server by other GMDR servers.

Exit codes

Exit codes for the DMA server are shown in the following table.

Table 18
Exit codes for the DMA server

Exit code	Description
50	The server has exited because of one of the following conditions: invalid command line arguments no control Device Manager in the DmaOA.cfg file failed to use IPC service out of memory X.25 failure
53	Communication resource error
56	Bad configuration file.

Error messages

Error messages for the DMA server are shown in the following table:

Table 19
Error messages for the DMA server

Error message	Meaning and action
Problem with ipc_init	Fatal, could not initialize IPC system. Make sure MNSD is running by means of the Server Administration tool.
Problem with Server Object	Fatal, could not register the DMA service. Another DMA is probably running, verify the server configuration by means of the Server Administration tool.
Check DmaOA.cfg file if MDI information is provided	Fatal, error in /opt/MagellanNMS/cfg/DmaOA.cfg.
Failed to connect to IMDR for workstation server surveillance	Non-fatal, start IMDR server by means of the Server Administration tool.
Cannot communicate with the DMA server	The DMA server is not running and/or is not connected to GMDR.
(Sheet 1 of 3)	

Table 19 (Continued)
Error messages for the DMA server

Error message	Meaning and action
Alarm not found	The alarm has already been cleared.
Global Clear service not configured	The DMA server is running and connected to GMDR, but the command line option for Global Clear is not turned on or the DmaClrPP.cfg file is not present.
The DMA server has insufficient capabilities to clear alarms on EM/<NODE_NAME>	The User ID specified in the DMA configuration file has insufficient capabilities regarding Global clearing.
The DMA server was unable to authenticate with EM/<NODE_NAME>	Authentication failed for that particular node.
The DMA server's configuration does not allow communication with EM/<NODE_NAME>	This could happen for the following reasons: 1) authentication fails for the whole group for the User ID provided in the DMA configuration file (either the User ID is not defined on the switch or the password provided is incorrect) 2) the node is not part of any group defined in the DMA server configuration file 3) the FDTM server is down, or 4) the HGDS server is down. This message is sent as soon as there is a problem connecting to the group.
The DMA server is unable to reach Passport EM/<NODE_NAME>	The node is unreachable.
The DMA server cannot connect to the Passport Communications Manager Server (FDTM).	The FDTM server is not running.
The DMA server cannot connect to the Host Group Directory Services server (HGDS).	The HGDS server is down.
(Sheet 2 of 3)	

Table 19 (Continued)
Error messages for the DMA server

Error message	Meaning and action
The Active Alarm List feature is not activated on EM/<NODE_NAME>	The AAL feature has not been activated on the switch.
Internal error	An internal protocol error occurred or there are cross versions of GMDR-DMA servers.
(Sheet 3 of 3)	

Chapter 11
Data Synchronization Server (DATASYNCSERVER)

This section contains information about the Data Synchronization Server (DATASYNCSERVER). See the following topics for more information:

About the DATASYNCSERVER

The DATASYNCSERVER ensures that the administration database is in synchronization with the information on the network element. In addition, it supports the node's Backup and Restore and Data Synchronization Administration tools. The Database Synchronization Controller resides on the DATASYNCSERVER and receives information from the Backup Server when a backup has occurred. The following steps occur when a backup or restore has occurred.

1 The Backup Server initiates a backup request and sends a probe to the nodes to collect a view.

2 When the Backup Server finishes retrieving a new view and journal log files from the nodes, and storing them to local disk, it notifies the Database Synchronization Controller.

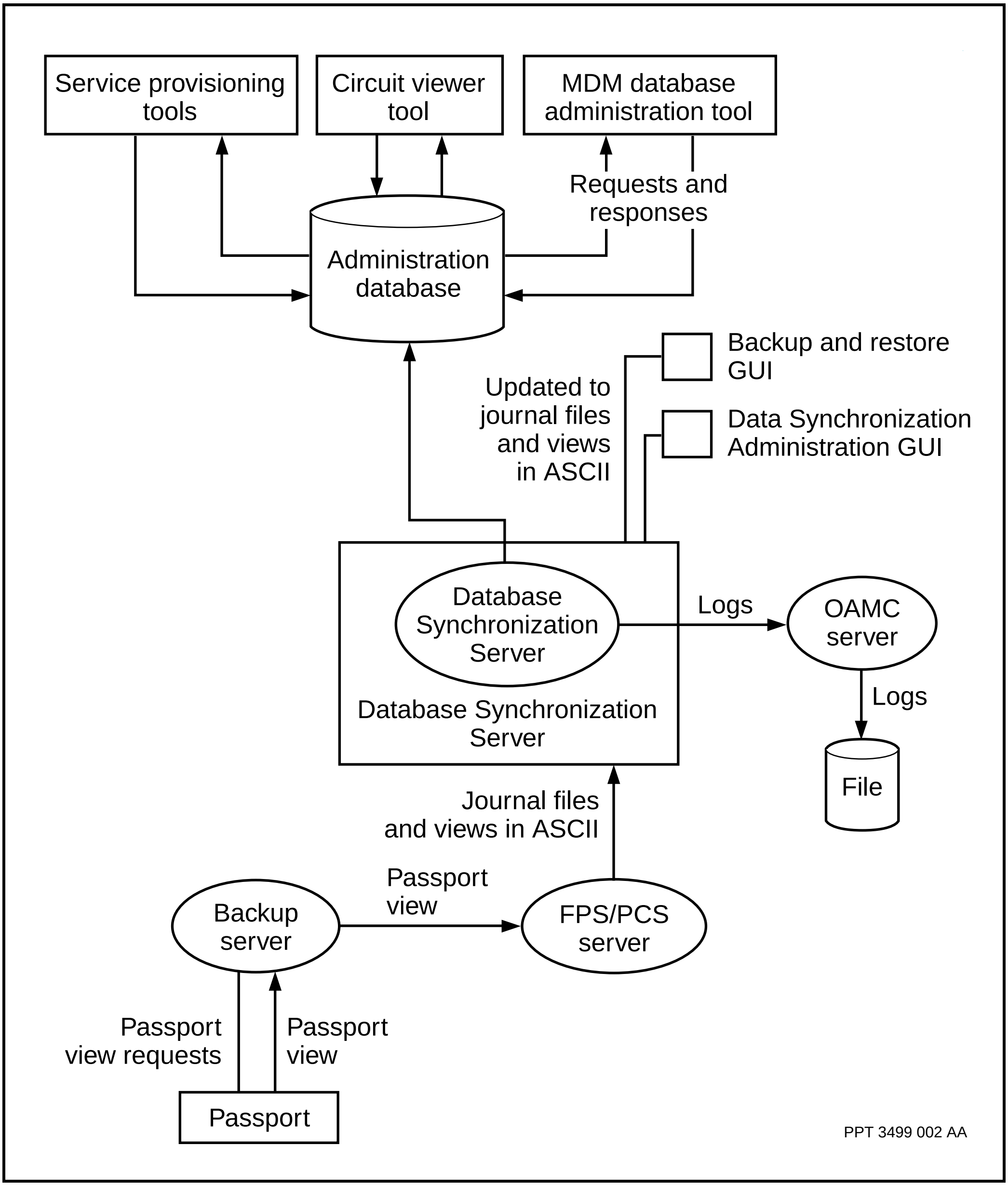
3 The Database Synchronization Controller triggers the FPS and PCS servers to retrieve the view from a mount point specified by the Backup Server, and translates the view information into an ASCII form compatible with what is in the administration database.

4 The Database Synchronization Controller compares the journal file and view information stored in the database, against what was collected by the backup server and updates the information in the database accordingly.

5 Users of the tools access the contents of the database to perform provisioning and circuit management operations.

The following diagram illustrates the servers associated with the administration database.

Figure 14
Data flow diagram for Data Synchronization Server


Service provisioning tools
Circuit viewer tool
MDM database administration tool
Requests and responses
Administration database
Backup and restore GUI
Data Synchronization Administration GUI
Updated to journal files and views in ASCII
Database Synchronization Server
Logs
OAMC server
Logs
Database Synchronization Server
File
Journal files and views in ASCII
Passport view
Backup server
FPS/PCS server
Passport view requests
Passport view
Passport
PPT 3499 002 AA

Database Synchronization Controller Logging

The OAMC server collects logs from all systems in the Database Synchronization server. These logs are also sent directly to a log file. These logs are stored in one file called /opt/MagellanNMS/data/log/DataSync/dataSyncServer.dlog. For more information on logging, see the 241-6001-400 *Preside MDM Administration Database User Guide*.

Configuring the DATASYNCSERVER

Use the Server Administration tool to add the DATASYNCSERVER to the server list and to configure the DataSync.cfg configuration file. This file specifies whether the DBSyncController is notified when the device is backed up and the name of the DBSyncController to notify when a device is backed up.

For more information, refer to:

- "Adding the DATASYNCSERVER to the server list" (page 136)
- "Configuring the DATASYNCSERVER" (page 136)
- "Starting the DATASYNCSERVER" (page 138)

Adding the DATASYNCSERVER to the server list

Prerequisites

If you are using only the Backup and Restore tool, verify that the following three servers are running:

- Context Server
- MNSD Agent
- PP NP Config Server

Note: If you are also using the Data Synchronization Administration tool, verify that, in addition to the previous three servers, **the PP Command Access Svr is running**.

Procedure steps

1 In the application main window, select **System** -> **Administration** -> **Server Administration** to open the **Server Administration** tool.

2 From the Security menu, select the **Authorize** command.

3 In the **SVM Enter Authorization Password** dialog box, type a valid password and click **OK**.

4 From the **Edit** menu, select **New Server**.

The SVM New Server Selection window displays.

5 Expand the server categories to locate the Data Sync Server in this list.

6 Select the server and click **Select Server**.

The SVM Edit Server dialog displays with the fields pre-filled with default information.

7 In the **Startup command** field, type the following command, as follows:

```
/opt/MagellanNMS/bin/dataSyncServer
[-help]
```

8 Enable the **Automatic startup at reboot time** option.

9 Click **OK**.

A confirmation dialog displays.

10 Click **Yes** to accept this information.

11 Click **Close**.

Configuring the DATASYNCSERVER

If you have a database, you must configure the DATASYNCSERVER.

Procedure steps

1 In the SVM Server window, select the **Data Sync Server** entry.

2 Right-click and select **Server Configuration**.

The Configuration Editor displays.

3 Expand the **Embedded Servers** selection.

4 Select **DBSyncController**.

5 Verify that the **Enabled** field is set to **True**.

6 From the **File** menu, select **Save**.

7 Select the Data Sync Server entry again and right click to select **Edit Configuration** and then **DBSync Controller**.

8. Enter values in the displayed fields. For information on these fields, refer to 241-6001-807 *Preside MDM Network Backup and Restore* and 241-6001-400 *Preside MDM Administration Database User Guide*.

   Note: You must enter values for the authentication fields.

9. From the **File** menu, select **Save**.

10. From the **File** menu, select **Exit**.

Starting the DATASYNCSERVER

You can either use the command line, as described earlier, to start the DATASYNCSERVER or use the following procedure.

Procedure steps

1. In the SVM Server window, right-click on the **Data Sync Server** entry.

2. From the pop-up menu, select **Start**.

   The Data Synchronization Server starts.

Interdependencies

To support the Current View mode of backup/database synchronization, you must add the -notification option to the command line for the Backup Controller using the Server Administration tool. For more information, refer to "Backup Controller (NSCTLBCK)" (page 55).

Exit Codes

The exit codes for the DATASYNCSERVER are shown in the following table.

Table 20
Exit codes for the DATASYNCSERVER

Exit codes	Description
50	Recoverable application error. Server will restart.
100	Fatal application error. Server will not restart
143	Server killed via kill signal

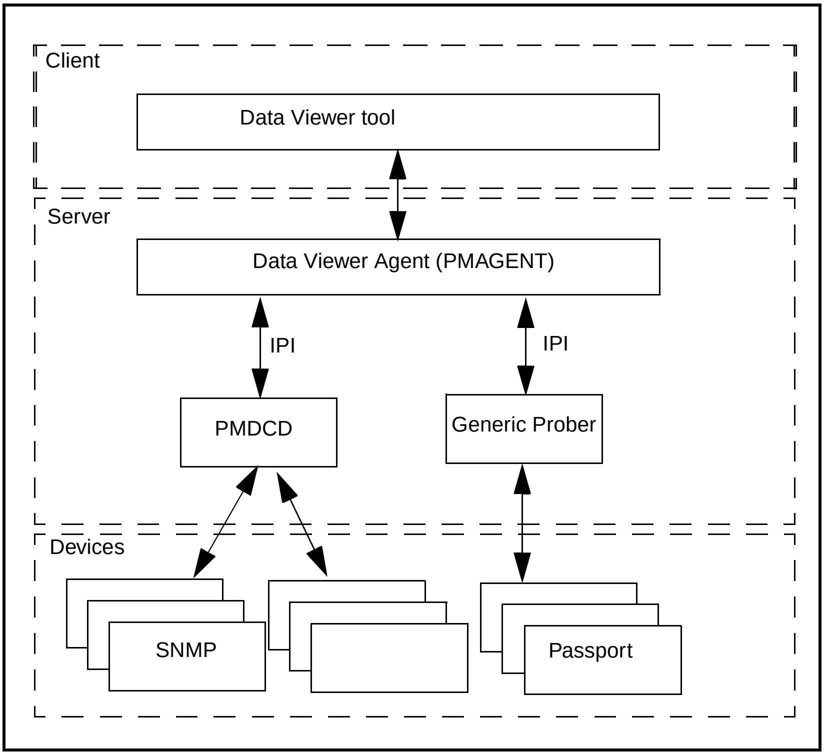
Chapter 12
Data Viewer Agent (PMAGENT)

This section contains information about the Data Viewer Agent (PMAGENT). This section contains the following information:

About the PMAGENT

The Data Viewer application lets you collect and display real-time performance information for Passport and SNMP devices in the network. For details on using the Data Viewer, see 241-6001-031 *Preside MDM Performance Management User Guide*. Figure 15 shows the architecture of the Data Viewer within Preside Multiservice Data Manager.

Figure 15
Data Viewer architecture



The PMAGENT provides information to the graphical user interface (GUI). The agent parses metric files based on the prefix of the component name and then builds a tree that the GUI can query. When a request for polling occurs, the agent sets up communication to the device and starts polling for the requested values. The agent sends the values to the GUI where they are displayed.

The agent also collects statistic attributes from network components using the Generic Prober (GP) server through Internal Program Interface (IPI). The agent starts and establishes a connection to the GP server. The GP server allows the agent to connect to a specific node and specify a list of component/ attribute pairs to be monitored during a polling interval.

The agent collects statistic attributes from SNMP and MPE 9500 components using PMDCD.

Managing the PMAGENT

See the following sections for information to manage the PMAGENT:

- Configuration (page 141)
- Suggested name in Server Administration (page 142)
- Start-up command (page 142)

Configuration

The information for the metric file is extracted from the map file located in /opt/MagellanNMS/lib/cfg/pmr/pmrtype.map. You can customize this file and override the default settings. To customize, first copy the pmrtype.map file from the directory /opt/MagellanNMS/lib/cfg/pmr/ into the directory /opt/MagellanNMS/cfg/pmr. Then make your changes in /opt/MagellanNMS/cfg/pmr. You can copy the map file to a different directory, but you need to specify the location of the .map file when starting the PM agent from the command line. For details, see Start-up command (page 142).

The supported options for the agent file are as follows:

- <device prefix>
  where <device prefix> specifies the device prefix as know to Preside Multiservice Data Manager (MDM). The device prefix is EM for Passport devices and SRS for MPE 9500 devices.
- <metric file>
  where <metric files> specifies the full path name of the metric file. One or more metric files can be specified. If you specify more than one metric file, use a semicolon (;) to separate the metric file names.
- <Passport type>
  where <Passport type> is the family type of the Passport devices. This information must be the same as the PPType field in the /opt/MagellanNMS/lib/cfg/PPTypeConfig.cfg file.
- <service name>
  where <service name> is PMDCD for SNMP MPE 9500devices.

You can specify one or more map files. Each map file contains at least one device type definition. The following are the search paths for the map files:

- /opt/MagellanNMS/cfg/pmr (defined by you)
- /opt/MagellanNMS/ext/lib/cfg/pmr (defined by second-party developers)
- /opt/MagellanNMS/lib/cfg/pmr (defined in MDM software)

The PM agent loads all the map files with the .map extension using the search paths. If a map file or a map entry associated with a device type is already loaded, it is not replaced by the same map file or map entry in the other directories, as follows:

- Map file or map entry found in /opt/MagellanNMS/cfg/pmr are loaded
- Map file or map entry found in /opt/MagellanNMS/ext/lib/cfg/pmr or /opt/MagellanNMS/lib/cfg/pmr are ignored.

Suggested name in Server Administration

The recommended name to use for the PMAGENT in the Descriptive Information field of the Server Administration tool is: Data Viewer Agent.

Start-up command

The command to start the PMAGENT is:

```
/opt/MagellanNMS/bin/pmagent [-map <config file path>]
```

where:

`<config file path>` is the location of an optional pmrtype.map file. For information about this file, see Configuration (page 141).

Chapter 13
Data Viewer Data Collection Daemon (PMDCD)

This section contains information on the Data Viewer data collection daemon (PMDCD). This section contains the following information:

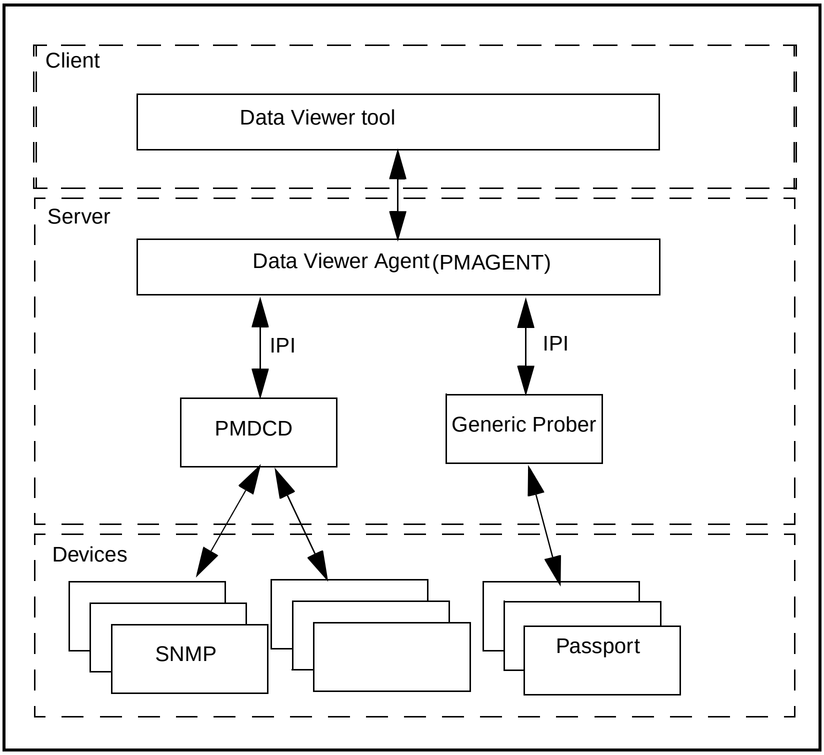
- “About the PMDCD” on page 143
- “Managing the PMDCD” on page 144

About the PMDCD

The Data Viewer application lets you collect and display real-time performance information for Passport and SNMP devices in the network. For details on using the Data Viewer, see 241-6001-031 *Preside MDM Performance Management User Guide*.

The PMDCD provides a means to collect fault information from SNMP devices in the network and provide it to the Data Viewer Agent, which in turn provides the information to users of the Data Viewer tool. The figure Data Viewer architecture (page 144) shows the architecture of the Data Viewer within Preside Multiservice Data Manager.

Figure 16
Data Viewer architecture



Managing the PMDCD

See the following sections for information to managed the PMDCD:

- Configuration (page 144)
- Suggested name in Server Administration (page 145)
- Startup command (page 145)

Configuration

A default PMDCD configuration file /opt/MagellanNMS/lib/cfg/pmr/pmdcd.cfg is provided with Preside Multiservice Data Manager software.

To customize the Data Viewer data collection configuration file, copy the file pmdcd.cfg from the directory /opt/MagellanNMS/lib/cfg/pmr into the directory /opt/MagellanNMS/cfg/pmr. Make the configuration changes to the file pmdcd.cfg in the directory /opt/MagellanNMS/cfg/pmr. You can also copy the configuration file to a different directory instead of using the directory /opt/MagellanNMS/cfg/pmr. If you copy the configuration file to a directory other than /opt/MagellanNMS/lib/cfg/pmr or /opt/MagellanNMS/cfg/pmr, specify the location of the configuration file when starting PMDCD using the command line option -cfg. For details about starting the PMDCD, see Startup command (page 145).

The supported options for the configuration file are as follows:

- logFile:<log file>
  where <log file> specifies the name of the file into which the logs are written. The default log file is /opt/MagellanNMS/data/pmdcd.log.
- snmpTimeOut:<time out value in seconds>
  where <time out value in seconds> specifies the number of seconds the process waits for a response from the device. The default is 30 seconds.
- sysName:<device prefix>:<oid>
  where <device prefix> specifies the prefix for the device and <oid> specifies the oid value to use for reachability requests on the devices with the specified device prefix. There is no default setting for this option.

Suggested name in Server Administration

The recommended name to use in the Descriptive Information field of the Server Manager Administration tool when starting this server is: Data Viewer DCD.

Startup command

Use the following command to start the PMDCD:

```
/opt/MagellanNMS/bin/pmdcd [-cfg <config filepath>]
```

where:

<config file path> is the absolute path name to a customized configuration file pmdcd.cfg. For information about this file, see Configuration (page 144).

Chapter 14
End-to-end Server (ETESERVER)

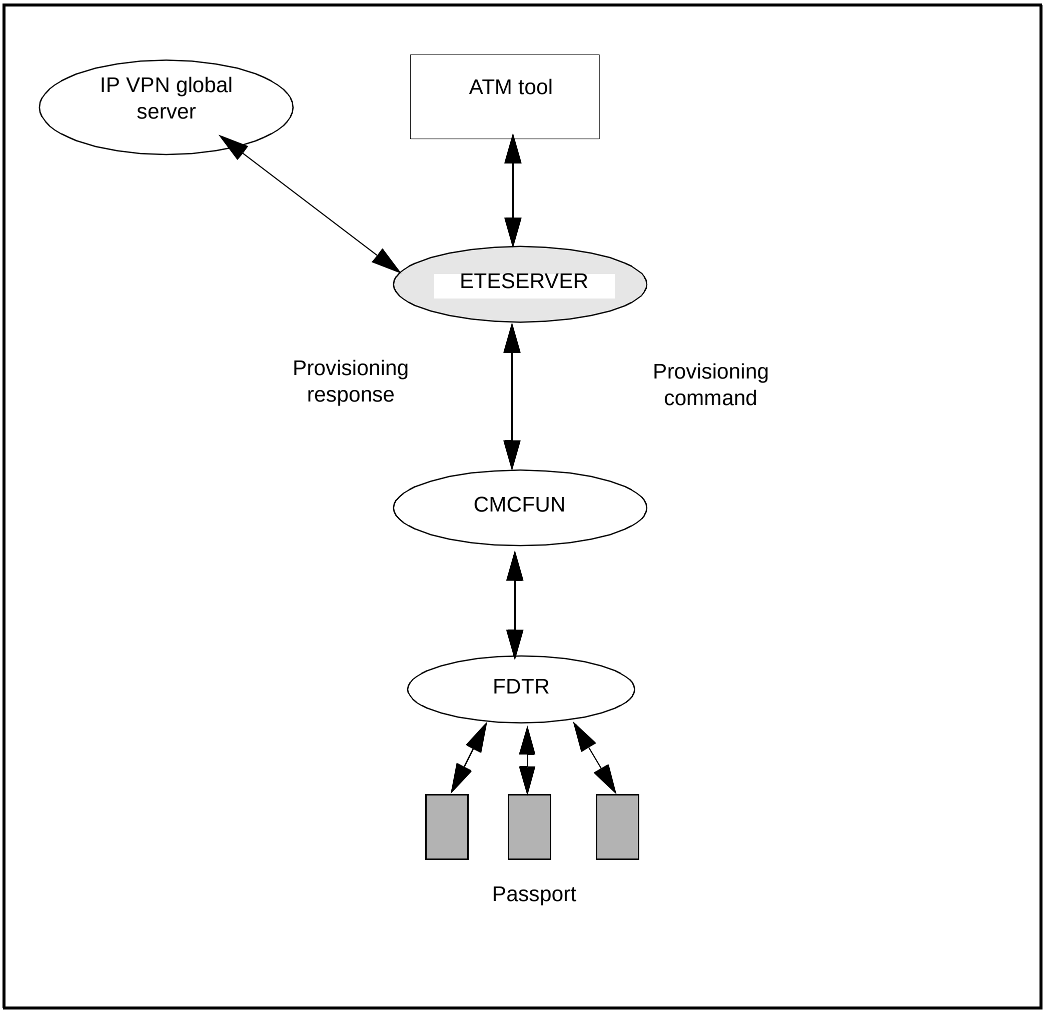
This section contains information on the end-to-end server (ETESERVER). See the following topics for more information:

- "About the ETESERVER server" (page 147)
- "Managing the ETESERVER" (page 148)
- "Interdependencies" (page 149)
- "Exit codes" (page 149)

About the ETESERVER server

The ETESERVER acts as an intermediary between the end-to-end provisioning applications and the Preside Multiservice Data Manager Command Console Functional Process (CMCFUN) server.

Figure 17
ATM data flow diagram



Managing the ETESERVER

Use the Server Administration tool to enter or to edit the startup command and to start, stop, and configure this server to start automatically when the workstation is rebooted. Any changes you make to the startup command or to the configuration file become active whenever the server is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

For more information, see the following:

- “Suggested name in Server Administration” (page 149)
- “Startup command” (page 149)

Suggested name in Server Administration

The recommended name for the server is End-to-End Server.

Configuring ETESERVER with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the ETESERVER is as follows:

```
/opt/MagellanNMS/bin/eteserver [-p <portno>]
```

where:

`-p <portno>` specifies the TCP port number for listening to incoming requests. The default value is 6600. This port number must not be used by any other process.

Interdependencies

The ETESERVER depends on the HGDS, CM, and CMCFUN servers. The ATM tool depends on the ETESERVER.

Exit codes

Exit codes for the ETESERVER are shown in the following table.

Table 21
Exit codes for the ETESERVER

Exit code	Description
51	Out of memory.
55	Bad argument on command line.
59	Could not initialize IPC system.
60	Could not register service. (Is the server running?)

Chapter 15
General Management Data Router (GMDR)

This section contains information on the General Management Data Router (GMDR). See the following topics for more information:

About the GMDR server

The GMDR server is mandatory, regardless of the types of nodes in your network. It performs the following functions:

- collects surveillance information from network elements in the network through a number of access servers (FMDR, NCSMGR, NMDR, OAMC)

- assigns criticality values to the components it manages using information stored in criticality schema and override configuration files.
- stores the alarms and surveillance information in a common format
- supplies the alarms and surveillance data to fault tools and servers according to filtering criteria supplied by the client application. This criteria includes the criticality of a component and the criticality threshold set by the client of the GMDR server.
- dispatches manual alarm clear requests
- manages alarm acknowledgment. The GMDR server processes ack and unack requests, and sends an ackStateChange notification to the SURNUP server for each request. The ackStateChange notification tells the SURNUP server when to ack or unack the component state. The GMDR server sends the resulting alarm up to clients and forwards the request down to any GMDR server at the next level in the hierarchy.
- accepts alarms from equipment that is not supported by Preside Multiservice Data Manager, such as external modems. These alarms and state change notifications are supplied to the GMDR server through the OAM Collector (OAMC) server and for Nortel Networks Multiservice Provider Edge 9500, through the Injected Management Data Router (IMDR). This method of viewing alarms is used only when access to alarms from other tools, such as the AlarmDisplay tool, is not available.

Surveillance data routing

The GMDR server receives data from the following sources:

- the FMDR server. The GMDR server uses the FMDR server to receive alarm and state change event notifications from Passport nodes and to retrieve the initial state of the their components. See “Passport Management Data Router (FMDR)” (page 389) for a description of the FMDR server.
- the NMDR server. The GMDR server uses the FMDR server to receive alarm and state change event notifications from MPE 9500 nodes and to retrieve the initial state of the their components. When redundant information is received from the same MPE 9500 devices, GMDR

resolves duplicate redundant information. See “MPE 9500 Management Data Router server (NMDR)” (page 283)for a description of the NMDR server.

- the DMDR server. The GMDR server uses the services of the DMDR server to collect surveillance information from DPN and other devices.

  Because the Meridian I Management interface information also comes through the NCS, the DMDR server is also needed to make this information available to the GMDR server.

- the SMDR server. The GMDR server uses the services of the SMDR server to collect surveillance information for devices that are managed through the SMDR-based DCD. See “SNMP Management Data Router (SMDR)” (page 509) for a description of the SMDR server.
- the OAMC server. The OAMC server the Preside Multiservice Data Manager (MDM) server that collects logs generated by MDM and sends alarms to the GMDR server, makes logs available to the System Log Display tool, forwards security-related logs to the Security Audit Log Collector (SALC) server and collects and writes MDM audit events to a file.
- an inbound alarm API. The GMDR server accepts alarms from equipment that is not supported by Preside Multiservice Data Manager. These are supplied to the GMDR server through an inbound alarm API. For a description of this API, see 241-6001-203 *Preside MDM Alarm and Status API Reference Guide*.

  Instead of using the inbound alarms API to inject alarms into GMDR consider injecting alarms through to the IMDR server through the IMDR API. Using IMDR and the IMDR API has a number of advantages. These include:

  — The IMDR API accepts component deletions.

  — The IMDR API accepts component property queries. This allows you to obtain information about an injecting component. For example, the component’s IP address.

- a subordinate GMDR server that is located on this workstation or on another workstation. This situation can occur when hierarchical GMDR servers are used. For an explanation of the term subordinate GMDR server see “Hierarchical GMDR servers” (page 159).

  You can use the GMDR Administration tool to set a criticality threshold so that only information for components having a high enough criticality is reported by a subordinate GMDR. For the instructions to set a threshold when configuring access to a subordinate GMDR server, see 241-6001-303 *Preside MDM Administrator Guide*.

Resynchronization-based automatic component deletion

Components are automatically deleted from the GMDR when a disconnected FMDR, NMDR, SMDR, or subordinate GMDR is reconnected with the GMDR. This is referred to as resynchronization-based automatic component deletion. For disconnects that are the result of administrator action, the components are deleted on the GMDR and the GMDRs downstream clients and are also deleted from the network model and fault applications using the fault API or EPI. For disconnects that are the result of communication failure, the components stay in the unknown state until the component indicates no alarms; the component is then eligible for deletion. Exceptions to resynchronization-based automatic component deletion include:

- partial view sub-servers
  Components not deleted on a server can be deleted by a client of this server if the client has multiple data providers for a switch and one, or more, of these data providers provides only a partial view of the switch.

- component discovered by event, only
  Components that exist in the network can be deleted throughout the fault stack. Most components discovered by event, only, have already been removed from the network model because the network model recognizes these components as dynamic.

- components with historical alarms
  Components with historical alarms are tagged as eligible for deletion but are not deleted. After a resynchronization of servers, if the component no longer has any alarms, the component is deleted. This situation only occurs if the disconnect is due to network congestion; the active alarms become historical alarms.

- a server using pre-release R14.1 Preside Multiservice Data Manager (MDM) software and any server above this server in the fault stack; unless one of the servers above this server in the fault stack is a redundant server using release R14.1, or greater, software

The figure “GMDR data flow diagram” (page 156) illustrates the data flow through the GMDR server to its various clients.

Figure 18
GMDR data flow diagram

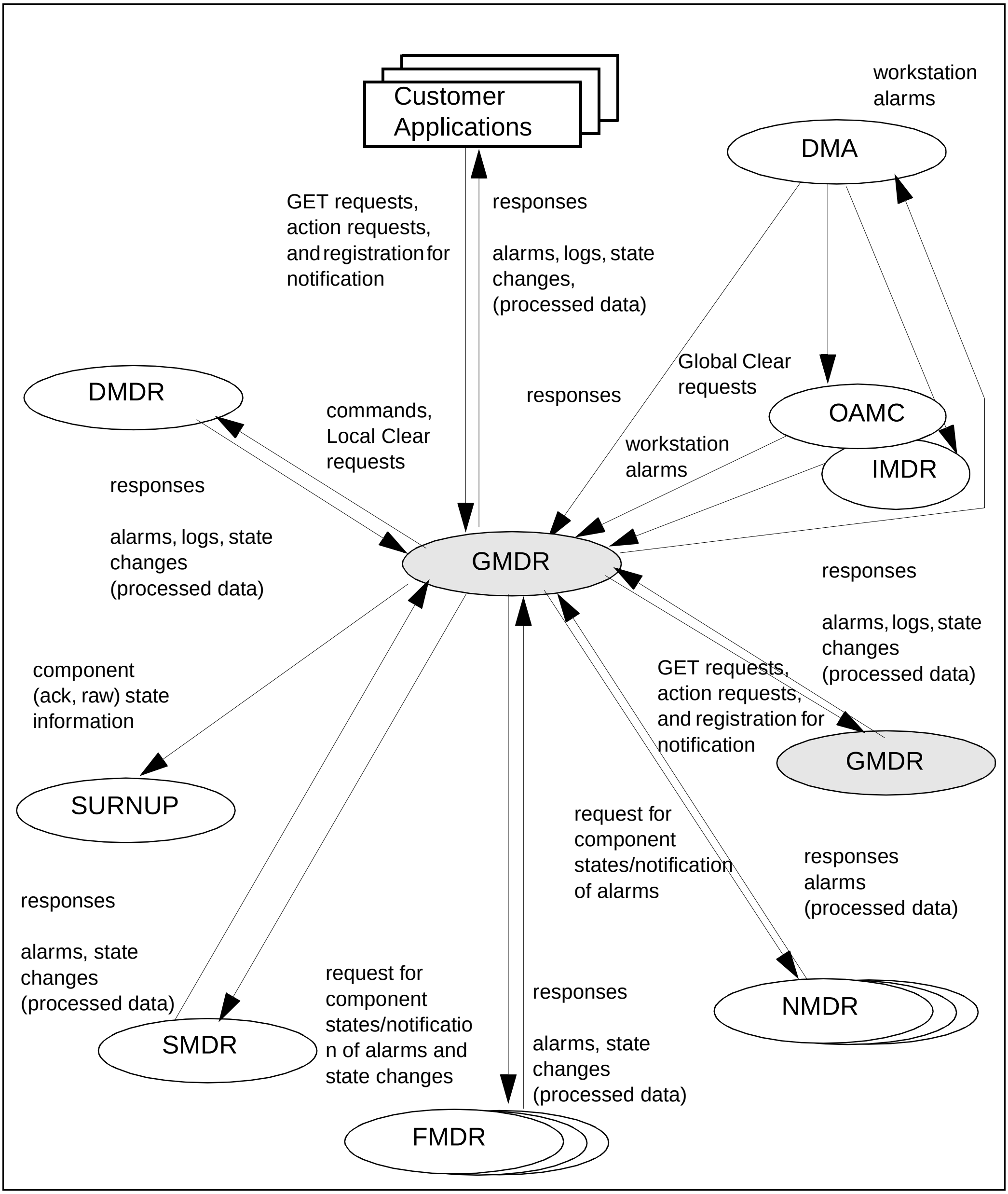
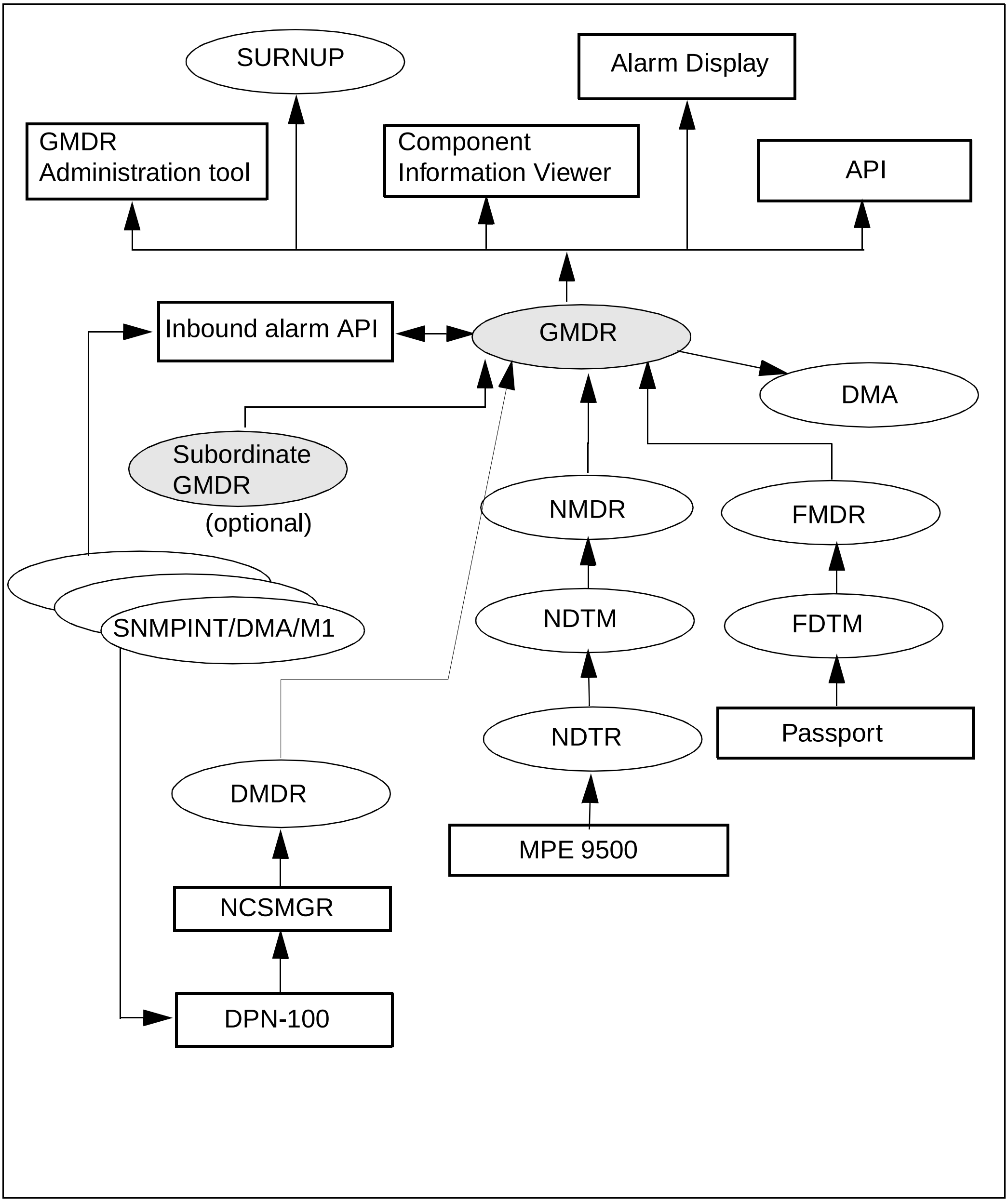

Customer Applications
workstation alarms
DMA
GET requests, action requests, and registration for notification
responses
alarms, logs, state changes, (processed data)
Global Clear requests
DMDR
commands, Local Clear requests
responses
OAMC
workstation alarms
IMDR
responses
alarms, logs, state changes (processed data)
GMDR
responses
alarms, logs, state changes (processed data)
component (ack, raw) state information
GET requests, action requests, and registration for notification
GMDR
SURNUP
request for component states/notification of alarms
responses alarms (processed data)
responses
alarms, state changes (processed data)
request for component states/notification of alarms and state changes
responses
alarms, state changes (processed data)
NMDR
SMDR
FMDR

Figure 19
Flow of surveillance data in a mixed network



GMDR sub-server administration

To allow the GMDR server to collect surveillance data from the FMDR, NMDR, DMDR, and SMDR servers, and from a subordinate GMDR server that runs on this workstation, you must use the GMDR Administration tool to specify the server name, host, registration user ID and password for each FMDR, NMDR, DMDR, SMDR, and subordinate GMDR server that will supply GMDR with surveillance data. The configuration data is stored in file /opt/MagellanNMS/cfg/GMDR.cfg. You can change or modify the connections through the GMDR Administration tool.

GMDR sub-server connections

When the GMDR server is initialized, it automatically connects to all of the servers defined in the file. If the GMDR server fails to connect to a server, it continues to retry every 30 seconds until stopped by user action. If the user ID and passwords are not valid, the connection to the server is dropped and error messages are sent to the Preside Multiservice Data Manager log and to the message window of the GMDR Administration tool.

Alarm Acknowledgment

The GMDR server performs the following functions for Alarm Acknowledgment (acknowledging (ack) and unacknowledging (unack) active alarms):

- processes new ackAlarm client requests in the form of an API ACTION, for acknowledging and unacknowledging alarms, whereby the resulting alarms are sent up to all clients. Acknowledgment attributes are added to, or updated in, the relevant active alarms.
- forwards ack or unack requests down to other GMDR servers in the hierarchy, so that each lower level GMDR can reflect the same acknowledgment information
- time stamps ack and unack requests if no ackTime is provided, and forwards these requests including the time stamp to any other GMDR at the next level down in the hierarchy
- ensures that acknowledged or unacknowledged alarms are discarded as duplicates, or not discarded as duplicates, as necessary
- issues an ackStateChange notification when the effect of acknowledging an alarm causes a component state to become acknowledged

- issues an ackStateChange notification when the effect of unacknowledging an alarm causes a component state to become unacknowledged
- issues an ackStateChange notification when the effect of clearing an alarm causes a component state to become acknowledged
- issues an ackStateChange notification when the effect of receiving a new alarm causes a component state to become unacknowledged

There may be instances where alarm acknowledgement is not persistent. For example, if an alarm is temporarily lost due to the loss of GMDR connection, the alarm may reappear without alarm acknowledgement.

You can use the GMDR Administration tool to configure, monitor, and perform administrative tasks on all of the GMDR servers in your network. For instructions to use the GMDR Administration tool, see 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use this tool.

Alarm-based and State-based surveillance

Alarm Acknowledgment provides a link between alarm-based surveillance and state-based surveillance. If the Ack Alarms dialog is invoked from an alarm list menu item, an ackAlarm request is sent to the GMDR server for each selected alarm (alarm-based application). If the Ack Alarms dialog is invoked from a menu item, an ackAlarm request is sent to the GMDR server for every active alarm for the current component (state-based application).

The API protocol used to inform GMDR of an alarm acknowledgment or unacknowledgment, is based on the Local Clearing protocol, where GMDR forwards the ACTION to any other GMDR servers below it in the hierarchy. The GMDR API provides these new alarm attribute values to its clients via GET requests and NOTIFY actions. For additional information on API protocol, see 241-6001-203 *Preside MDM Alarm and Status API Reference Guide*.

Hierarchical GMDR servers

You can deploy several GMDR servers in a hierarchy as shown in the sample deployment in the figure “Sample deployment of GMDR servers in a hierarchy” (page 162). The hierarchy shown in this figure is only intended as a sample, many other arrangements are possible.

For more information, see the following:

- “Advantages” (page 160)
- “Naming hierarchical GMDR servers” (page 161)

Advantages

The ability to set up a hierarchy of GMDR servers provides a number of advantages, as follows:

- A large network can be divided into regions to gather and manage surveillance information on a regional basis or it can be divided into groups according to the equipment type to gather and manage surveillance information on an equipment basis.
- Central surveillance can be established by setting up a top GMDR server at a convenient location that gathers surveillance information from subordinate GMDR servers in each of the regions (or functional areas).
- Using the criticality mapping capabilities of GMDR, it is possible to set a criticality threshold on a connection to a subordinate GMDR server to filter out alarms and state changes, such that only those for the most important components managed by the subordinate GMDR server are reported. This can greatly reduce the amount of information the superior GMDR server has to manage.
- It is not necessary to inject alarms from equipment that is not supported by Preside Multiservice Data Manager into every GMDR server in the network to have the alarms appear on more than one workstation. Alarms injected into the network at a subordinate GMDR server through an inbound alarm Applications Programming Interface (API) are automatically propagated up the hierarchy to the top GMDR server.

  Instead of using the inbound alarms API to inject alarms into GMDR consider injecting alarms through to the IMDR server through the IMDR API. Using IMDR and the IMDR API has a number of advantages. These include:

  — the IMDR API accepts component deletions

  — the IMDR API accepts component property queries. This lets you obtain information about an injecting component. For example, the component’s IP address.

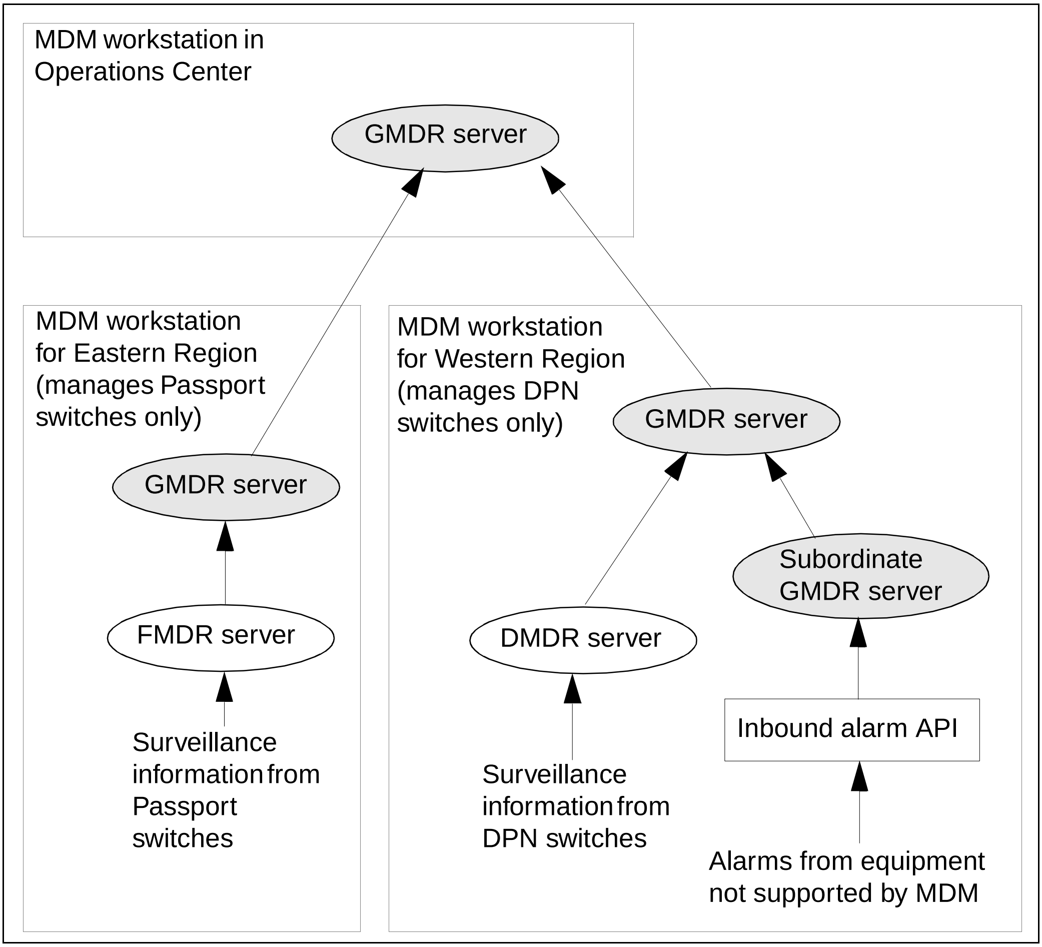
Naming hierarchical GMDR servers

A GMDR server can run on a workstation by itself or can run on the same workstation as one or more GMDR servers that are subordinate to it. When the GMDR server and its subordinate(s) run on the same workstation, the servers require different names to distinguish one from the other.

The GMDR server that has a subordinate GMDR server reporting to it, also known as the superior GMDR server, uses the default name GMDR and the subordinates must be named GMDR_<name>, where <name> is a unique name that differentiates the subordinate server from its superior and from any other subordinate that runs on the same workstation. For example, if a superior server and a subordinate run on the same workstation, the superior server must use the default name GMDR and the subordinate could be named after the region or functional area it manages: GMDR_SOUTHWEST.

You must enter the names of the subordinate GMDR servers into the Name field of the Add Server dialog box when setting up the servers that feed surveillance information to a workstation. For more information on configuring GMDR to access the surveillance servers, see 241-6001-303 *Preside MDM Administrator Guide*.

Figure 20
Sample deployment of GMDR servers in a hierarchy


MDM workstation in Operations Center
GMDR server
MDM workstation for Eastern Region (manages Passport switches only)
GMDR server
FMDR server
Surveillance information from Passport switches
MDM workstation for Western Region (manages DPN switches only)
GMDR server
Subordinate GMDR server
DMDR server
Inbound alarm API
Surveillance information from DPN switches
Alarms from equipment not supported by MDM

Surveillance data storage

GMDR provides short-term storage of data from network elements. The GMDR database is reset each time GMDR is initialized. The GMDR database stores the following information:

- the current raw state for DPN components and the current raw state and OSI state for Passport components. No state history is provided.
- alarms. Duplicate alarms are automatically discarded.

  The maximum number of stored alarms is specified in the GMDR startup command. When the maximum number of stored alarms is reached and a new alarm is received, the oldest non-active alarm is discarded. If the database contains only active alarms, the incoming alarm is compared to the oldest alarm. If the oldest alarm in the database has a higher severity, the incoming alarm is discarded. To prevent a single component from filling the database to capacity, the maximum number of alarms to be stored for a single component can be specified in the GMDR startup command.

Component criticality mapping

The GMDR server automatically assigns a criticality value to each component it manages. This value is in the 0 to 255 range. A GMDR client (a superior GMDR server, the SURNUP server, or an API client) can set a criticality threshold for receiving alarms and state change information from a GMDR server. Alarms and changes from components whose criticality values are lower than the threshold are filtered out so that the GMDR client only receives alarms and state changes from the most important (critical) components.

Component criticality is configured with the three following files:

- a default component criticality schema file provided with the Preside Multiservice Data Manager software (/opt/MagellanNMS/lib/cfg/GMDRCriticality.cfg).

  This file cannot be modified.

- a customized component criticality schema file (/opt/MagellanNMS/cfg/GMDRCriticality.cfg).

  You can create this file to complement or modify the configuration specified by the default component criticality schema file.

- a customized component criticality overrides file (/opt/MagellanNMS/cfg/GMDRCritOverrides.cfg)

  You can create this file and fill it with exceptions for individual components whose criticality values do not match the values assigned in the default and customized component criticality files.

Whenever the system is restarted, the GMDR database is reset, or the GMDR server receives a HUP signal, the contents of these files are read in the following order:

- default component criticality schema file
- customized component criticality schema file
- customized component criticality override file

GMDR appends the mapped criticality value to most of the information records it provides to its clients (raw and OSI states GETs and change notifications and alarms but not DPN-100 Status Records). The clients can therefore filter on the criticality value to control the information they get.

A criticality mapping that is based only on the configuration files is sometimes insufficient and must be adjusted in real time because of an ambiguous schema or component model. For example, for DPN-100 links. These links are important and should therefore have fairly high criticality values. However software deduces states of these links from their terminating ports, which are also used for access purposes and are therefore normally assigned a much lower criticality value. When the criticality threshold is set to a value greater than the criticality value assigned in the configuration files, information about these ports is not reported. As a result, information about the links is not reported either. To avoid this situation, GMDR performs a criticality auto-adjustment. If a link is created and attached to a sub-component that is assigned a lower criticality value, GMDR automatically raises that sub-component’s criticality, and those of its parents, as needed, to that of the links. The same applies for a newly created sub-component that is assigned a higher criticality value than that of its parent. When GMDR auto-adjusts the criticality of these components, it re-emits their alarms and raw state information so that alarms and states which were previously filtered out are reported to the client for which the threshold was set. Criticality auto-adjustment can be disabled with the -E option in the server's startup command.

Managing the GMDR server

In order for Preside Multiservice Data Manager (MDM) clients to retrieve surveillance data from GMDR, you must configure the servers for network access, surveillance access and provisioning access to Passport, as described in 241-6001-303 *Preside MDM Administrator Guide*.

After software installation, GMDR is in the Server Administration server list. However, the GDMR server is configured to startup using the default startup command. You can change this by editing the GMDR server, as described in 241-6001-303 *Preside MDM Administrator Guide*. After editing the server parameters you can use the Server Administration tool to manually start the server, or you can edit the parameters for all of the Passport servers. For more information on configuring MDM servers for Passport nodes, see 241-6001-303 *Preside MDM Administrator Guide*. Then reboot the workstation to start the servers automatically.

For more information, see the following:

- "Suggested name in Server Administration" (page 165)
- "Startup command" (page 166)

Suggested name in Server Administration

The recommended names for GMDR servers are as follows:

- GMDR for the superior GMDR server on the workstation. If there is only one GMDR server on the workstation, this is the name it will use.
- GMDR_<name> for any subordinate GMDR servers that run on the workstation.

Configuring DBNLWatch with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start up the GMDR server has the following syntax:

```
/opt/MagellanNMS/bin/gmdr \
[-n <service name>] \
[-a <alarm size>]
[-b <msg congestion threshold>] \
[-A <max alarms per component>]
[-C <GMDR*.cfg path>] \
[-S <:hostnmae:servername:userid:passwd:>]
[-x] \
[-D] \
[-O] \
[-E] \
[-m <criticality schema file path>] \
[-o <criticality overrides file path>] \

[-h]
[-H]
[-R]
[-s]
[-f]
[-P <hash table size>]
[-L] <server heartbeat intervals>
[-W] <loss connection window in seconds>
```

where:

`[-n <service name>]` specifies the alternative service name, prefixed with GMDR_, of a subordinate GMDR server to be started. This option only needs to be specified when two or more GMDR servers are running on the workstation and you wish to start a subordinate GMDR server. This parameter is the unique name for the subordinate GMDR server. Omitting this parameter starts the superior GMDR server on this workstation.

`[-a <alarm DB size>]` specifies the maximum number of alarms to store. The default value is 2000.

CAUTION
Risk of performance degradation
Do not specify values greater than 10,000. High values can lead to degradation of workstation performance and an increase in response time due to the large amount of memory and time needed to scan the very long alarm list associated with a large number of alarms.

`[-b <msg congestion threshold>]` specifies the number of congested replies before the client is cut off. If this option is not specified, the default value of 10,000 is used. The maximum allowable is 50,000.

`[-A <max alarms per component>]` specifies the maximum number of alarms to store per component. The default value is 50. Maximum value is the alarm database size.

`[-C <GMDR*.cfg path>]` specifies an alternate GMDR.cfg file. The default configuration file path is /opt/MagellanNMS/cfg/GMDR*.cfg.

`[-S <:hostname:servername:userid:passwd:>]` specifies the host name the server name, the user ID, and the password of a Preside Multiservice Data Manager (MDM) server from which GMDR obtains surveillance information. This option performs the same function as the GMDR Administration tool. Use the GMDR Administration tool to set up GMDR server to obtain surveillance information from an MDM server. For information about the GMDR Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

`[-x]` disables the storage of historical alarms

`[-D]` clears discarded SET alarms

`[-O]` turns off optimization of queries. By default, GMDR optimizes queries when criticality thresholding is applied (assuming that all criticalities are properly ordered in the component tree). This option turns the optimization off, and forces GMDR to avoid pruning its search when it finds a component with a lower criticality.

`[-E]` disables auto-adjustment of alarm criticality

`[-m <criticality schema file path>]` specifies a file to use as the custom criticality mapping schema instead of the default file (/opt/MagellanNMS/cfg/GMDRCriticality.cfg). This option can be used if multiple GMDR servers are to run on the same workstation but with different criticality mappings.

`[-o <criticality overrides file path>]` specifies the file to use as the custom criticality override instead of the default file (/opt/MagellanNMS/cfg/GMDRCritOverrides.cfg). This option can be used if multiple GMDR servers are to run on the same workstation but with different criticality mappings.

`[-h]` displays the syntax of the GMDR server's startup command.

`[-H]` resynchronizes only active alarms from a sub server.

`[-R]` forwards raw state change notifications resulting from a resynchronization only to SurNUp clients.

`[-s]` does not forward OSI state change notifications to any client.

`[-f]` disables the resynchronization-based deletion of components.

[-P <hash table size>] specifies the size of the hash tables in GMDR. The default hash table size is 8192. The number of switches in the GMDR database determines the optimal use of the hash table. The default value works optimally with networks up to 2500 switches. In general, the hash table size should be three times the number of switches. This value is expressed as a power of 2. For example, in a network of 4000 switches, the hash table size is specified as 16384.

`-L` specifies the time in minutes between heartbeats to GMDR subservers. The default value is 2 minutes. GMDR sends a heartbeat to every subserver. If a subserver fails to respond to the heartbeat, the subserver is declared to be lost.

`-W` specifies the size in seconds of the loss conn window for each server/module pair. The default value is 0 (no window). The Loss of Connection window is configured in seconds on the command line of GMDR. A loss of connection SET alarm must be received from all servers of a module within a time that all server windows overlap to cause the generation of a loss of connection SET alarm.

The following table lists the command line options that are related to congestion and resynchronization of the GMDR server.

Table 22
GMDR command line options related to resynchronization and notification congestion

Option	Description
-a <nb>	Controls the number of alarms that GMDR can store. The priority is given to an active alarm list. The default is 2000, and the maximum is 50,000.
-b <nb>	Controls the number of congestion buffers allowed for a client. The default is 10,000.
-A <nb>	Controls the maximum number of alarms that can be stored per component. The default is 50.
-H	If this option is set, it resynchronizes only active alarms from a a sub server. By default, it resynchronizes active and historical alarms.
-R	If this option is set, it forwards raw state change notifications resulting from a resynchronization only to SurNUp clients. Raw state change notifications are explicated towards GMDR clients only if no supporting alarm or OSI state change notification is not available. By default, all clients can get all state change notifications.
D:\Profiles\dmcken1\My Documents\WebWorks\FrameSource\6_310\gmdr.fm	

Table 22 (Continued)
GMDR command line options related to resynchronization and notification congestion

Option	Description
-s	If this option is set, OSI state change notifications are not forwarded to any client.
-G	If this option is set, GET requests from GMDR clients are optimized to avoid sending UNKNOWN middle components.
-T <minutes>	Specifies the number of minutes to throttle Resynch Done notifications to clients after a completed resynchronization. The default is 2 minutes.
D:\Profiles\dmcken1\My Documents\WebWorks\FrameSource\6_310\gmdr.fm	

Configuration

The criticality threshold used for filtering alarms and state change information from a subordinate GMDR server is configured with the GMDR Administration tool. For more information on the GMDR Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

However, the criticality value assigned to a component is governed by three configuration files:

- a default component criticality schema file provided with the Preside Multiservice Data Manager software which cannot be modified.
- the two customizable configuration files described in the following sections:
  - "Customized component criticality schema file" (page 170)
  - "Customized component criticality overrides file" (page 172)

Customized component criticality schema file

The customized file component criticality schema file (/opt/MagellanNMS/cfg/GMDRCriticality.cfg) contains customized criticality mappings and is loaded after the default component criticality schema file that is provided with the Preside Multiservice Data Manager (MDM) software (/opt/MagellanNMS/lib/cfg/GMDRCriticality.cfg). The customized mappings therefore override the defaults provided with MDM.

Before creating a customized component criticality schema file, look at the default component criticality file in /opt/MagellanNMS/lib/cfg/GMDRCriticality.cfg. The default file shows the default criticality values assigned to components and the file format is similar to the one used in the customized component criticality file.

The file format is as follows:

```
Component_type: <component type>
Criticality: <criticality value>
```

where:

`component type` is the component type being given a criticality mapping. Its value can be specified as follows:

`<module type or *>` identifies a module type (For example: EM for Passport). If a single asterisk (*) is used, this entry specifies the default criticality mapping for all modules and their subcomponents whose criticality is not mapped by other means, such as by an entry in the customized components criticality overrides file.

`<module type>-<last subcomponent type or *>` identifies a subcomponent type (For example: EM-PO for a DPN-100 port normally identified as EM <> PE <> PI <> PO <>). If the subcomponent portion is specified as a single asterisk (*), the mapping applies to all subcomponents of this module whose criticality is not mapped by other means.

`<link type or *>` identifies a link type (For example: PTK, for a Passport trunk). If an asterisk is used (as in *:) the mapping applies to all otherwise unspecified link types.

> ***Note:*** All types are specified in upper case.

`criticality value` is a number between 0 and 255. The higher the number the greater the component’s criticality.

Customized component criticality overrides file

The component criticality overrides file (/opt/MagellanNMS/cfg/GMDRCritOverrides.cfg) contains specific component criticality mappings in the following format:

```
Component: <component name>
Criticality: <criticality value>
```

where:

`component name` is the name of the component whose criticality is being specified. The component name must be specified in uppercase letters. For modules, it is specified in the usual way with space separators (For example: PM TOTO PE 3 PI 3 PO 1). For links, it is also specified in the usual way with space and colon separators (For example: NL:PM TOTO PE 3 PI 3 PO 1:PM TITI PE 5 PI 5 PO 2:).

`criticality value` is a number between 0 and 255. The higher the number the greater the component's criticality.

> ***Note:*** Criticality auto-adjustment mode is always forced while the customized component criticality overrides file is being loaded. There is therefore no need to specify all the components in a hierarchy to raise the bottom one's mapping. GMDR automatically adjusts the mapping of the parent as needed. However, when you specify a link in the customized component criticality overrides file, you also need to specify immediately before the link entry, the endpoints of the link. The endpoints must be overridden to the same extent as the link.

Interdependencies

The GMDR server is always required by and relies on the FMDR, NMDR, IMDR, DMDR, SMDR, and subordinate GMDR servers for surveillance data.

Exit codes

Exit codes for the GMDR server are shown in the following table.:

Table 23
Exit codes for the GMDR server

Exit code	Description
51	Out of memory
55	Bad arguments on command line
59	Could not initialize IPC system
60	Could not register service (already running?)
62	Terminated due to licencing problem (see MDM Logs). Look at the System Log Display tool for logs about licencing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.

Error messages

Error messages for the GMDR server are shown in the following table.

Table 24
Error messages for the GMDR server

Error message	Meaning and action
GMDR - Could not open configuration file. GMDR - Configuration file read error. GMDR - Configuration file write error.	Fatal, could not access configuration file. Check the access modes on /opt/MagellanNMS/cfg/GMDR*.cfg.
GMDR - Invalid server specification. GMDR - Circular server specification. GMDR - Duplicate server specification.	Fatal, error in the configuration. Revise /opt/MagellanNMS/cfg/GMDR*.cfg.
(Sheet 1 of 3)	

Table 24 (Continued)
Error messages for the GMDR server

Error message	Meaning and action
GMDR - Could not load Criticality Schema file /opt/MagellanNMS/cfg/GMDRCriticality.cfg	Fatal, error in the Component Criticality configuration. Revise the configuration and access modes.
GMDR - Could not load Criticality Override file /opt/MagellanNMS/cfg/GMDRCritOverrides.cfg	
GMDR - Syntax error at/around line <> in Criticality Schema file /opt/MagellanNMS/cfg/GMDRCriticality.cfg	
GMDR - Syntax error at/around line <> in Criticality Override file /opt/MagellanNMS/cfg/GMDRCritOverrides.cfg	
GMDR - Invalid command line argument.	Fatal, invalid command line arguments. Revise the server configuration with the Server Administration tool.
GMDR - Alarm size value out of range.	
GMDR - Alarms per component out of range.	
GMDR - <server name> : invalid password or userid.	Non-fatal, could not connect to server due to authentication. Revise the server configuration in GMDR Administration tool
GMDR - <server name> : lost connection.	Non-fatal, communication problems with the server.
GMDR - <server name> : reconnected.	
GMDR - <server name> : communications error.	
GMDR - Communications error: passthrough request.	
GMDR - <client name> : killed connection due to congestion.	Non-fatal, client is too slow for query or query too wide.
GMDR - active alarm discarded, alarm buffer full.	
GMDR - active alarm discarded, alarm buffer full for <client name>.	

(Sheet 2 of 3)

Table 24 (Continued)
Error messages for the GMDR server

Error message	Meaning and action
GMDR - Cannot allocate licensing context	The GMDR server licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
GMDR - License refused: <reason>	A run-time license cannot be allocated to the GMDR server for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running GMDR.
GMDR - License warning: <reason>	The license required to run GMDR is about to expire. Contact your System Administrator immediately.
GMDR - License not confirmed: <reason>	There is a problem with the license required to run GMDR, it has most likely expired. Contact your System Administrator immediately.
GMDR - active alarm discarded, alarm buffer full.	The alarm buffer is filled with active alarms; discard the incoming alarm if it is not a SET alarm otherwise discard the oldest alarm in the buffer.
GMDR - <client name> : killed connection due to congestion.	The specified client is discarded.
GMDR - server could not create STATUS sieve : <reason>	CREATE sieve request rejected.
GMDR - server could not delete STATUS sieve : <reason>	DELETE sieve request rejected.
GMDR - server status event notification communications error.	Sieve deleted.
(Sheet 3 of 3)	

Chapter 16
Host Group Directory Server (HGDS)

This section contains information on the Host Group Directory server (HGDS). See the following topics for more information:

About the HGDS server

HGDS provides information to Preside Multiservice Data Manager (MDM) that describes how nodes in the network are grouped. How nodes are grouped is part of the system configuration. For information about groups, see

- configuring servers for DPN servers in 241-6001-303 *Preside MDM Administrator Guide* for more information about OA groups
- configuring servers for Passport switches in 241-6001-303 *Preside MDM Administrator Guide* for more information about Passport groups
- configuring servers for MPE 9500 switches in 241-6001-303 *Preside MDM Administrator Guide* for more information about MPE 9500 groups

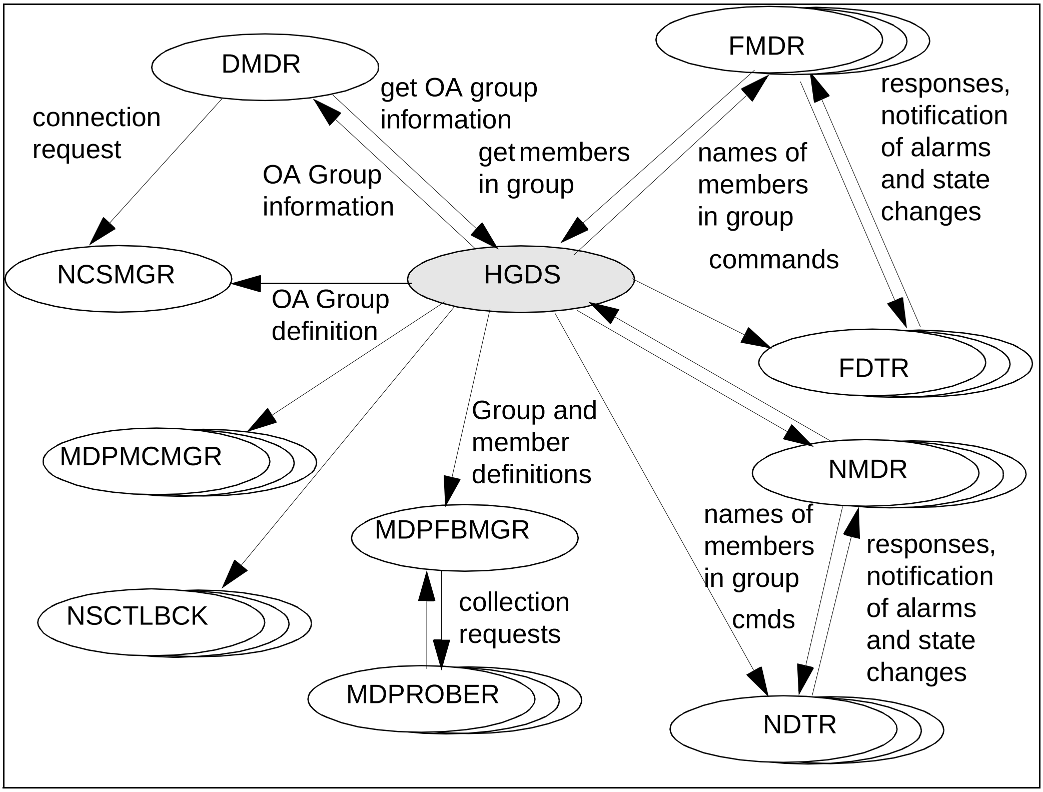
For Passport, the HGDS maps groups to their member hosts, host names to their IP addresses, and provides this information to the FMDR server and to the FDTR process. The FMDR server and FDTR process use this information to login to nodes and manage the connections to them.

For MPE 9500 access, the HGDS maps groups to their member hosts, host names to their IP addresses, and provides this information to the NMDR server and to the NDTR process for MPE 9500. The NDTR process use this information to login to nodes and manage the connections to them.

For DPN access, the HGDS server defines the X.25 access parameters for each OA. It also groups OAs for use by the DMDR server for surveillance. The data for the mappings is provided in the HGDS Information file (/opt/MagellanNMS/cfg/HGDS.cfg).

The information in the file /opt/MagellanNMS/cfg/HGDS.cfg is made active by restarting the HGDS server with the Server Administration tool. The related servers DMDR, FDTM, FMDR, NCSMGR, NDTM, NMDR, NSCTLBCK, MDPFPMGR and MDPCMMGR also need to be restarted for the information in the file /opt/MagellanNMS/cfg/HGDS.cfg to be made active. For the instructions to configure these servers and start them in the correct order, see the roadmap to MDM servers in 241-6001-303 *Preside MDM Administrator Guide*.

Figure 21
HGDS data flow diagram



HGDS server information file

File /opt/MagellanNMS/cfg/HGDS.cfg must reside on each workstation that is running the HGDS before Preside Multiservice Data Manager system initialization. The file must be updated on all workstations on which it resides if any changes are made to the switch IP addresses, OA configurations, or groupings. The recommend way to update the HGDS.cfg file is through the Host Group Administration application. For more information, see the 241-6001-303 *Preside MDM Administrator Guide*.

For more information, see the following:

- “File format” (page 180)

- “The Passport definition section” (page 180)
- “The SRS definition section” (page 181)
- “The OA definition section” (page 181)
- “The Passport definition section” (page 182)
- “The SRS group definition section” (page 183)
- “The OA group definition section” (page 183)

File format

File /opt/MagellanNMS/cfg/HGDS.cfg is divided into two sections. The first section contains records that list the host name and corresponding IP address of each Passport node in the network, or OA access information. The second section contains records that list the group names and their corresponding member host names.

> ***Note:*** The member definition section must precede the group definition section.

The file records are in ASCII format and contain two or more fields. Each line in the file may contain only one field, so that each record may take up several lines in the file. Each record is followed by a blank line. Comments are preceded by an asterisk (*).

The Passport definition section

This section consists of a series of records that map a Passport host name to its IP address. The records in this section have the following format:

```
FMember: <hostname>
IPAddress: <IP_address>
```

where:

`hostname` is an uppercase character string 1 to 12 characters long that is the official host name of the node, as stored in the service data for the node.

`IP_address` is the Internet Protocol (IP) address assigned to the node during system configuration. The IP address consists of four numerical fields, 1 to 3 digits long, separated by periods. An example of an IP address is 10.125.40.3.

The SRS definition section

This section consists of a series of records that map an SRS hostname to its IP address. The records in this section have the following format:

```
SRSMember: <srs_hostname>
IPAddress: <IP_address>
```

where:

`srs_hostname` is an uppercase character string 1 to 20 characters long that is the official host name of the SRS node, as stored in the service data for the node.

`IP_address` is the Internet Protocol address assigned to the SRS node during system configuration. The IP address consists of four numerical fields, 1 to 3 digits long, separated by periods. An example of an IP address is 10.125.40.3.

The OA definition section

This section of the file consists of a series of records that maps the DPN host name to its DNA, and that define the communications parameters required to access the OA member. The OA member is also called the destination mnemonic.

The records in the OA definition section have the following format:

```
OAMember: <OA (MDIMDI) name>
Name: <OA name>
DNA: <MDI access DNA>
CUG: <MDI access CUG index>
PktSz: <MDI packet size>
X75: <Y if calls can go over X.75 otherwise N>
RPOA: <the RPOA number (optional)>
```

where:

`OAMember` is a name used to identify the connection on the management data interface (MDI) of the operations agent (OA). It is an uppercase character string 1 to 12 characters long that is the official host name of the DPN node, as stored in the service data for the node.

NAME is the name of the operations agent on which the MDI is located. It is the mnemonic for the operations agent consisting of up to 12 alphanumeric characters.

DNA is the data network address used to access the MDI. It is a valid DN consisting of up to 15 decimal digits.

CUG is the index number of the closed user group (CUG) to which the WS-MDI belongs. It is a valid two-digit CUG index.

PktSz is the default size for data packets transmitted between the workstation and the MDI. It is 128, 256 or 512. The default is 256.

X75 is the indication (y or n) that calls pass over an X.75 connection.

RPOA is the number used to identify a Recognized Private Operations Agency. This is only used for X.75 connections.

The Passport definition section

This section of the file consists of a series of records that map Passport nodes to a group. The records in this section have the following format:

```
FGroup: <group_name>
Member: <hostname>
Member: <hostname>
Member: <hostname>
   .
   .
Member: <hostname>
```

where:

group_name is an uppercase character string 1 to 12 characters long. A group name must be unique system-wide. If the name has more than one word separate the words with an underscore.

hostname is an uppercase character string 1 to 12 characters long that is the official host name of the node, as stored in the service data for the node and defined as a member in the group definition section of the file.

The SRS group definition section

This section of the file consists of a series of records that map the SRS members to an SRS group. The records in this section have the following format:

```
SRSGroup: <group_name>
Member: <srs_hostname>
Member: <srs_hostname>
Member: <srs_hostname>
   .
   .
Member: <hostname>
```

where:

`group_name` is an uppercase character string 1 to 12 characters long. A group name must be unique system-wide. If the name has more than one word separate the words with an underscore.

srs_`hostname` is an uppercase character string 1 to 20 characters long that is the official host name of the SRS node, as stored in the service data for the node and defined as a member in the SRS definition section of the file.

The OA group definition section

This section of the file consists of a series of records that maps OA members to an OA group. The records in this section have the following format:

```
OAGroup: <OA group name>
Member: <OA member name as above>
Member: <OA member name as above>
   .
   .
Member: <OA member name as above>
```

where:

`OAGroup` is the name of the group to which an OA member belongs. It is an uppercase character string 1 to 12 characters long. A group name must be unique system-wide. If the name has more than one word, separate the words with an underscore.

`Member` is a member name as defined in the field OAMember, at the beginning of this file. A group can have more than one member.

Example

The figure “A sample HGDS information file” (page 184) shows a sample HGDS Information file. The sample file shows one group called ALLOA. This group contains three members: TOP, EUROPE and AMERICA.

Figure 22
A sample HGDS information file

```
OAMember: TOP
NAME: TOP
DNA: 12345678
CUG: 03
PktSz: 510
X75: N

OAMember: EUROPE
NAME: EUROA
DNA: 18234567
CUG: 03
PktSz: 512
X75: Y
RPOA:1234

OAMember: AMERICA
NAME: AMOA
DNA: 18723456
CUG: 03
PktSz: 512
X75: N


OAGroup: ALLOA
Member: TOP
Member: EUROPE
Member: AMERICA
```

Managing the HGDS server

Use the Server Administration tool to edit the HGDS server entry to enter the startup command, set the server options, and to specify that the server should be started automatically at reboot time. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

For more information, see the following:

- "Suggested name in Server Administration" (page 185)
- "Startup command" (page 185)

Suggested name in Server Administration

The recommended name for the HGDS server is Host Group Directory.

Configuring HGDS with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the HGDS server has the following syntax:

```
/opt/MagellanNMS/bin/hgds [-cfg <HGDS_config_file>]
```

where:

`-cfg <HGDS_config_file>` is the complete path of the HGDS configuration file. If you do not specify a file path, Preside Multiservice Data Manager uses the file /opt/MagellanNMS/cfg/HGDS.cfg.

Enter the startup command in the Startup command field of the Server Administration dialogs of the Server Manager Administration tool. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to enter the server startup command.

Interdependencies

The HGDS server does not rely on any other optional Preside Multiservice Data Manager servers.

Exit codes

HGDS also outputs useful diagnostic messages (for example, messages warning of syntax errors in the configuration file) to the System Log Display. Exit codes for the HGDS server are shown in the following table.

Table 25
Exit codes for the HGDS server

Exit code	Description
0	Successful exit. Do not restart the server.
1	Failure. Restart the server.
2	Internal error.
3	Could not register service name (already running?).
50	Do not restart (for backward compatibility)
51	Memory resource error (For example, not enough memory). No restart.
52	Disk resource error (For example. not enough space, no file). No restart.
53	Communication resource error (For example, cannot connect or register). No restart.
54	Timeout/deadlock/congestion (For example, too much congestion). No restart.
55	Bad command line arguments. No restart.
56	Bad configuration file or environment. No restart. See the OAM logs for more information
57	Fork exec failure. No restart.
58	Manual server shutdown (admin?). No restart.
59	ipc_init unsuccessful. No restart.
60	ipc service cannot be registered. No restart.
61	Exit signal received. No restart.
100	Other unclassified errors. Restart the server.
D:\Profiles\dmcken1\My Documents\WebWorks\FrameSource\6_310\hgds.fm	

Error messages

Error messages for the HGDS server are shown in the following table.

Table 26
Error messages for the HGDS server

Error message	Meaning and action
HGDS: Could not allocate node lists. HGDS: Could not locate configuration file. HGDS: Could not allocate attribute. HGDS: Could not allocate node.	Fatal, internal errors (memory), the server will restart.
HGDS: Bad command line argument.	Fatal, invalid command line arguments. Revise server configuration with the Server Administration tool.
HGDS: Could not register service.	Fatal, could not register the service name. Server may already be running. Revise server configuration with the Server Administration tool
HGDS: Could not open configuration file.	Fatal, configuration file cannot be opened. Check command line and file access modes (/opt/MagellanNMS/cfg/HGDS.cfg).
HGDS: Bad value in configuration file (line = <line no>). HGDS: Missing attribute in configuration file (line = <line no>). HGDS: Entry out of order in configuration file (line = <line no>).	Fatal, syntax error in the configuration file correct at the indicated line (or just above).
(Sheet 1 of 2)	

Table 26 (Continued)
Error messages for the HGDS server

Error message	Meaning and action
HGDS: Bad member reference in configuration file (line = <line no>).	
HGDS: Group without members in configuration file (line = <line no>).	
HGDS: Unknown entry in configuration file (line = <line no>).	
HGDS: No members defined in configuration file (line = <line no>).	
HGDS: Too many members for a group in configuration file (line = <line no>).	
HGDS: Group name too long in configuration file (line = <line no>).	
HGDS: Duplicate Group defined in configuration file (line = <line no>).	
HGDS: Duplicate Host defined in configuration file (line = <line no>).	
(Sheet 2 of 2)	

Chapter 17
Injected Management Data Router (IMDR)

This section contains information on the Injected Management Data Router (IMDR). See the following topics for more information:

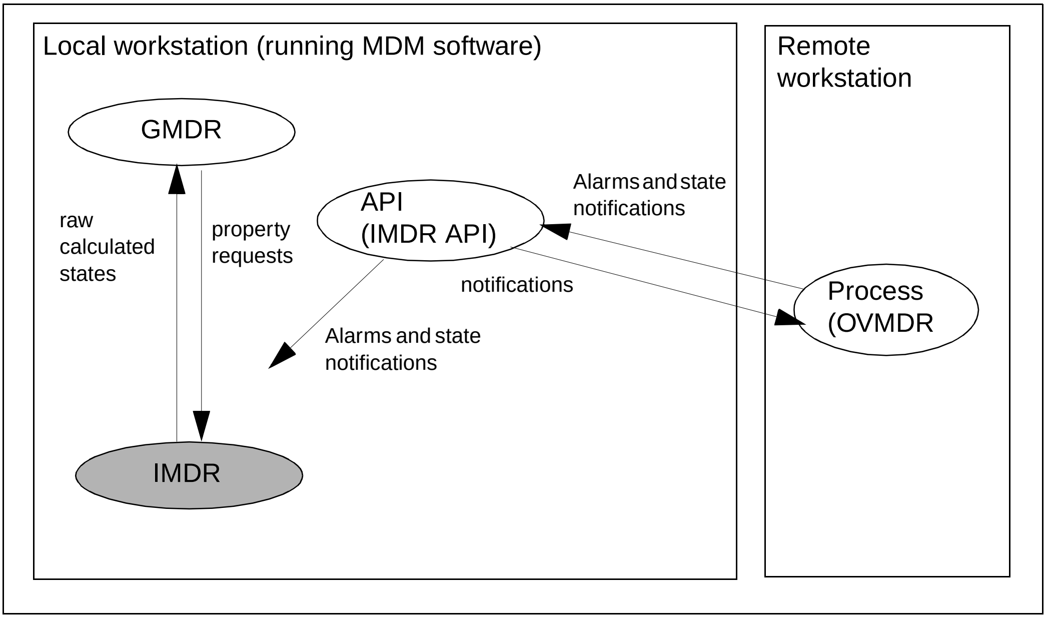
About the IMDR server

The IMDR server performs the following functions:

- collects surveillance data including alarms and state notifications that are injected into it from other surveillance processes though the IMDR API.
- stores alarms and state notifications in its internal database
- calculates a raw state for each component in the internal database
- updates the raw state each time an alarm or a state notification is received

- provides the calculated raw state to clients, such as GMDR. The raw state (or external state or state from the network) is the state computed by the Surveillance Data Servers (GMDR, NMDR, DMDR, FMDR, IMDR), for the component on the basis of network management information received for the component. Possible raw state values are:
  - UNK (unknown): Preside Multiservice Data Manager (MDM) has not heard from the component and does not know its state.
  - INSV (in-service): The component is known to be working properly.
  - OOS (out-of-service): The component is not working.
  - TRB (troubled): The component is known to be in-service, but experiencing some difficulties. For example, the component is overloaded.
  - NEX (non-existent)
    For details on how IMDR determines the raw state see “State calculation” (page 192). IMDR provides the raw state to clients (for example, GMDR, IMDR API clients) in the following ways:
  - raw state change notifications
  - alarms (raw state is embedded in alarms)
- supports property requests from clients, such as GMDR

Figure 23
IMDR data flow diagram



IMDR API

The IMDR server supports an applications programming interface (API) which provides a set of special services for client processes to inject alarms and state notifications into IMDR. These special services are as follows:

- REGISTER requests for injection clients are equipped with a mandatory userCapability attribute with a value of mdInject to identify the client as an injection client. The request can also contain a userDiscClean attribute to indicate whether data injected by the client must be deleted should the client be disconnected.
- CREATE sieves are provided for local clear and property request notifications. These notifications replace the corresponding ACTION requests that IMDR would send to its subservers.
- ACTION requests are provided for injection. These include: inject alarm, inject raw state, and inject property.

For information about the services provided by the IMDR API, see 241-6001-203 *Preside MDM Alarm and Status API Reference Guide*.

State calculation

IMDR calculates a raw state for each component stored in its internal database. The raw state is based on the list of active alarms for a component, as follows:

- If there are no active alarms for a component, its raw state is calculated as INS (in service)
- If all the active alarms for a component have a severity of WARNING, the raw state is calculated as TRB (troubled)
- In all other cases the raw state is calculated as OOS (out of service)

IMDR updates the raw state each time an alarm or a state change is received.

Raw state information can also be injected to IMDR. Raw state injection directly specifies the raw state of a component.

If the injected raw state does not agree with the raw state calculated from the active alarms stored against a component IMDR’s internal database, IMDR raises a proxy alarm. For information about how IMDR deals with injected raw states that do not match active alarms stored in its own internal database, and proxy alarms raised by IMDR, see 241-6001-501 *Preside MDM Alarms Reference Guide*.

If the injected raw state is worse than the calculated raw state, a proxy SET alarm is created. If the injected raw state is better than the calculated raw state, a proxy CLEAR alarm is created which clears some or all of the active alarms in IMDR’s internal database, so that the calculated raw state agrees with the injected raw state.

Managing the IMDR server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for IMDR. Any changes you make to the startup command or options take effect when IMDR is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

For more information, see the following:

- “Suggested name in Server Administration” (page 193)
- “Startup command” (page 193)

Suggested name in Server Administration

The recommended name for the IMDR server is IMDR for the first IMDR server on the workstation. Additional IMDR servers should be named IMDR_<suffix>. For example: IMDR_EAST.

Configuring IMDR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start an IMDR server has the following syntax:

```
/opt/MagellanNMS/bin/imdr [-a] [-B] [-n <suffix>] \
[-b <buffer size>]
```

where:

`-a` prevents active alarms from being fully recorded in the IMDR server’s internal database. When this option is specified, approximately 1 Kbtye less memory is used for each alarm. A disadvantage to specifying this option is that IMDR does not supply GMDR with an active alarm list when a connection to the GMDR is established.

If this option is not specified, the IMDR server fully records active alarms by default. The -B option overrides the -a option. So, if you specify the -a option and the -B option, the -a option is ignored

`-B` specifies that babbler alarms are to be forwarded to the GMDR server. A babbler alarm has the same component ID, fault code, and severity as an active alarm, but is issued at a later time.

If this option is not specified, babbler alarms are discarded by default.

`-n <suffix>` specifies a suffix to be added to the IMDR server name. This option is required to uniquely identify an IMDR server when more than one IMDR server is running on the same workstation. When this option is specified, the server name becomes IMDR followed by an underscore, followed by the suffix. For example, IMDR_EAST.

If -n is specified <suffix> must also be specified. If this option is not specified, the server name defaults to IMDR.

`-b <buffer size>` specifies the number of messages that can be stored before congestion occurs. If buffer size is not specified, the default value of 1000 is used.

Example: Start an IMDR server called IMDR_EAST:

```
/opt/MagellanNMS/bin/imdr -n EAST
```

Interdependencies

The GMDR server must be set up with the GMDR Administration tool to receive information injected into the IMDR server.

Configuration

There are no configuration files to set up for the IMDR server.

Exit codes

When a server fails, an exit code is displayed in the message area of the Server Administration tool. The reason for the failure is displayed in the System Log Display and logged in the Preside Multiservice Data Manager log. Exit codes for the IMDR server are shown in the following table.

Table 27
Exit codes for the IMDR server

Exit code	Description
51	allocation of memory to construct a new data object has failed
55	the command line contains an invalid argument
59	this process failed its registration (ipc_init) with IPC

Error messages

There are two categories of error messages for the IMDR server:

- error messages displayed in the System Log Display

  These messages are shown in the table “Error messages from IMDR that are sent to the System Log Display” (page 195).

- error messages returned to the client application such as GMDR that makes a request to IMDR through the IMDR API.

  These messages are shown in the table “Error messages from IMDR in response to a client request” (page 196).

Table 28
Error messages from IMDR that are sent to the System Log Display

Error message	Meaning and action
IMDR -- Invalid command line argument	One of the command line arguments is invalid.
IMDR -- Options -a and -B are incompatible	Partial alarm recording and babbler alarm transmission are incompatible options; the -a option is ignored.
<IMDR name> -- Killed client connection due to congestion	Connection has been dropped to a client process because of congestion.
<IMDR name> -- Unexpectedly lost client connection; error code: <code>	A protocol error identified by <code> has caused a client connection to be dropped.

Table 29
Error messages from IMDR in response to a client request

Error message	Meaning and action
ACCESS_DENIED Not sieve owner	A client has attempted to delete or modify a sieve created by another client.
SYNTAX_ERROR	IMDR has received a request with an invalid command name.
INVALID_ACTION_TYPE	IMDR has received an ACTION request with an invalid action type.
INVALID_ATTRIBUTE_ NAME	IMDR has received a CREATE request with an invalid attribute name.
INVALID_ATTRIBUTE_ VALUE <attribute>	IMDR has received a CREATE request with an attribute containing an invalid value.
INVALID_ATTRIBUTE_ VALUE compId attribute name	The component ID attribute in a component delete ACTION request has no value.
INVALID_ATTRIBUTE_ VALUE eventFilter attribute name	A filter present in a CREATE request uses an attribute name on which filtering is not supported.
INVALID_ATTRIBUTE_ VALUE eventFilter attribute type	A filter present in a CREATE request uses a value type that is not supported.
INVALID_ATTRIBUTE_ VALUE repFilter attribute name	An attribute used in a repFilter expression of a CREATE request is not supported.
INVALID_ATTRIBUTE_ VALUE repFilter attribute value	The value type specified in a repFilter expression of a CREATE request is invalid.
INVALID_SCOPE	IMDR has received a request with an invalid scope.
INVALID_OBJECT_ CLASS	IMDR has received a request with an invalid object class.
INVALID_OBJECT_ID	IMDR has received a request with an invalid object ID.
(Sheet 1 of 3)	

Table 29 (Continued)
Error messages from IMDR in response to a client request

Error message	Meaning and action
MISSING_ATTRIBUTE <attribute>	A required attribute is missing from a CREATE request.
MISSING_ATTRIBUTE_ VALUE	A required attribute is missing in an ACTION request received by IMDR.
MISSING_ATTRIBUTE_ VALUE component ID	The component ID is missing in a property request ACTION received by IMDR.
MISSING_ATTRIBUTE_ VALUE property name	The property name is missing in a property request ACTION received by IMDR.
NO_SUCH_OBJECT_ID	The object ID specified in a request received by IMDR does not exist.
OUT_OF_SEQUENCE REGISTER required	The request received by IMDR cannot be processed before the client has registered.
OUT_OF_SEQUENCE DEREGISTER required	The client is already registered and must deregister or disconnect before registering again.
NOT_SUPPORTED Filter operator	An operator used in a filter expression is not supported.
NOT_SUPPORTED obj_id	A CREATE request contains an object ID (not supported).
NOT_SUPPORTED sup_obj_id	A CREATE request contains a superior object ID (not supported).
NOT_SUPPORTED ref_obj_id	A CREATE request contains a reference object ID (not supported).
APPLICATION_ERROR Multiple eventType eventFilter	A filter present in a CREATE request uses several event types (not supported).
APPLICATION_ERROR Missing eventType eventFilter	A CREATE request does not identify the event type for which it is intended.
(Sheet 2 of 3)	

Table 29 (Continued)
Error messages from IMDR in response to a client request

Error message	Meaning and action
APPLICATION_ERROR Maximum number of sieves reached	A new sieve cannot be created because the maximum number of sieves has already been reached.
APPLICATION_ERROR Cleared alarm not found	The alarm specified in a local clear ACTION request cannot be found.
APPLICATION_ERROR Action not allowed	An ACTION request is received from a client which is not allowed to make such requests.
APPLICATION_ERROR Component does not exist	A property request ACTION has been received for a component that is not present in IMDR component tree.
APPLICATION_ERROR invalid or missing parameters	Required parameters in an ACTION request are missing or invalid.
APPLICATION_ERROR no process to query	A property request ACTION cannot be satisfied since the component specified is not monitored by an injecting client supporting these requests.
APPLICATION_ERROR admState attribute duplicated	The administrative state attribute is specified more than once in a CREATE request.
APPLICATION_ERROR internal error	A request cannot be satisfied because of an internal IMDR error (should not happen).
(Sheet 3 of 3)	

Chapter 18
Log Collector (OAMC)

This section contains information on the Log Collector (OAMC). See the following topics for more information:

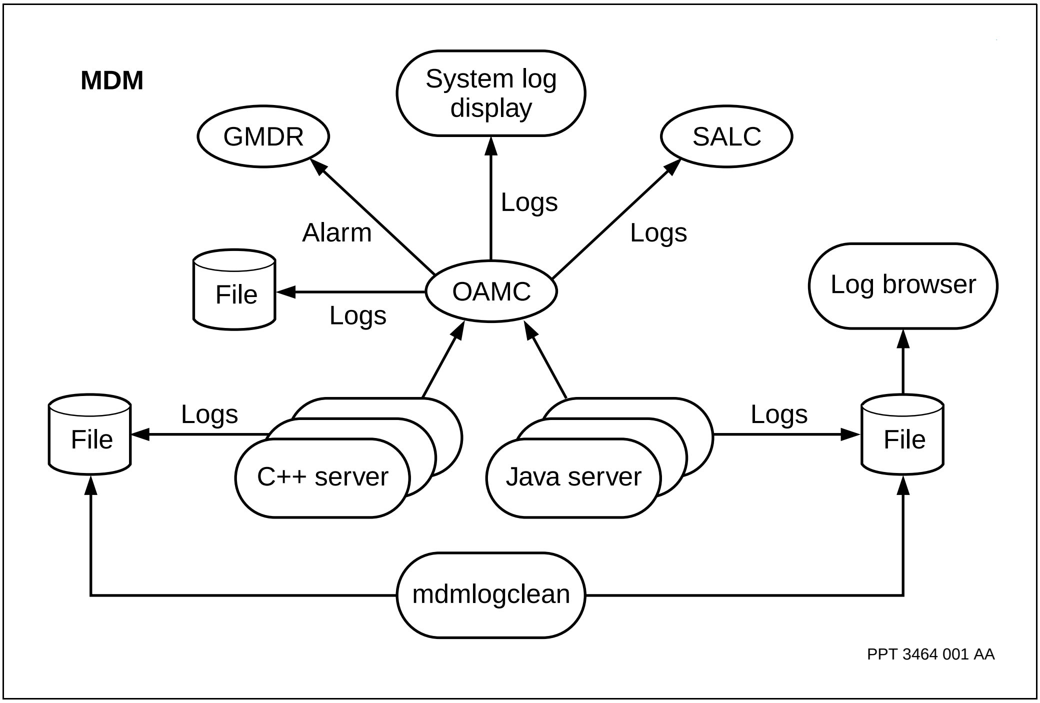
About the OAMC server

The OAMC server is the Preside Multiservice Data Manager (MDM) server that collects logs generated by MDM (see “OAMC data flow diagram” (page 200). The OAMC also has the following functions:

- sends logs to file
- sends alarms to the GMDR server
- makes logs available to the System Log Display tool (see 241-6001-303 *Preside MDM Administrator Guide* for more information on using the System Log Display tool).
- forwards security-related logs to the Security Audit Log Collector (SALC) server (see “Security Audit Log Collector (SALC)” (page 461))
- collects and writes MDM audit events to file

If the OAMC server cannot start, the OAMC writes a log message.

Figure 24
OAMC data flow diagram



Audit events

Audit events audit Preside Multiservice Data Manager workstations and allow the operator to audit the activities of users. Additionally, these logs record tool usage by NetRx. Logs are written to /opt/MagellanNMS/data/logs/oamc/audit.nlog according to the system-wide level set defined in /opt/MagellanNMS/lib/cfg/OAMLog.cfg.

Log cleanup

A log cleanup process, mdmlogclean, removes log files after a certain number of days. The mdmlogclean is run from the command line as the root user. For more information, see 241-6001-303 *Preside MDM Administrator Guide*.

Managing the OAMC server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the OAMC server. Any changes you make to the startup command or options take effect when OAMC is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 201)
- “Startup command” (page 201)

Suggested name in Server Administration

The recommended name for the OAMC server is Log Collector.

Configuring OAMC with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The OAMC runs on every instance of Preside Multiservice Data Manager and serves as the common location where logs are sent. The startup command for the OAMC server has the following syntax:

```
/opt/MagellanNMS/bin/oamc
[-help]
[-history <nb logs>]
[-outputFile <log level set>] \
[-logFile <log level set>] \
```

Variable definitions

Variable	Value
-help	provides usage information.
-history	is the number of received logs to keep in memory. The default is 500.
-outputFile [<log level set]	is used to override settings for writing application logs to /opt/MagellanNMS/data/log/oamc/output.alog as a daily rolling log that is readable by all, but writable only by the owner of the OAMC process, which is usually the root user. The <log level set> can be set to one or more of the following: [NOTICE, CLEARED, WARN, ERROR, CRIT, ALERT, FATAL]. By default, logs are written according to the OAMLog.cfg file content.
-logFile [<log level set>]	is used for writing application logs that the OAMC server generates for itself. Logs are written to /opt/MagellanNMS/data/log/oamc.alog The <log level set> can be set to one or more of the following: [TRACE, DEBUG, INFO, NOTICE, CLEARED, WARN, ERROR, CRIT, ALERT, FATAL]. By default, OAMC logs are written to file with the given level set FATAL, ALERT, CRIT, CLEARED, NOTICE.

Configuring options for the OAMC server

Use the following procedures to configure options for the OAMC server:

- “Configuring the system-wide audit level” (page 204)
- “Configuring the system-wide log level” (page 205)

Configuring the system-wide audit level

Use this procedure to change the level of Preside Multiservice Data Manager audit logs that are written to /opt/MagellanNMS/data/log/oamc/audit.nlog from the default.

Prerequisites

- You must be logged in as the root user.

Procedure steps

1 Copy /opt/MagellanNMS/lib/cfg/OAMLog.cfg file to /opt/MagellanNMS/cfg/OAMLog.cfg, if it doesn’t already exist.

2 Edit the /opt/MagellanNMS/cfg/OAMLog.cfg file to change the default level of logs sent to OAMC:

```
Example:
```

```
systemWideAuditLevel WARN,NOTICE,CRIT,FATAL,ALERT
```

Variable definitions

Variable	Value
systemWideAuditLevel	is the system level of audit-level logs that are written to /opt/MagellanNMS/data/log/oamc/audit.nlog. This parameter can have the following levels in the order of increasing severity: NOTICE, CLEARED, WARN, ERROR, CRIT, ALERT, FATAL.

Configuring the system-wide log level

Use this procedure to change the defaults in the /opt/MagellanNMS/cfg/OAMLog.cfg for the level of logs that all Preside Multiservice Data Manager servers send to the OAMC server.

Prerequisites

- You must be logged in as the root user.

Procedure steps

1 Copy /opt/MagellanNMS/lib/cfg/OAMLog.cfg file to /opt/MagellanNMS/cfg/OAMLog.cfg, if it doesn’t already exist.

2 Edit the /opt/MagellanNMS/cfg/OAMLog.cfg file to change the default level of logs to be sent to OAMC:

```
Example:

systemWideLogLevel: WARN, NOTICE, CRIT, FATAL, ALERT
```

Variable definitions

Variable	Value
systemWideLogLevel	is the level of logs that all MDM servers send to the OAMC. The /opt/MagellanNMS/lib/cfg/OAMLog.cfg lists the levels in order of increasing severity: NOTICE, CLEARED, WARN, ERROR, CRIT, ALERT, FATAL.

Exit codes

Exit codes for the OAMC server are shown in the following table:

Table 30
Exit codes for the OAMC server

Exit code	Description
50	Miscellaneous errors
51	...memory failure
52	Unable to write oamc log file
53	Communication resource error
55	Bad arguments provided in command line
56	Bad dictionary file encountered
59	Unable to initialize IPC with name
60	Communication resource error

Error messages

Error messages for the OAMC server are shown in the following table.

Table 31
Error messages for the OAMC server

Error message	Meaning and action
Could not create OAM Collection service	Fatal, could not register the service. MNSD is not up or server is already running.
Could not open log file “<file name>”	Fatal, could not open log file. Check the file’s access mode.

Chapter 19
GMDR Agent (GMDRAGENT)

This section contains information about the GMDR Agent (GMDRAGENT) . See the following sections for information about the agent:

- “About the GMDRAGENT” (page 207)
- “Managing the GMDRAGENT” (page 207)
- “Interdependencies” (page 208)

About the GMDRAGENT

Operator Client software is an add-on package to Preside Multiservice Data Manager software. Operator Client allows users on a PC or on a UNIX workstation to perform fault management operations on devices in the network and to run commands on the devices.

The GMDRAGENT provides an interface between the Operator Client and the GMDR server to access alarm information, alarm counts, and status counts for devices in the network.

If the network is partitioned into managed regions, the GMDRAGENT can be configured to collect information from the devices in a region from the NDAM server instead of the GMDR server.

Managing the GMDRAGENT

See the following sections for information to manage the Operator Client Command Access Agent:

- “Configuration” (page 208)

- “Suggested name in Server Administration” (page 208)
- “Start-up command” (page 208)

Configuration

Installing the Operator Client software with the MDM Installer, as described in 241-6001-100 *Preside MDM Installation*, automatically adds the information needed to configure and start the GMDRAGENT when the workstation reboots.

Suggested name in Server Administration

The recommended name for the GMDRAGENT to enter in the Server Manager Administration took is GMDR Agent.

Configuring the GMDRAGENT with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the GMDRAGENT has the following syntax:

```
/opt/MagellanNMS/bin/gmdragent
```

Interdependencies

The GMDRAGENT requires that the Preside Multiservice Data Manager base software be installed and configured, and that the following servers are configured and running:

- GMDR or NDAM.
- GMDRAGENT obtains information from a GMDR server in networks that are not partitioned into separately managed regions, or from the NDAM server in networks that are partitioned.

Chapter 20
NM Agent (NMAGENT)

This section contains information about the Network Model Agent (NMAGENT). See the following sections for information about the agent:

- “About the NMAGENT server” (page 209)
- “Managing the NMAGENT” (page 209)
- “Interdependencies” (page 210)

About the NMAGENT server

Operator Client software is an add-on package to Preside Multiservice Data Manager (MDM) software. Operator Client allows users on a PC or on a UNIX workstation to perform fault management operations on devices in the network and to run commands on the devices.

The NMAGENT provides an interface between the Operator Client and MDM Network Model server to provide information about the states of components and devices in the network.

Managing the NMAGENT

See the following sections for information to manage the Command Access Agent:

- “Configuration” (page 210)
- “Suggested name in Server Administration” (page 210)
- “Start-up command” (page 210)

Configuration

Installing the Operator Client software with the MDM Installer, as described in 241-6001-100 *Preside MDM Installation*, automatically adds the information needed to configure and start the NMAGENT when the workstation reboots.

Suggested name in Server Administration

The recommended name for the NMAGENT to enter in the Server Manager Administration tool is NM Agent.

Configuring the NMAGENT with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information to the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the NMAGENT has the following syntax:

```
/opt/MagellanNMS/bin/nmagent
```

Interdependencies

The NMAGENT requires that the Preside Multiservice Data Manager base software is installed and configured, and that the NMSERVER server is configured and running.

Chapter 21
RTAC agent (RTCAGENT)

This section contains information about the Real Time Alarm (RTAC) Collection Agent (RTCAGENT). See the following sections for information about the agent:

- “About the RTACAGENT server” (page 211)
- “Managing the RTCAGENT” (page 211)
- “Interdependencies” (page 212)

About the RTACAGENT server

Operator Client software is an add-on package to Preside Multiservice Data Manager software. Operator Client allows users on a PC or on a UNIX workstation to perform fault management operations on devices in the network and to run commands on the devices.

The RTCAGENT provides an interface between the Operator Client and the real time alarm collection (RTAC) server to access archived information about alarms for devices in the network.

Managing the RTCAGENT

See the following sections for information to manage the RTAC Collection Agent:

- “Configuration” (page 212)
- “Suggested name in Server Administration” (page 212)
- “Start-up command” (page 212)

Configuration

Installing the Operator Client software with the MDM Installer, as described in 241-6001-100 *Preside MDM Installation*, automatically adds the information needed to configure and start the RTCAGENT when the workstation reboots.

Suggested name in Server Administration

The recommended name for the RTCAGENT to enter in the Server Manager Administration took is RTAC Agent.

Configuring the RTCAGENT with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the RTCAGENT has the following syntax:

```
/opt/MagellanNMS/bin/rtacagent
```

Interdependencies

The RTCAGENT requires that the Preside Multiservice Data Manager base software is installed and configured, and that the following servers are configured and running:

- RTAC server

Chapter 22 MDP DPN File Collector (MDPCOL)

This section describes the information required to control the Management Data Provider (MDP) DPN-100 File Collector server using the Preside Multiservice Data Manager Server Administration (SVM) tool.

See the following sections of information about this server:

- About the MDPCOL server (page 213)
- Managing the MDPCOL server (page 213)
- Interdependencies (page 214)

About the MDPCOL server

The Management Data Provider (MDP) DPN-100 File Collector (MDPCOLl) controls the collection of DPN-100 data files from DPN-100 switches.

The DPN-100 switch must be configured as an MDP spooling site.

For more information about MDPCOL, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPCOL server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPCOL server. Any changes that you make to the startup command take effect when the MDPCOL server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPCOL can be configured using the MDP Configuration tool (gmdpconfig) or by editing the file /opt/MagellanMDP/cfg/mdp/MDPDPNMgr.cfg.

Suggested name in Server Administration

The recommended name for the MDPCOL server is **MDP DPN Collector n** where n corresponds to the file collector number.

Registering the MDPCOL server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPCOL server has the syntax of one of the following:

```
/opt/MagellanMDP/bin/mdpcol -pid <PID>

/opt/MagellanMDP/bin/mdpcol -pid <PID> -v

/opt/MagellanMDP/bin/mdpcol -pid <PID>
#<file_collector>

/opt/MagellanMDP/bin/mdpcol -pid <PID>
/opt/MagellanMDP/data/mdp/spool
```

where:
<PID> is a unique protocolID for each DPN-100 file collector
-v is verbose logging
<file_collector> is an integer. The first file collector should be numbered 1, the next file collector should be numbered 2, and so on.
/opt/MagellanMDP/data/mdp/spool is the destination directory for the collected DPN data

Interdependencies

If collected DPN-100 data files are to be converted to published format (PF) or bulk data format (BDF), the MDPCOL server requires that the MDP DPN-100 File Manager (mdpdpnmgr) be active.

Chapter 23
MDP DPN File Manager (MDPDPNMGR)

This section describes the information required to control the Management Data Provider (MDP) DPN File Manager server using the Preside Multiservice Data Manager Server Administration (SVM) tool.

See the following sections for information about this server:

About the MDPDPNMGR server

The Management Data Provider (MDP) DPN File Manager controls

- the conversion of DPN raw data to published format (PF) or bulk data format (BDF)
- the storage of converted data
- the creation of raw data backups

For more information about the MDPDPNMGR, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPDPNMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPDPNMGR server. Any changes that you make to the startup command take effect when the MDPDPNMGR server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPDPNMGR can be configured using the MDP Configuration tool (gmdpconfig) or by editting the file /opt/MagellanMDP/cfg/mdp/MDPDPNMgr.cfg.

Suggested name in Server Admininstration

The recommended name for the MDPDPNMGR server is **MDP DPN File Manager**.

Registering the MDPDPNMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPDPNMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpdpnmgr
```

Interdependencies

The MDPDPNMGR server requires that a MDP DPN Collector server be active to collect DPN data for conversion.

The MDPDPNMGR server requires that the MDP Disk Manager server be active to control MDP disk space.

If converted DPN data files are to be transferred to a customer host for downstream processing, the MDPDPNMGR server requires the MDP File Mover Manager server to be active.

Error Messages

MDPDPNMGR error messages are written to file
/opt/MagellanMDP/data/mdp/admin/
MDPDPNMgr(Sun|Mon|Tue|Wed|Thu|Fri|Sat)_YYYYMMDD.log.

MDPDPNMGR error messages are not written to the OAM Log Collector.

Table 32 describes the MDPDPNMGR error messages.

Table 32
Error messages for the MDPDPNMGR server

Error message	Meaning and action
Another instance of this program is already running.	Some programs can only have one instance running to preserve file integrity. Stop the first instance of a program before starting another.
Diskmanager is not running.	The disk manager must be running before the File Managers will start. Configure the disk manager. If it is configured but not started, start it using SVM.
Database is corrupt.	Stop and restart the File Manager. If you are running the BDF Converter, use the -force option.
Duplicate file received from switch.	No action required.
Current configuration does not process all file types.	If you do not wish to process the file type specified, you can ignore this message.
Started processing file.	MDP has started processing the raw file.
FINISHED processing file successfully.	MDP has successfully completed processing the raw file.
Finished processing file, errors occurred.	This message indicates the number of errors found in the raw file.
Processed records, unsupported records, and error records.	This message indicates the number of: records, unsupported records, and error records found in a file.

Chapter 24
MDP Disk Manager (MDPDISKMGR)

This section describes the information required to control the Management Data Provider (MDP) Disk Manager server (MDPDISKMGR) using the Preside Multiservice Data Manager Server Administration (SVM) tool. See the following sections for information about this server:

- About the MDPDISKMGR server (page 219)
- Managing the MDPDISKMGR server (page 220)
- Interdependencies (page 221)
- Error Messages (page 221)

About the MDPDISKMGR server

The MDPDISKMGR monitors the MDP host data disk space. If the available disk space nears the configured threshold, logs and alarms are generated. If the available disk space exceeds the configured threshold, data conversion is halted.

The MDPDISKMGR also launches a process each day at 0100hrs to remove obsolete data files. Data files are determined to be obsolete by exceeding the retention period for their respective directories.

For more information about MDPDISKMGR, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPDISKMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPDISKMGR. Any changes that you make to the startup command take effect when the MDPDISKMGR is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPDISKMGR server can be configured using the MDP Configuration tool (gmdpconfig) or by editting the file /opt/MagellanMDP/cfg/mdp/ MDPClean.cfg.

Suggested name in Server Administration

The recommended name for the MDPDISKMGR server is **MDP Disk Manager**.

Registering the MDPDISKMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPDISKMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpdiskmgr
[-freespace <space>]
[-loglevel <loglevel>]
[-help | -h]
```

where:

`[-freespace <space>]`

This optional parameter specifies the amount of available disk space to be retained on an MDP host. The default is 100000K.

`[-loglevel <loglevel>]`

This optional parameter specifies the level of logs to be reported. The default level is 4.

`[-help | -h]`

This optional parameter lists the command options.

Interdependencies

The MDPDISKMGR server does not depend on any other Preside Multiservice Data Manager server.

Error Messages

MDPDISKMGR error messages are written to file /opt/MagellanMDP/data/mdp/admin/
MDPDiskMgr(Sun|Mon|Tue|Wed|Thu|Fri|Sat)_YYYYMMDD.log.

MDPDISKMGR error messages are not written to the OAM Log Collector.

Table 33 describes the MDPDISKMGR error messages.

Table 33
Error messages for the MDPDISKMGR server

Error message	Meaning and action
Diskmanager is not running	This message is generated by MDP File Managers. The disk manager must be running before the File Managers will start. Configure the disk manager using gmdpconfig. If it is configured but not started, start it using SVM.

Chapter 25
MDP File Mover Manager (MDPFMMGR)

This section describes the information required to control the Management Data Provider File Mover Manager (MDPFMMGR) server using the Preside Multiservice Data Manager Server Administration (SVM) tool.

See the following sections for information about this server:

- About the MDPRMMGR server (page 223)
- Managing the MDPRMMGR server (page 224)
- Interdependencies (page 225)
- Error Messages (page 225)

About the MDPRMMGR server

The MDPRMMGR monitors the MDP host for converted data files. When closed converted data files are discovered, these files are transferred (using FTP) to customer hosts for network billing and performance analysis.

Also transferred to customer hosts from the MDP host are data files from these MDP applications: Statistics Retrieval System (SRS), outage calculation, availability calculation, and file processing audits.

The File Mover Manager controls multiple File Mover processes. Each MDP file type (a combination of switch type and datatype) has a dedicated File Mover process. This dedicated File Mover can transfer multiple instances of a specific file type and can use a maximum of three concurrent FTP sessions to transfer files to a maximum of three customer hosts.

If the File Mover Manager fails, any File Mover processes running at that time will continue to run. If the File Mover Manager and File Mover processes fail at the same time (for example, complete workstation shutdown), the File Mover Manager ensures that the appropriate File Mover processes are restarted.

If the File Mover Manager is shutdown, any File Mover processes running at that time are also shutdown.

For more information about the MDP File Move Manager, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPRMMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPRMMGR. Any changes that you make to the startup command take effect when the MDPRMMGR is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPRMMGR server can be configured using the MDP Configuration tool (gmdpconfig) or by editing file /opt/MagellanMDP/cfg/mdp/MDPMover.cfg.

Suggested name in Server Administration

The recommended name for the MDPRMMGR server is **MDP File Mover Manager**.

Registering the MDPRMMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPRMMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpfmmgr
```

Interdependencies

The MDPRMMGR server does not depend on any other Preside Multiservice Data Manager server.

Error Messages

MDPRMMGR error messages are written to file /opt/MagellanMDP/data/mdp/admin/**MDPFMMgr_YYYYMMDD.log**.

MDPRMMGR error messages are not written to the OAM Log Collector.

Chapter 26
MDP File Prober Manager (MDPFPMGR)

This section describes the information required to control the Management Data Provider (MDP) File Prober Manager (MDPFPMGR) server using the Preside Multiservice Data Manager Server Administration (SVM) tool.

See the following sections for information about this server:

About the MDPFPMGR server

Spooled data is collected by the Management Data Provider (MDP) using the File Prober. The File Prober establishes a proxy file transfer protocol (FTP) connection from a Passport node to an MDP and transfers closed spool data files. An FTP session is initiated for each datatype.

The File Probers are automatically launched at scheduled intervals.

The File Probers are controlled by the File Prober Manager (MDPFPMGR). The File Prober Manager provides

- ease of configuration using the MDP Configuration tool
- improved error recovery
- confirmation of node and MDP host connectivity before attempting data collection

- security protection using encrypted passwords

The File Prober Manager obtains network information from the Preside Multiservice Data Manager (MDM) Host Group Directory Server (HGDS). For more information about the HGDS, see 241-6001-310 *Preside MDM Server Reference Guide*.

For more information about MDPFPMGR, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPFPMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPFPMGR server. Any changes that you make to the startup command take effect when the MDPFPMGR server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPFPMGR can be configured using the MDP Configuration tool (gmdpconfig) or by editing the file /opt/MagellanMDP/cfg/mdp/MDPFPSched.cfg.

Suggested name in Server Administration

The recommended name for the MDPFPMGR server is **MDP PP Prober Manager**.

Registering the MDPFPMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPFPMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpfpmgr
[-max_concurrent <m>]
[-loglevel <0-4>]
```

where:

```
[-max_concurrent <m>]
```

is the maximum number of concurrent probers allowed. The default number is 2. The maximum limit is 128.

```
[-loglevel <0-4>]
```

specifies the level of logs reported. The default level is 3.

Interdependencies

The MDPFPMGR requires that the MDP Data Model Manager server and HGDS be active.

If collected node data files are to be converted from FMIP to BDF, the MDPFPMGR server requires that the MDP PP File Manager be active.

If converted node data files are to be transferred to a customer host for downstream processing, the MDPFPMGR server requires the MDP File Mover Manager server to be active.

Error Messages

MDPFPMGR error messages are written to file /opt/MagellanMDP/data/mdp/admin/**mdprober/<datatype>/mdprober_<Passport_nodename>_YYYYMMDD.log**.

MDPFPMGR error messages are not written to the OAM Log Collector.

Table 34 describes the MDPFPMGR error messages.

Table 34
Error messages for the MDPFPMGR server

Error message	Meaning and action
Failed to connect to host.	Check connectivity to host including user ID/ password.
FTP failed.	Retry FTP connection.

Table 34
Error messages for the MDPFPMGR server

Error message	Meaning and action
Failed to connect to switch.	Check connectivity to switch including user ID/ password.
FMIP communication error.	There is a problem with the FMIP communication between a node and the MDP workstation. If the problem persists, contact Nortel Networks Customer Support.

Chapter 27

MDP Passport Data Model Manager (MDPDMM)

This section describes the information required to control the Management Data Provider (MDP) Data Model Manager (MDPDMM) server using the Preside Multiservice Data Manager Server Administration (SVM) tool.

See the following sections for information about this server:

About the MDPDMM server

The Management Data Provider (MDP) stores a data model for each unique data model available in the network. These data models are used to convert performance metrics from FMIP to bulk data format (BDF).

The data models on the MDP host are controlled by the Data Model Manager. The MDP file prober identifies data models on the Passport node and queries the Data Model Manager to determine if the data model already exists on the MDP host. If the data model does not exist on the MDP host, the MDPDMM retrieves the data model before the collected node performance metrics are converted.

The MDPDMM also checks the MDP disk space to confirm that the available disk space is adequate to collect the node data.

For more information about the MDPDMM and data models, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPDMM server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPDMM. Any changes that you make to the startup command take effect when the MDPDMM is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPDMM server does not require configuration.

Suggested name in Server Administration

The recommended name for the MDPDMM server is **MDP Data Model Manager**.

Registering the MDPDMM server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPDMM server has the following syntax:

```
/opt/MagellanMDP/bin/mdpdmm
[-port <port>]
[-loglevel <0-4>]
[-spoolfree <n>]
```

where:

```
[-port <port>]
```

is the port for mdpdmm to listen on. By default, a port is dynamically allocated. This option is only required to support pre-14.3 clients using a fixed port.

```
[-loglevel <0-4>]
```

specifies the level of logs reported. The default level is 3.

```
[-spoolfree <n>]
```

is the amount of free space that must exist in the /opt/MagellanMDP/data/mdp/spool directory for mdprober. Default setting is 50M.

Interdependencies

The MDPDMM server does not depend on any other Preside Multiservice Data Manager server.

Error Messages

MDPDMM error messages are written to file /opt/MagellanMDP/data/mdp/admin/**MDPDMM_YYYYMMDD.log**.

MDPDMM error messages are not written to the OAM Log Collector.

Chapter 28
MDP MPE 9500 Collector Manager (MPEMCMGR)

The Management Data Provider Collector Manager (MPEMCMGR) server manages the collection of spooled data from an MPE 9500. This server is configured using the Preside Multiservice Data Manager Server Administration (SVM) tool.

See the following sections for information about this server:

About the MDPMCMGR server

Spooled data is collected by the Management Data Provider (MDP) using the File Collector (MDPMPECOL) process. The File Collector processes establish one of two supported file transfer sessions:

- a file transfer protocol (FTP) connection from a MPE 9500 switch to an MDP and transfers closed spool data files.

- a secure file transfer protocol (SFTP)connection from a MPE 9500 switch to an MDP and transfers closed spool data files.

An FTP or SFTP session is initiated for each MPE 9500 datatype. The default file transfer mode is SFTP.

The MPE 9500 File Collector instances are automatically launched at scheduled intervals as specified in /opt/MagellanMDP/cfg/mdp/MDPMCSched.cfg.

The MPE 9500 File Collectors are controlled by the File Collector Manager (MDPMCMGR). The MPE 9500 Collector Manager provides

- ease of configuration using the MDP Configuration tool
- error recovery
- security protection using encrypted passwords

The File Collector Manager obtains MPE 9500 network information from the Preside Multiservice Data Manager (MDM) Host Group Directory Server (HGDS). For more information about the HGDS, see 241-6001-310 *Preside MDM Server Reference Guide*.

For more information about MDPMCMGR, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPMCMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPMCMGR server. Any changes that you make to the startup command take effect when the MDPMPEMGR server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPMPEMGR can be configured using the MDP Configuration tool (gmdpconfig) or by editing the file /opt/MagellanMDP/cfg/mdp/MDPMCSched.cfg.

Suggested name in Service Administration

The recommended name for the MDP MPE 9500 File Manager server is **MDP MPE Collector Manager**.

Registering the MDPMCMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPMCMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpmcmgr [ -max_concurrent <n> ]
[ -loglevel <verbosity> ]
```

where:
<n> is the maximum number of concurrent mdpmpecol processes allowed. The default value is 8.

<verbosity> specifies the level of verbosity for logs written to the mdpmcmgr logfile. Enter a number between 0 and 4, that represents one of the following settings: CRITICAL = 0, ERROR = 1, WARNING = 2, INFO = 3, and VERBOSE = 4.

Interdependencies

The MDPMCMGR server requires that HGDS be active to collect MPE 9500 data.

If collected MPE 9500 data files are to be converted from XML to BDF, the MDPMCMGR server requires the MDP MPE File Manager be active.

If converted MPE 9500 data files are to be transferred to a customer host for down-stream processing, the MDPMCMGR server requires the MDP File Mover Manager server to be active.

Error Messages

MDPMCMGR error messages are written to file /opt/MagellanMDP/data/mdp/admin/ MDPMCMgr_YYYYMMDD.log.

MDPMCMGR error messages are not written to the OAM Log Collector (OAMC) and Security Audit Log Collector (SALC). Security audit logs and security-related events are sent to SALC by way of the OAMC, and Preside Multiservice Data Manager alarms are sent to OAMC.

MPEMCMGR-managed processes

Use the MPE 9500 Collector Manager (MPECMMGR) to manage MPE 9500 File Collector (MDPMPECOL) instances. Any changes that you make manually to the MDPMPECOL take effect when the MPEMCNGR server is restarted.

MDPMPECOL Configuration

The MDPMPECOL is configured using the MDP Configuration Editor tool.

MDPMPECOL start-up command

The command to start the MDPMPECOL process, although generally not invoked manually, is:

```
mdpmpecol -type <fileType> -ne <nodeName>
-user <userid> -pass <password>
[ -xfermode <mode> ]
[ -loglevel <verbosity> ]
[ -spoolfree <diskSpace> ]
[ -files <numFiles> ]
[ -timeout <timeout> ]
[ -noerase ]
```

where:

<fileType> specifies the file type to collect.
File types are defined on the MPE switch. For the accounting, statistics and trace streams, the file type may be a specific file type, such as, <stream usertype>, or may be only the stream. When the file type is specified by only the stream all file types in the stream are collected.

<nodeName> specifies the HGDS configured name of the MPE switch to collect from.

<userid> and <password> specifies a MPE user who has permission to write collected data file to the directory "/opt/MagellanMDP/data/mdp/mpe/spool" and to read the spooled files of the type specified. The password must be encrypted.

<mode> specifies the file transfer mechanism to use:

- -xfermode ftp: for ftp mode
- -xfermode sftp: for sftp mode

If the mode is not specified, the command uses the setting defined in /opt/MagellanNMS/cfg/FTS.cfg, or if that file does not exist the setting is taken from /opt/MagellanNMS/lib/cfg/FTS.cfg.

<verbosity> specifies the level of verbosity for logs written to the mdpmpecol logfile. Enter a number between 0 and 5, that represents one of the following settings: CRITICAL = 0, ERROR = 1, WARNING = 2, INFO = 3, VERBOSE = 4 and DEBUG = 5.

<diskSpace> specifies the amount of free disk space (in Megabytes) to maintain in the spool directory. The default is 50M. Values of less than 20M are not acceptable.

<numFiles> specifies a maximum number of files to transfer. Enter a number or “all” to transfer all files. The default value is 60.

<timeout> specifies the file transfer timeout period (in seconds). The default value is 300 seconds. The timeout value range is 60 to 1800 seconds.

-noerase ensures that files are not removed from the MPE after a transfer to the MDP. The default behaviour is to remove files from the network element after a successful transfer to the MPE 9500.

MDPMPECOL Log Messages

MDPMPECOL log messages are written to file
/opt/MagellanMDP/data/mdp/admin/
mdpmpecol_<nodename>_<fileType>_YYYYMMDD.log.

MDPMPECOL log files report the result of each collection job.

MDPMPECOL error messages are not written to the OAM Log Collector (OAMC) and Security Audit Log Collector (SALC). Security audit logs and security-related events are sent to SALC by way of the OAMC, and Preside Multiservice Data Manager alarms are sent to OAMC.

Chapter 29
MDP MPE 9500 File Manager (MDPMPEMGR)

The Management Data Provider MPE 9500 File Manager (MDPMPEMGR) server manages the conversion of raw data files to BDF format (when applicable) and the transfer of spooled files to down stream devices. This server is configured using the Preside Multiservice Data ManagerServer Administration (SVM) tool.

See the following sections for information about this server:

About the MDPMPEMGR server

The MDPMPEMGR server monitors the /opt/MagellanMDP/data/mdp/mpe/spool directory for collected MPE 9500 data files and controls file processing for BDF conversion and GZIP decompression. Specifically it controls the following:

- the conversion of MPE 9500 raw level 2 FrameRelay statistics and accounting metric data to bulk data format (BDF)
- the decompression of GZIP files
- the storage of converted data
- the creation of raw data backups

For more information about the MDPMPEMGR File Manager, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPMPEMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPMPEMGR server. Any changes that you make to the startup command take effect when the MDPMPEMGR server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPMPEMGR can be configured using the MDP Configuration tool (gmdpconfig) or by editing the file /opt/MagellanMDP/cfg/mdp/MDPMPEMgr.cfg.

Suggested name in Service Administration

The recommended name for the MDP MPE 9500 File Manager server is **MDP MPE File Manager**.

Registering the MDPMPEMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPPPMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpmpemgr
```

Interdependencies

The MDPMPEMGR server requires that the MDP MPE Collector Manager server be active to collect MPE 9500 data for conversion.

The MDPMPEMGR server requires that the MDP Disk Manager server be active to control MDP disk space.

If converted MPE 9500 data files are to be transferred to a customer host for down-stream processing, the MDPMPEMGR server requires the MDP File Mover Manager server to be active.

Error Messages

MDPMPEMGR error messages are written to file /opt/MagellanMDP/data/mdp/admin/MDPMPEMgr_YYYYMMDD.log.

MDPMPEMGR error messages are not written to the OAM Log Collector (OAMC) and Security Audit Log Collector (SALC). Security audit logs and security-related events are sent to SALC by way of the OAMC, and Preside Multiservice Data Manager alarms are sent to OAMC.

Table 35 describes the MDPMPEMGR error messages.

Table 35
Error messages for the MDPMPEMGR server

Error message	Meaning and action
Another instance of this program is already running.	Some programs can only have one instance running to preserve file integrity. Stop the first instance of a program before starting another.
Diskmanager is not running.	The disk manager must be running before the File Managers will start. Configure the disk manager. If it is configured but not started, start it using SVM.
Started processing file.	MDP has started processing the raw file.
FINISHED processing file successfully.	MDP has successfully completed processing the raw file.
Finished processing file, errors occurred.	This message indicates the number of errors found in the raw file.

Chapter 30
MDP Passport File Manager (MDPPPMGR)

This section describes the information required to control the Management Data Provider Passport File Manager (MDPPPMGR) server using the Preside Multiservice Data ManagerServer Administration (SVM) tool.

See the following sections for information about this server:

- About the MDPPPMGR server (page 245)
- Managing the MDPPPMGR server (page 246)
- Interdependencies (page 246)
- Error Messages (page 247)

About the MDPPPMGR server

The MDPPPMGR controls

- the conversion of Passport raw performance metric data to published format (PF) or bulk data format (BDF)
- the storage of converted data
- the creation of raw data backups

For more information about the MDP Passport File Manager, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPPPMGR server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPPPMGR server. Any changes that you make to the startup command take effect when the MDPPPMGR server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

Configuration

The MDPPPMGR can be configured using the MDP Configuration tool (gmdpconfig) or by editing the file /opt/MagellanMDP/cfg/mdp/MDPPPMgr.cfg.

Suggested name in Service Administration

The recommended name for the MDP Passport File Manager server is **MDP PP File Manager**.

Registering the MDPPPMGR server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPPPMGR server has the following syntax:

```
/opt/MagellanMDP/bin/mdpppmgr
```

Interdependencies

The MDPPPMGR server requires that the MDP PP Prober Manager server be active to collect Passport data for conversion.

The MDPPPMGR server requires that the MDP Disk Manager server be active to control MDP disk space.

If converted data files are to be transferred to a customer host for downstream processing, the MDPPPMGR server requires the MDP File Mover Manager server to be active.

Error Messages

MDPPPMGR error messages are written to file
/opt/MagellanMDP/data/mdp/admin/
MDPPPMgr(Sun|Mon|Tue|Wed|Thu|Fri|Sat)_YYYYMMDD.log.

MDPPPMGR error messages are not written to the OAM Log Collector.

Table 36 describes the MDPPPMGR error messages.

Table 36
Error messages for the MDPPPMGR server

Error message	Meaning and action
Another instance of this program is already running.	Some programs can only have one instance running to preserve file integrity. Stop the first instance of a program before starting another.
Diskmanager is not running.	The disk manager must be running before the File Managers will start. Configure the disk manager. If it is configured but not started, start it using SVM.
Database is corrupt.	Stop and restart the File Manager. If you are running the BDF Converter, use the -force option.
Duplicate file received from switch.	No action required.
Current configuration does not process all file types.	If you do not wish to process the file type specified, you can ignore this message.
Started processing file.	MDP has started processing the raw file.
FINISHED processing file successfully.	MDP has successfully completed processing the raw file.
Finished processing file, errors occurred.	This message indicates the number of errors found in the raw file.
Processed records, unsupported records, and error records.	This message indicates the number of: records, unsupported records, and error records found in a file.

Chapter 31
MDP Statistics Retrieval System (MDPSRS)

This section describes the information required to control the Management Data Provider Statistics Retrieval System (MDPSRS) server using the Preside Multiservice Data Manager Server Administration (SVM) tool.

About the MDPSRS server

The MDPSRS polls nodes for non-spooled real-time statistical information. Statistic records are converted to bulk data format (BDF) and transferred to a performance or billing host for down-stream processing.

Valid statistics are limited to the operational attributes of a Passport component. Two types of statistics can be reported, raw attributes and delta values. Raw attributes are retrieved from the node and written to the BDF file. Delta values indicate the difference between the value of a raw attribute from the previous poll and the value of the raw attribute from the current poll.

You select the node components and attributes to poll using the MDP Configuration tool.

SRS can be used with Passport Enterprise software release 5.0, or higher, or any software release for Passport Carrier/Wireless.

For more information about MDP SRS, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

Managing the MDPSRS server

Use the Server Administration (SVM) tool to configure the server startup command and to start, stop, and set options for the MDPSRS server. Any changes that you make to the startup command take effect when the MDP SRS server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

For more information, see the following:

- “Configuration” (page 250)
- “Suggested name in Server Administration” (page 250)
- “Start-up command” (page 250)

Configuration

The MDPSRS can be configured using the MDP Configuration tool (gmdpconfig) or by editing the file /opt/MagellanMDP/cfg/mdp/MDPSRS.cfg.

Suggested name in Server Administration

The recommended name for the MDP Statistics Retrieval System server is **MDP SRS**.

Registering the MDP SRS server with the Server Administration tool requires that the server name be used as the Descriptive name in the Server Administration dialog. The Server Administration tool stores this information in the file /opt/MagellanNMS/cfg/SVMList.cfg.

Start-up command

The command to start the MDPSRS server has the following syntax:

```
/opt/MagellanMDP/bin/mdpsrs
```

Interdependencies

The MDP SRS server requires that the HGDS and FDTM servers by active.

If SRS data files are to be transferred to a customer host for down-stream processing, the MDP SRS server requires the MDP File Mover Manager server to be active.

Error Messages

MDP SRS error messages are written to file /opt/MagellanMDP/data/mdp/admin/ **MDPSRS(Sun|Mon|Tue|Wed|Thu|Fri|Sat)_YYYYMMDD.log**.

MDP SRS error messages are not written to the OAM Log Collector.

Table 37 describes the MDPSRS error messages.

Table 37
Error messages for the MDPSRS server

Error message	Meaning and action
The specified component does not exist on the specified Passport.	Either the component name is incorrect or the component is not provisioned on the node.
System busy.	The node was unable to send polled information due to traffic congestion. Increase the polling interval or reduce the number of components polled.
FMIP communication error.	There is a problem with the FMIP communication between a node and the MDP workstation. If the problem persists, contact Nortel Networks Customer Support.
Another instance of this program is already running.	This program can only have one instance running to preserve file integrity. Stop the first instance of a program before starting another.

Chapter 32
Multi-nodal Name Server Agent (MNSDAGENT)

This section contains information on the multi-nodal name server agent (MNSDAGENT). See the following topics for more information:

About the MNSDAGENT

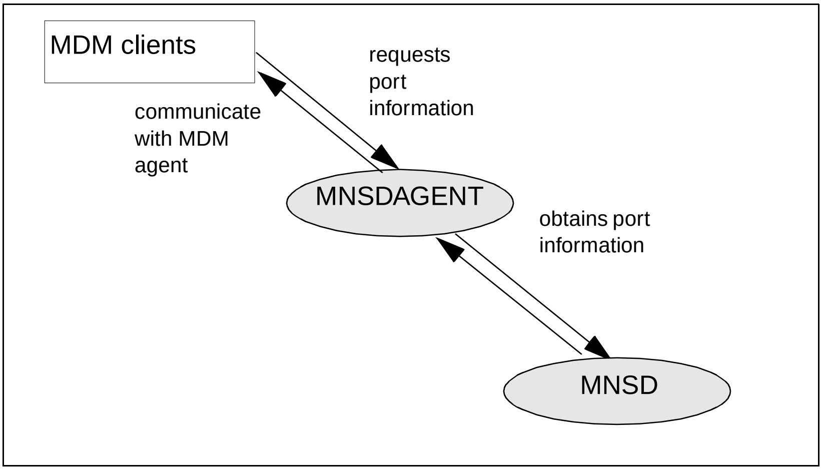
The MNSDAGENT functionality is similar to the multi-nodal name server (MNSD), with the exception that it contains an interface for Preside Multiservice Data Manager (MDM) Java applications to use. MNSDAGENT is the only agent with a fixed port. When a client needs to communicate with an agent, the client contacts the MNSDAGENT to look up the agent's current port. MNSDAGENT communicates with the MNSD server which stores port information. The JAVA/Web agent servers register their name and port number with the MNSD server. The JAVA/Web applications (clients) know how to communicate with the MNSDAGENT. When they want to establish communication with other agents (such as gmdragent, nmagent), they request the port number from the MNSDAGENT. The MNSDAGENT retrieves the information from the MNSD server and returns it to the client.

> ***Note:*** JAVA/Web applications cannot communicate directly with the MNSD server.

The MNSDAGENT provides Java clients with access the mnsd server for the purpose of looking up server ports. As a result, the servers that the Java clients communicate with do not have to run on a fixed port. Rather, ports can be dynamically allocated at startup.

The MNSDAGENT communicates with the MDM context server to set and retrieve MDM context information for Java clients.

Figure 25
Mnsdagent data flow diagram



Managing MNSDAGENT

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 255)
- “Startup command” (page 255)

Suggested name in Server Administration

The recommended name for the multi-nodal name server agent is MNSD Agent.

Configuring MNSDAGENT with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for MNSDAGENT is as follows:

```
/opt/MagellanNMS/bin/mnsdagent [-p <portno>]
```

where:

`-p <portno>` specifies the TCP port number for listening to incoming requests. The default value is 5934. This port number must not be used by any other process.

Interdependencies

MNSDAGENT depends on the MNSD server.

Exit codes

Exit codes for the MNSDAGENT are shown in the following table.

Table 38
Exit codes for the MNSDAGENT

Exit code	Description
51	Out of memory.
55	Bad argument on command line.
59	Could not initialize IPC system.
60	Could not register service. (Is the server running?)

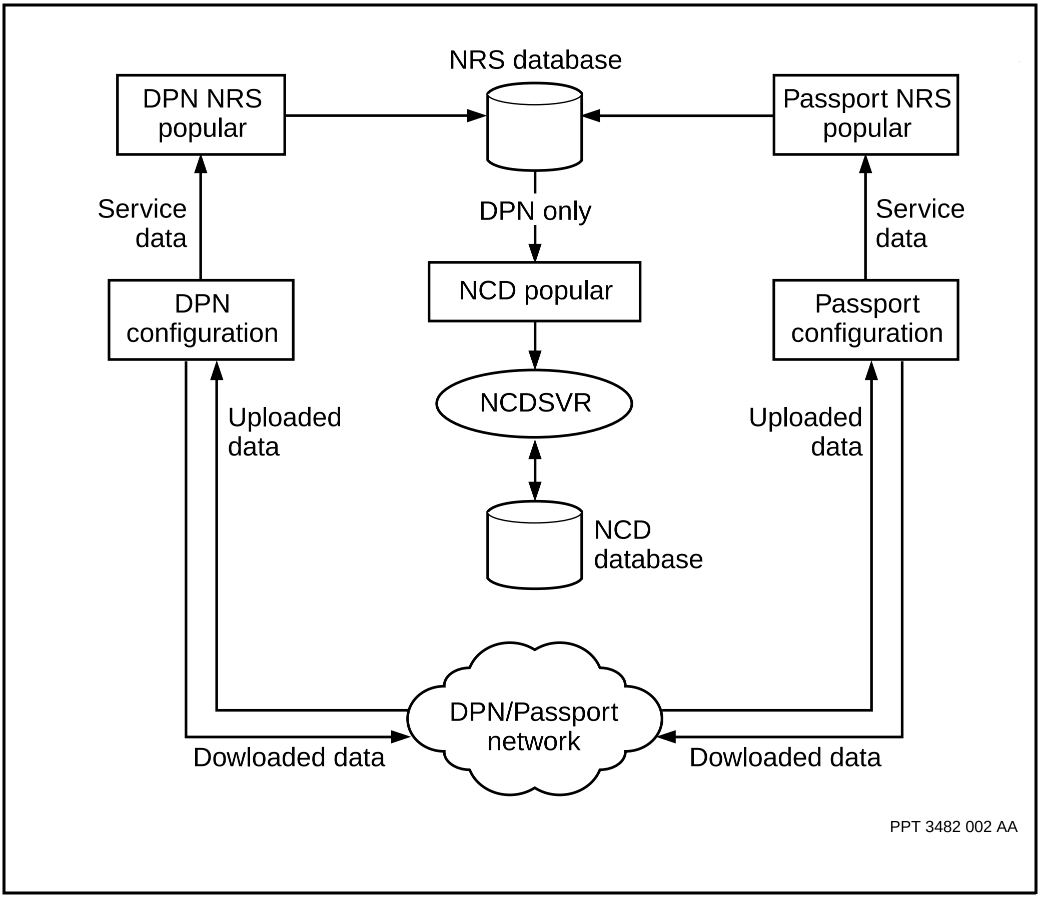
Chapter 33
Network Configuration Database Server (NCDSVR)

This section contains information on the Network Configuration Database server (NCDSVR). See the following topics for more information:

About the NCDSVR

The NCDSVR provides access to an internal database that contains service configuration data which must be unique across all DPN switches in the network. Unique items stored in this database include: data network addresses (DNAs), network administrator identifiers (NAMSIDs), gateway identifiers (GATEWAY_ID), and IP addresses.

Figure 26
NCD data flow diagram



Managing the NCDSVR

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the NCDSVR. Any changes you make to the startup command or options take effect when the NCDSVR is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

See also...

- “Suggested name in Server Administration” (page 259)
- “Startup command” (page 259)

Suggested name in Server Administration

The recommended name for the NCDSVR is:

```
NCD Server <database name>
```

where:

`<database name>` is the name of the NCD database you specified when you created an NCD database for this server by means of the ncd_create command

Configuring NCDSVR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the NCDSVR is as follows:

```
/opt/MagellanNMS/bin/ncdsvr -ncd <database name> \
[-log [<log_file>]] \
[-debug]

[-h]
```

where:

`-ncd <database name>` specifies the name of the NCD database associated with the NCDSVR that is to be started. This parameter is mandatory and no default is provided.

`[-log [<log_file>]]` specifies the log file for the NCDSVR. If -log is specified without a log file name, the default log file name used is ncdsvr.NCD_<target_NCD>.log.

`[-debug]` outputs debug messages.

`[-h]` outputs command usage information.

> ***Note 1:*** Entering the command without any parameters or options outputs command usage information (a simplified version of the -h option).
>
> ***Note 2:*** If the specified NCD database <database name> is already being used by an existing NCDSVR, an error message is output which indicates that the specified NCD database is in use, and the NCDSVR is not started.

Interdependencies

Before starting the NCDSVR, you must create the NCD database by means of the ncd_create command as described in the installation section in 241-6001-308 *Preside MDM Network Configuration Database for DPN Administrator Guide*.

Exit codes

When a server fails for any of the reasons listed, the exit code is displayed in the message area of the Server Manager Administration tool. The reason for the failure is displayed in the System Log Display and logged in the OAM log. Exit codes for the NCDSVR are shown in the following table.

Table 39
Exit codes for the NCDSVR

Exit code	Description
1	Error: command line syntax errors.
2	Error: specified target NCD has not been created yet.
3	Error: server “NCD_<database name>” has already been started.
4	Error: cannot open the log file.
5	Error: server’s version does not match the target NCD database version.
10	Warning: administrator attention is required (For example too many clients).
(Sheet 1 of 2)	

Table 39 (Continued)
Exit codes for the NCDSVR

Exit code	Description
11	Signal 1 (hangup) received.
50	Signal 2 (interrupt), 3 (qui0, or 15 (software termination received). The SVMDN process will not attempt to restart the server automatically.
62	Terminated due to licencing problem (see MDMLogs). Look at the System Log Display tool for logs about licencing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.
(Sheet 2 of 2)	

Error messages

Error messages for the NCDSVR are shown in the following table.

Table 40
Error messages for the NCDSVR

Error message	Meaning and action
NCD - Cannot allocate licensing context	The NCDSVR licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
NCD - License refused: <reason>	A run-time license cannot be allocated to the NCDSVR for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running NCD.
(Sheet 1 of 2)	

Table 40 (Continued)
Error messages for the NCDSVR

Error message	Meaning and action
NCD - License warning: <reason>	The license required to run NCD is about to expire. Contact Nortel Networks immediately.
NCD - License not confirmed: <reason>	There is a problem with the license required to run NCD, it has most likely expired. Contact Nortel Networks immediately.
(Sheet 2 of 2)	

Chapter 34
Network Data Access Mediator (NDAM)

This section contains information on the Network Data Access Mediator (NDAM). See the following topics for more information:

About the NDAM server

The NDAM server provides management data from Passport nodes to client applications.

The NDAM server performs the following functions to provide management data to a client application:

- receives requests for information from one or more client applications
- extracts state, alarm, and component information from the Preside Multiservice Data Manager (MDM) software by means of the Network Model Server (NMSERVER) and the General Management Data Router (GMDR). The GMDR the server is the prime source of this information.

- filters the data according to component types and device names. When the client application registers with the NDAM server, the registration request includes information that the NDAM server uses to locate files that contain criteria to filter management data for the client application. For a description of these files, see “Configuration” (page 269).
- forwards the filtered data to the client application

The NDAM server can therefore act as distributor of management data for applications such as HP Openview Desktop for MDM. It can also act as a filter between two hierarchical GMDR servers. Finally, it can act in place of a GMDR server to perform type and regional filtering for fault clients.

See the figure “NDAM server data flow diagram” (page 266) for a data flow diagram for the NDAM server. The items in the diagram and their functions are as follows:

- Devices are the network devices from which MDM software collects management data.
- Data collectors are specialized data collection processes. Each process collects alarms and state change notifications for a group of devices that belong to a family of Nortel Networks products. The data collection processes compute the raw states of the devices and their components, maintain a list of the active alarms for the devices, and forward the raw states to the GMDR server. Examples of the data collection processes for Passport devices are the FMDR, FDTM, and the FDTR servers.
- GMDR is an MDM server that consolidates the alarms and raw states that it receives from the data collectors and makes them available to other MDM servers, including the SURNUP server and the NDAM server. Although the figure “NDAM server data flow diagram” (page 266) shows only one GMDR server, there can be several GMDR servers that are arranged into a hierarchy, in which each GMDR server provides the alarms and raw states to another GMDR server.
- SURNUP is an MDM server that calculates component states based on algorithms and the alarms and raw states that the GMDR server provides to SURNUP. The SURNUP server keeps the Network Model up to date with the calculated state information.

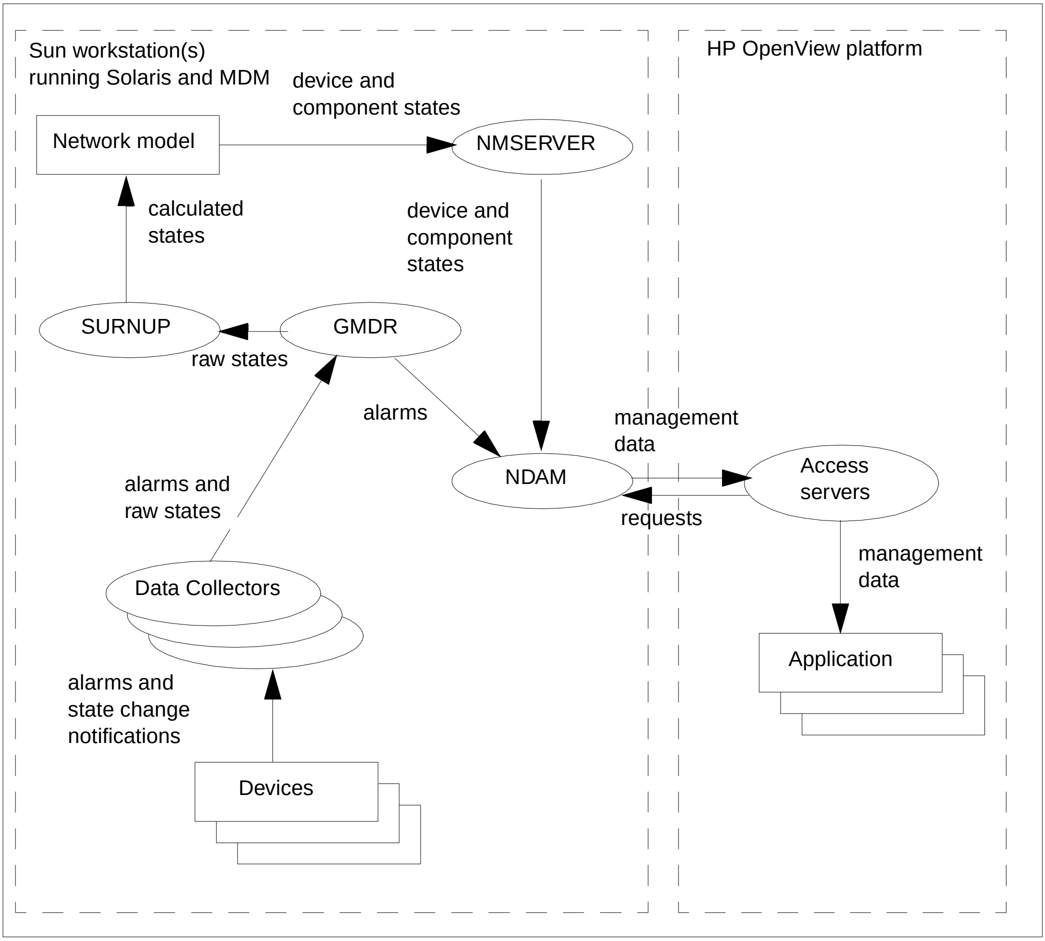
- The Network Model is a repository for the management data collected and calculated by MDM servers. The Network Model contains a current view of the devices and components in the network and their states.

  When using the Network Viewer or other fault tools, the usual practice is to organize the Network Model into sites and regions, as described in 241-6001-015 *Preside MDM Network Model Administrator Guide*. However, if you are not using the Network Viewer and other fault tools it is not necessary for you to perform this organization step. All that is necessary is to have the GMDR, SURNUP, and NMSERVER running with sufficient shared memory to accommodate the Network Model.

- NMSERVER is an MDM server that provides access to the Network Model. The NMSERVER provides NDAM with the current component states from the Network Model.
- NDAM is an MDM server that provides management data for Passport devices. MDM collects this management data and makes it available to client applications that run on the HP Openview platform.
- Access servers are processes running on HP OpenView that:
  - — send requests for management data to the NDAM server
  - — translate the information they receive into a form that the application can use

Examples of these servers are the Postmaster Daemon (PMD) server, the OpenView Alarm Translator (OVAT), and the OpenView Data Access Mediator (OVDAM) servers that supply management data to HP Openview applications such as the Event Browser.

Figure 27
NDAM server data flow diagram



Managing the NDAM server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for NDAM. Any changes you make to the startup command or options take effect when NDAM is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

For more information, see the following:

- "Suggested name in Server Administration" (page 267)
- "Startup command" (page 267)

Suggested name in Server Administration

The recommended name for the NDAM server is NDAM.

Startup command

The startup command for the NDAM server has the following syntax:

```
/opt/MagellanNMS/bin/ndam \
[-b size]
[-g [<GMDR service name>@] <GMDR host>]
[-m <NMSERVER host>]
[-n <NDAM name>]
[-s]
[-F]
```

where:

`[-b size]`

is the maximum number of messages that NDAM stores for each of its client applications. The default is 10 000 messages and the maximum is 50 000 messages.

`[-g [<GMDR service name>@]<GMDR host>]`

specifies the service name of the GMDR server that the NDAM server uses as the source of management data and the name of the host on which the GMDR server is running. By default, the GMDR service name is GMDR, and the default GMDR host is the host name of the workstation that is currently set by the administrator of the Service Selection tool

`[-m <NMSERVER host>]`

specifies the name of the host that runs the NMSERVER used by the NDAM server to access the Network Model. By default, the NMSERVER host is the host name of the workstation that is currently set by the administrator of the Service Selection tool. If the special value "~" is specified (the quotes are mandatory), NDAM does not attempt to maintain a connection to the NMSERVER. This is needed when NDAM is used as a inter-GMDR filter or as a replacement for GMDR since no Network Model information is needed in those cases.

`[-n <NDAM name>]`

is an alternate service name for the NDAM server. This option only needs to be specified when two or more NDAM servers are running on the same workstation. By default, the service name for the NDAM server is NDAM. To configure NDAM as a proxy for a GMDR server, specify GMDR as the NDAM service name. You will then also need to configure a wildcard authentication if NDAM is also running in secure mode. In this mode, NDAM acts just like GMDR towards its clients, with two exceptions. The GMDR Administration tool and the inbound API, which must be connected to the real underlying GMDR server. The Alarm and Status API repFilter, repInfo, repScope, repOClass, and repOid sieve attributes are not available because NDAM does not have an object model of its own these filters can be applied to.

`[-s]`

indicates that NDAM should run in secure mode. NDAM then only provides data to clients that register with a user ID and password that matches one of the configured ones. NDAM also performs automatic filtering on this client connection as configured for each valid authentication. See “Secure mode authentication configuration” (page 275) for more details on valid authentication configurations.

`[-F]`

indicates that NDAM should perform component type filtering using fully qualified type specifications. By default, NDAM only compares on the first and last component categories. All typeset files used in this mode must be fully qualified too (the typesets provided with Preside Multiservice Data Manager are only specified in using the default first-last method) as no mix specifications are allowed. GLOB style pattern matching is supported.

Interdependencies

To provide client applications with management data, the NDAM server requires access to the following servers:

- data collectors (FMDR and FDTM)
- SURNUP
- NMSERVER
- GMDR

Configuration

When an application sends the NDAM server a registration request, the request includes information that the NDAM server uses to locate files which contain criteria for filtering management data for the client application. The registration request messages can also contain additional filtering criteria from the client application.

The NDAM server looks first at a typeset file for the criteria to filter management data according to the component type, then at a deviceset file for the criteria to filter management information according to the device name. See the following sections for information about the two files:

- “Typeset configuration files” (page 269)
- “Deviceset configuration files” (page 273)

Typeset configuration files

A typeset configuration file specifies filtering criteria that the NDAM server uses to filter data according to component types for a client application.

Typeset files are named:

```
NDAM_<typeset name>.typ
```

where:

`typeset name`

is the name of a set of device types. For example, NDAM_ALL_EM.typ contains default filters that are used for all Passport devices.

Preside Multiservice Data Manager software provides a template typeset file in directory /opt/MagellanNMS/lib/cfg. Do not modify the file in the /opt/MagellanNMS/lib/cfg directory. If you wish to modify the file for your purposes, copy the lines you wish to modify out of the template file into a new file that has the same name in directory /opt/MagellanNMS/cfg. Then modify the new file. It is not necessary to include the entire contents of the template file, only the lines you wish to modify.

The syntax used in a typeset file is specified in one of two ways:

- as first and last category only (default)
- fully qualified (with the -F command line option)

The syntax for the default first-last category mode is as follows:

```
# <string>
```

introduces a comment. An example of a comment is a line that contains the words EM #Passport Module,. In this line, #Passport Module is a comment that the software ignores.

```
#include <filename>
```

treats the contents of the file specified in the file name as if the contents of the file are part of this typeset file. The include statement lets you specify typesets inside typeset files. For example, you can create a typeset file for a geographic region that includes typeset files for the cities within the region.

```
<device type>
```

passes module level data for devices of the specified type. For example, a line containing the entry EM passes data for all Passport modules.

```
*
```

passes module level data for devices of all types

```
<device type>-<last subcomponent type>
```

passes data for subcomponents of a specified device and subcomponent type. For example, EM-DS1 passes data for all DS1 subcomponents of all EM devices. See 241-6001-015 *Preside MDM Network Model Administrator Guide* for the list of allowed subcomponent types.

`<device type>-*`

passes data for all subcomponents of a specified device type. For example, EM-* passes data for all subcomponents of EMs.

`<link type>:`

passes data for links of the specified link type. for example PTK: passes data for all links of type PTK

`*:`

passes data for all types of links.

`!*`

rejects module level data for all types of devices.

`!<device type>`

rejects module level data for all devices of the specified type. For example, !EM rejects data for all EM modules.

`!<device type>-<last subcomponent type>`

rejects data for subcomponents of a specified device and subcomponent type. For example, !EM-DS1 rejects data for all DS1 subcomponents of EMs.

`!<device type>-*`

rejects data for all subcomponents of the specified device type. For example, !EM-* rejects data for all subcomponents of EMs.

`!*-*`

rejects data for all subcomponents of all device types.

`!<link type>:`

rejects data for links of the specified type. For example, !PTK: rejects data for all PTK links (trunks).

`!*:`

rejects data for types of links.

The syntax for the fully-qualified mode (-f option) is nearly identical for link, module and wild card specifications (*). In fully qualified mode, subcomponents are specified using all the intermediate categories (with GLOB style pattern matching supported). For example, the default mode specification of EM-FRAMER is equivalent in fully qualified mode to EM-*FRAMER. On the other hand, it is possible to distinguish in fully qualified mode between a Passport FrUni Framer (EM-FRUNI-FRAMER) and an Unacknowledged Trunk Framer (EM-TRK-UNACKED-FRAMER). Note that all typeset files provided with Preside Multiservice Data Manager are specified using the default mode and that is not possible to use both types of specifications at once.

The software examines the contents of a typeset file starting at the top of the file and filters when it finds a match. Therefore when creating a file, you need to put the most specific entries above any wild card (*) entries. If the software does not find a match for an entry in the file, it rejects data from the component by default. For example, say there is a typeset file that contains the following lines:

```
EM
EM-LP
PTK:
MPA
!MPA-PE
MPA-*
```

When the NDAM filters data, it passes data to the client for Passport devices, LPs and trunks, Passport 4400 (MPA) devices, and all subcomponents except PEs. Say we invert the order of the last two lines in the file like this:

```
MPA-*
!MPA-PE
```

With the lines inverted, the software passes information from all PEs because the software reads line labelled MPA-* first, and passes information from all MPA subcomponents including MPA- PEs. The last line is actually of no use.

Deviceset configuration files

A deviceset configuration file specifies the filtering criteria that the NDAM server uses to filter data according to device names for a client application.

Deviceset files are named:

```
NDAM_<deviceset_name>.dev
```

where:

```
deviceset_name
```

is the name of a set of devices. For example, EUROPE_Passport.

The syntax used in a deviceset file is as follows:

```
# <string>
```

is a comment. For example in a line that contains the words EM NODEX5 # a Passport module, # a Passport module is a comment that the software ignores.

```
#include <file name>
```

treats the contents of the file specified in the file name as if the contents of the file are part of this deviceset file. The include statement lets you specify devicesets inside deviceset files. For example, you can create a deviceset file for a geographic region that includes deviceset files for the cities within the region.

```
<device type> <device name>
```

includes the specified device type and device name in the set of devices for which the NDAM server passes management data to the client application. The device specified by the device type and the device name must be present in the Network Model to obtain information to filter. For example, assuming that device EM NODER99 is present in the Network Model, the line EM NODER99 includes NODER99 in the set of devices for which NDAM passes management data to the client application.

```
<device type> <name prefix>*
```

includes all devices with specified device type whose name begins with the specified prefix in the set of devices. For example, EM NODER* includes all EMs whose node name begins with NODER.

```
<device type> *
```

passes management data for all the devices of the specified type that are present in the Network Model to the client application

```
*
```

includes all of the devices in the Network Model in the device set

```
!<device type> <device name>
```

excludes the specified device type and device name from the device set defined by the previous lines in the file

```
!<device type> <name prefix>*
```

excludes all the devices of the specified type whose name starts with the specified name prefix from the device set defined by the previous lines in the file

```
!<device type> *
```

excludes all of the devices of the specified device type from the device set defined by the previous lines in the file

The software examines the contents of the deviceset file starting at the top of the file and filters when it finds a match. Therefore when creating a file, you need to put the most specific entries above any wild card (*) entries. If the software does not find a match for an entry in the file, NDAM rejects data from the component by default. For example, say there is a typeset file that contains the following lines:

```
EM NODEX5
!EM NODEX*
EM *
```

When the NDAM server reads the file it creates a device set that includes all Passport nodes in the network model, except the devices whose names start with NODEX. However, EM NODEX5 is included because the software encounters the line EMNODEX 5 before it encounters the exclude statement !EM NODEX *.

Say we invert the two first lines like this:

```
!EM NODEX*
EM NODEX5
EM *
```

With the first two lines inverted, the software excludes data from NODEX5 because it finds a match for the !EM NODEX* entry first.

Links are considered part of the device set if at least one of their endpoints is on a device accepted by the filter type and device sets.

Secure mode authentication configuration

The authentication information for NDAM running in secure (-s) mode is stored in file /opt/MagellanNMS/cfg/private/NDAM.passwd with a format similar to the UNIX password file (man -s4 passwd). The user name and password are encoded in the usual fields. The comment (also known as GECOS) field is used. The password information in the file must be encrypted so it is not possible to add entries to it by hand. The ndamuser utility is provided for this purpose and it has the following command line syntax:

```
/opt/MagellanNMS/bin/ndamuser <user name> \
[<password> [<typeset(.typ) and deviceset(.dev)
names...>]]
```

where:

`<user name>`

is x for a global wildcard, %<SERVICE_NAME> for a service specific wildcard, or another string for the corresponding authentication. A SERVICE NAME must be in uppercase letters. Specific wildcards accept any user authentication registered on the corresponding service if a matching non-wildcard authentication does not exist. Service specific wildcards may or may not have a password specified. The global wildcard accepts any authentication from any service if there is no matching specific nor service specific wildcard authentication. The global wildcard usually has no password associated with it (x) in order to allow Preside Multiservice Data Manager (MDM) fault tools to access the GMDR data.

> ***Note:*** If NDAM is used in the context of HP Openview Desktop for MDM, a proper wildcard authentication must be configured in NDAM.

`<password>`

is a valid password or x to indicate that no password is necessary.

`<typeset>.typ or <deviceset>.dev`

are either Component Type or Device filterset names. The name must not specify the standard prefix NDAM_ but must have a suffix of .typ” for typesets or .dev for devicesets. The corresponding files must exist as /opt/MagellanNMS/cfg/NDAM_<typeset>.typ or /opt/MagellanNMS/lib/cfg/NDAM_<typeset>.typ for typeset files and /opt/MagellanNMS/cfg/NDAM_<deviceset>.dev for devicesets. If the files do not exist at the time of running the ndamuser utility, a warning will be issued to remind you of this. Multiple typesets and devicesets can be specified. They will be used as forced filters for the connections registered with the corresponding authentication.

The ndamuser utility can be used to add new authentications and to change existing ones. To delete authentications, you must use a UNIX text editor to edit the NDAM password file and remove the corresponding line. The nmdamuser utility cannot be used for deleting authentications. You cannot manually enter new authentication in the NDAM password file because the password must be encrypted. You must use the ndamuser utility for this purpose.

Exit codes

Exit codes for the NDAM server as shown in the following table.

Table 41
Exit codes for the NDAM server

Exit code	Description
51	Memory resource error. For example, not enough memory. No restart.
55	Invalid command line argument
59	ipc_init unsuccessful. No restart.
60	IPC service cannot be registered. No restart.
62	Terminated due to licensing problem (see MDM Logs). Look at the System Log Display tool for logs about licensing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.

Error messages

Error messages for the NDAM server are shown in the following table.

Table 42
Error messages for the NDAM server

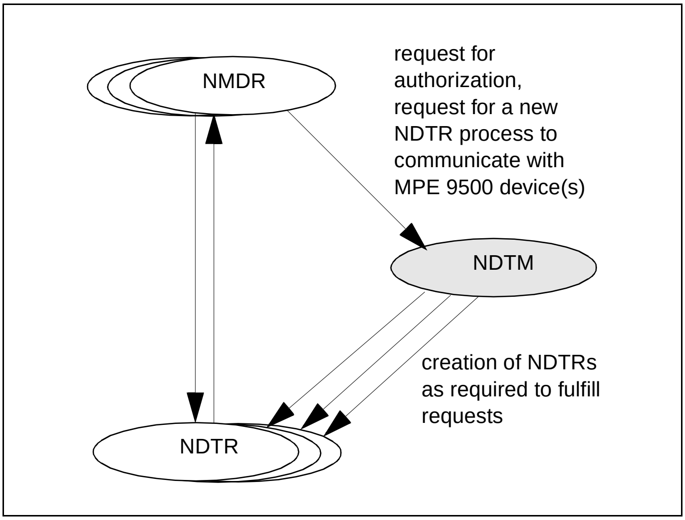
Error message	Meaning and action
NDAM - Unable to establish a licensing context	The NDAM server licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
NDAM - License request failed:	A run-time license cannot be allocated to the NDAM server for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running NDAM.
NDAM - License warning:	The license required to run FDTM is about to expire. Contact Nortel Networks immediately.
NDAM - Reloading current configuration files	The NDAM server has received a HUP signal and has reloaded its filter configuration files.
NDAM - Could not load filter description file <filter file name>	The NDAM server cannot read the file. The file may be non-existent, incorrect, or have invalid access rights.
NDAM - Invalid command line argument: <argument>	The specified command line argument passed to the NDAM server cannot be recognized. The server will not restart automatically.
NDAM - <client name> : killed client connection due to congestion.	The connection to the named client has been forced down due to excessive congestion. Verify whether the client is blocked
NDAM - <client name> : unexpectedly lost client connection.	The connection from the named client has been lost unexpectedly.
NDAM - Connected to GMDR Server <GMDR name> on <hostname>.	The NDAM server has successfully connected to the assigned GMDR server.
NDAM - Connected to NM Server on <hostname>	The NDAM server has connected successfully to the assigned NMSERVER.
NDAM - Dropping client <client name> due to server loss.	There has been an unexpected loss of connection to the named server (GMDR or NMSERVERl).

Chapter 35
MPE 9500 Communications Manager (NDTM)

The NDTM server performs the following functions:

- creates and manages Nortel Networks Multiservice Provider Edge 9500 (MPE 9500) data translation (NDTR) processes. The NDTR processes allow the workstation to communicate with a MPE 9500 device.

Figure 28
NDTM data flow diagram


NMDR
request for authorization, request for a new NDTR process to communicate with MPE 9500 device(s)
NDTM
creation of NDTRs as required to fulfill requests
NDTR

Navigation

- “Suggested name in Server Administration” (page 280)

- “Startup command” (page 280)
- “Interdependencies” (page 281)

Suggested name in Server Administration

To configure the NDTM server using the Server Administration tool, you must enter a name in the Descriptive name field. For the NDTM server, the suggested name is MPE Communications Manager.

Startup command

The command to start an NDTM server has the following syntax:

```
/opt/MagellanNMS/bin/ndtm \
[-connTimeOut <timeout_length>] [-h]\
[-numAuthNodes] <number_of_nodes>]\
[-numAuthRetries <number_of_retries>] \
[-connHeartBeatTime <interval_time>] \
[-msgTransTime <waiting_time>] \
[-ping <ping_time>]\
[-congestBuffer <buffer_size>] \
[-watch] \
[-reConnTime <reconnect_timer>] \
```

Variable definitions

Variable	Value
-connTimeOut <timeout_length>	the length of time that a MPE 9500 Shelf View or Data Viewer connection remains active. The range is 0-90 minutes and the default is 0. A value of 0 means that connections will not time out.
-h	displays help information about the start up command
-numAuthNodes <number_of_nodes>	the number of nodes that the NDTM uses to authenticate a userid and password. The nodes used are chosen at random. The range is 1-10 nodes and the default is 3.

Variable	Value
-numAuthRetries <number_of_retries>	the maximum number times that the NDTM will try to authenticate a user ID and password. The range is from 0-90 times and the default is 3.
-connHeartBeat Time <interval time>	the number of seconds between each heartbeat. The default is 30.
-msgTransTime <waiting_time>	the number of seconds that Preside Multiservice Data Manager waits for a response from a node. The default is 20.
-ping <ping_time>	the number of seconds used to determine if a node is reachable when attempting to connect to a group. The range is from 1-60 and the default is 2.
-congestBuffer <buffer_size>	the number of messages that will be stored in the outgoing queue to a client. When the queue overflows, the NDTR tears down the connection. The defaut setting is 2000.
-watch	
-reConnTime <reconnect_timer>	the number of seconds between each attempt to reconnect to a node. A value between 2 and 120 seconds represents the range of 2 seconds to 120 seconds. The default value is 120 seconds.

Interdependencies

The NDTM must be running on the same workstation as the NMDR.

Chapter 36
MPE 9500 Management Data Router server (NMDR)

The NMDR server routes alarm event reports from Nortel Networks Multiservice Provider Edge 9500 (MPE 9500) devices to a GMDR server.

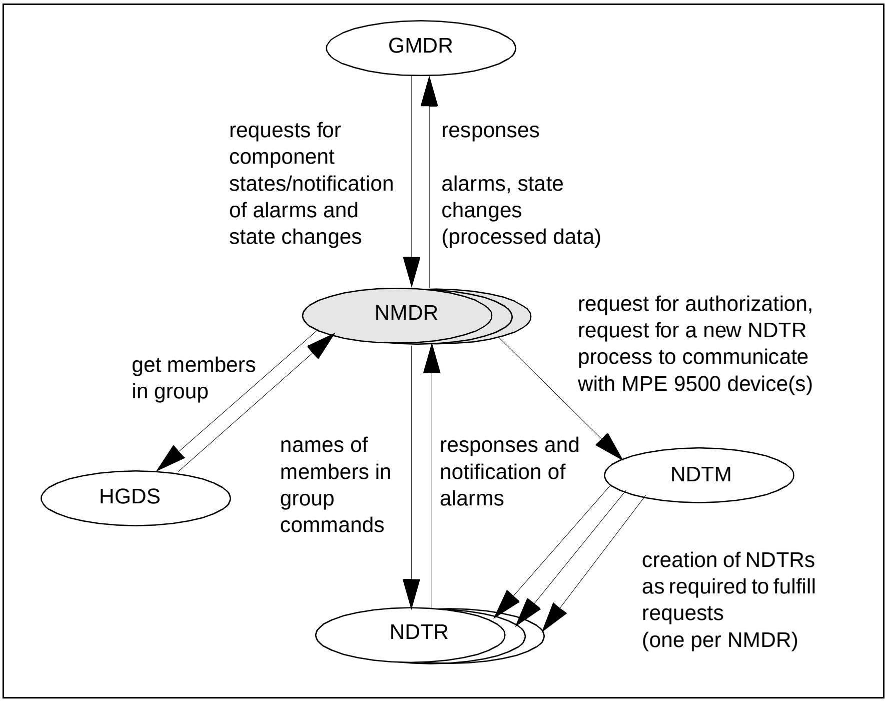
Navigation

NMDR functionality

The NMDR server performs the following functions:

- collects and stores surveillance information from MPE 9500 devices
- routes surveillance information such as alarm event reports from MPE 9500 devices to its GMDR clients
- detects new or obsolete components on MPE 9500 devices and notifies its GMDR clients
- supports the automatic deletion of dynamic components on MPE 9500 devices and notifies its GMDR clients
- supports get requests, local alarm clear requests, database reset requests, resynch requests, component delete requests, and query property requests from its GMDR clients

Figure 29
NMDR data flow diagram


GMDR
requests for component states/notification of alarms and state changes
responses
alarms, state changes (processed data)
NMDR
request for authorization, request for a new NDTR process to communicate with MPE 9500 device(s)
get members in group
names of members in group commands
responses and notification of alarms
NDTM
HGDS
creation of NDTRs as required to fulfill requests (one per NMDR)
NDTR

MPE 9500 access

NMDR servers automatically log on to a group of MPE 9500 devices during system initialization using the user ID and password specified in the server startup command. The NMDR server manages each MPE 9500 device in the group, using the group member information supplied by the HGDS server.

The NMDR server does not contact the MPE 9500 device directly; it asks the NDTM server for an NDTR process for MPE 9500 access. The NDTR process also issues a lost connectivity proxy alarm when the connection to a MPE 9500 device is lost. NMDR and NDTR automatically attempt to re-establish any failed connections.

Discovery

NMDR initializes or resynchronizes its surveillance database through an explicit discovery of the MPE 9500 devices in its group. The discovery allows NMDR to

- extract the active alarm list and alarm notification flow
- learn about component existence
- extract current OSI state and status information on components. Although OSI state and status attributes are extracted during a discovery, these values are not used for computing the raw state of the components; therefore, no proxy alarms are generated by NMDR. OSI state and status attributes are stored for later use through the NMDR API.

Between discoveries NMDR relies on the alarm notification flow from the MPE 9500 to accurately maintain raw state and alarm information for the MPE 9500 devices in its group. The raw state of a component is based on the severity of the associated alarm(s). Events are forwarded on to the clients of NMDR.

Events that trigger a discovery

A discovery is triggered automatically by the following events:

- initial connection to a MPE 9500 by the NMDR
- reconnection to a MPE 9500 by the NMDR
- the NMDR receiving a MPE 9500 Commit CLR alarm (sent after committing a provisioning session)

A discovery is also triggered when you perform the following administrative actions from the GMDR administration tool:

- synchronize the NMDR server on a specific module
- reset subserver database on the NMDR server
- delete a component on a MPE 9500 module (deletes the module and all of its subcomponents)

You can configure the NMDR so that a discovery on a particular MPE 9500 or the whole group is performed at regular intervals. By default, there are no recurrent discoveries. See “Configuring automatic discoveries” (page 289) for more information.

Alarm handling

NMDR may be configured to handle specific alarms in a special way, either by ignoring an alarm, or by changing the severity of a SET or MSG alarm. In this way, an alarm that is considered to be more important than is indicated by its severity may be mapped to a more critical state or an alarm that is considered to be less important than is indicated by its severity may be mapped to a less critical state. For more information see “Additional configuration for alarm exception handling” (page 291).

> ***Note:*** Unlike FMDR, NMDR does not generate proxy alarms to represent out of service (OOS) or troubled components without sufficient device alarms. All components that exist and have no device alarm against them are considered In-Service (INSV).

Dynamic components

A MPE 9500 device is identified as a dynamic components when an alarm is raised against it before it is discovered through the discovery process. Once all alarms against a dynamic component are cleared, the component is automatically deleted from NMDR database.

Interdependencies

The NMDR server relies on the HGDS server for host and group information.

The NMDR server relies on the NDTM server to provide the NDTR process which provides connection management and data translation.

NMDR administration

The NMDR supports the following administrative actions through the GMDR administration tool:

- reset the database
- delete a component
- trigger a resynchronization

For more information about using the GMDR Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

Suggested name in Server Administration tool

To manage the NMDR server using the Server Administration tool, you must enter a name in the Descriptive name field. For the NMDR server, the suggested name is MPE Management Data Router.

For more information about using the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

NMDR configuration

You can set various options to configure how the NMDR behaves at start up as well as configuring the NMDR to perform automatic discoveries and handle alarm exceptions.

Startup command

The command to start the NMDR server has the following syntax:

```
/opt/Magellan/bin/nmdr \
-g <GroupName> -u <userid> \
-p <password> \
[-b <size of the message buffer>] \
[-h] <display help menu>
[-w <maximum number of concurrent discoveries>] \
[-T <post-commit discovery delay in seconds>] \
[-r <number of seconds between retrying
[-R <maximum number of consecutive retrieve attempts]
[-e <alarm exceptions file name]
```

Variable definitions

Variable	Value
-g <GroupName>	the name of the NMDR group that the NMDR server is managing
-u <userid>	the user ID used to log into the devices in the MPE 9500 group. At minimum, the userid must have a login class of operator.

Variable	Value
-p <password>	the password used to log into the devices in the MPE 9500 group. If you are using password encryption, this variable must contain the full path name of the file that contains the encrypted password.
-b <size of message buffer>	the number of replies that can be contained in the queue before the client is cut off. The default value is 5000. It is recommended that you do change the default value.
-w <maximum number of concurrent discoveries>	The default value is 10.
-T <post-commit discovery delay in seconds>	the time, in seconds, between discoveries after a message commit alarm is received. • A value of 0 disables post-commit discoveries. • A value of 1 enables immediate post-commit discoveries. • A value of x>=300 enables a post-commit throttle of x seconds. A new discovery is postponed if one has already occurred within the x seconds interval. The default value is 900.
-h	displays the help menu
-r	displays number of seconds to wait before retrying to get the Active Alarm List from the MPE 9500. The default value is 10 seconds.

Variable	Value
-R	displays the maximum number of consecutive attempts to retrieve the Active Alarm List from the MPE 9500. The default value is 3 times.
-e	displays the full pathname of a file specifying the alarm exceptions. NMDR selects the appropriate alarm exceptions files for the MPE 9500 families. If you use this option to load a particular alarm exceptions file, you override the NMDR-selected files. As a consequence, NMDR uses the specific alarm exceptions file that you have loaded for all the nodes in the group.

Configuring automatic discoveries

You can configure automatic discoveries by creating a configuration file called /opt/MagellanNMS/cfg/NMDRDiscovery_<group>.cfg. The existence of the file triggers recurrent discoveries when the NMDR starts up. You need to create a separate file for each NMDR server that you want to run automatic discoveries.

As part of its name, this file uses the name of the MPE 9500 group that the NMDR server manages. For example, for server NMDR_WESTREGION, the filename is NMDRDiscovery_WESTREGION.cfg. Each entry in the file must be separated by a blank line. The format of an entry in this file is as follows:

Procedure steps

1 Using a text editor, create a file called /opt/MagellanNMS/cfg/NMDRDiscovery_<group>.cfg.

2 Create an entry for each automatic discover that you want to trigger. Each entry in the file must be separated by a blank line. The format of an entry in this file is as follows:

```
START_TIME: [<day>] [<hh:min>]
```

INTERVAL: [DAILY|WEEKLY|<hours>]

MODULE: [ALL |<module name>]

Variables

Table 43

Variable	Definition
<group>	
<day>	the day of the week at which the discoveries are to begin.Values are Sun, Mon, Tue, Wed, Thu, Fri, and Sat. If this parameter is omitted, the current day is used as the day to begin triggering discoveries.
<hh:min>	the time of day at which discoveries are to begin. Values are from 00:00 (midnight) to 23:59 (one minute before midnight). If this parameter is omitted, discoveries begin at 00:00.
DAILY|WEEKLY|<hours>	the frequency at which the discoveries are to be triggered once the START_TIME is reached. Values are DAILY, WEEKLY, or the number of hours, for example, 2. If this parameter is omitted, discoveries are triggered DAILY.
ALL|<module name>	the module on which the discovery is to be performed. Values are ALL or the name of a provisioned MPE 9500 module. If this parameter is omitted the discovery is performed on all modules managed by the NMDR server. More than one module can be specified, but each module name must be entered on a different line, and must be preceded by MODULE:

Example

To trigger a discovery every two hours and obtain the states of modules MSP NODEA03 and MSP NODEA04 starting at 00:00 hours of the current day, the configuration file would need to contain the following entries:

```
START_TIME: 00:00
INTERVAL: 2
MODULE: SRS NODEA03
MODULE: SRS NODEA04
```

By default, post-activation and recurrent discoveries will not be triggered less than 15 minutes after the last walk on the target MPE 9500 device in order to avoid over-stressing it.

Additional configuration for alarm exception handling

The handling of alarm events in NMDR is controlled by the NMDR alarm exceptions configuration file. The file /opt/MagellanNMS/cfg/NMDRAlarmExcep.cfg specifies the actions that are to be taken when certain alarms are received.

If the file does not exist in /opt/MagellanNMS/cfg/, then copy /opt/MagellanNMS/lib/cfg/NMDRAlarmExcep.cfg to /opt/MagellanNMS/cfg/NMDRAlarmExcep.cfg and modify it.

Procedure steps

1 Ensure that the file /opt/MagellanNMS/cfg/NMDRAlarmExcep.cfg exists in the correct directory. If it does not, then copy /opt/MagellanNMS/lib/cfg/NMDRAlarmExcep.cfg to /opt/MagellanNMS/cfg/NMDRAlarmExcep.cfg.

2 Modify /opt/MagellanNMS/cfg/NMDRAlarmExcep.cfg to create entries that specify actions to be taken when certain alarms are received. The entries must take the following form:

```
<event type> <fault code> <action>
```

Variables

Variable	Definition
<event type>	MSG, SET, or CLR

Variable	Definition
<fault code>	the 8-digit number that identifies the type of alarm
<action>	a code identifying the special action that is to be taken for this alarm. The possible action codes are as follows: • 0 no action (not an exception); that is, allow the alarm to pass • 1 ignore alarm; that is, throw the alarm away • 30 set severity to CRITICAL • 31 set severity to MAJOR • 32 set severity to MINOR • 33 set severity to WARNING There are other action codes that are used internally by NMDR.

Exit codes

Exit codes for the NMDR server are shown in the following table.

Table 44
Exit codes for the NMDR server

Exit code	Description
0	Successful exit. Do not restart the server.
1	Failure. Restart the server.
50	Do not restart (for backward compatibility)
51	Memory resource error (For example, not enough memory). No restart.
52	Disk resource error (For example, not enough space, no file). No restart.
53	Communication resource error (For example, cannot connect or register). No restart.
(Sheet 1 of 2)	

Table 44 (Continued)
Exit codes for the NMDR server

Exit code	Description
54	Timeout/deadlock/congestion (For example, too much congestion). No restart.
55	Bad command line arguments. No restart.
56	Bad configuration file or environment. No restart.
57	Fork exec failure. No restart.
58	Manual server shutdown (admin?). No restart.
59	ipc_init unsuccessful. No restart.
60	ipc service cannot be registered. No restart.
61	Exit signal received. No restart.Other unclassified errors. Restart the server.
100	Query problem to HGDS.
(Sheet 2 of 2)	

Error messages

Error messages output to the System Log Display for the NMDR server are shown in the following table:

Table 45
Error messages for the NMDR server

Error message	Meaning and action
NMDR_<name> -- Invalid command line argument.	Fatal. Correct the command line arguments in the definition for the server with the Server Administration tool.
NMDR_<name> -- Missing mandatory argument	Fatal. One of the following is missing from the command line: -g <passport group> -u <user id> -p <password> Correct the command line arguments in the definition of the server with the Server Administration tool.
NMDR_<name> -- Options -a and -B are incompatible.	Fatal. Correct the command line arguments in the definition of the server with the Server Administration tool.
NMDR_<name> -- Killed client <client> connection due to congestion.	Non-fatal. Check the logs for the problems relating to <client>.
NMDR_<name> -- Unexpectedly lost client <client> connection; error code: <code>	Non-fatal. Check the logs for problems relating to <client>.
NMDR_<name> -- Failed to connect to HGDS.	Fatal. Check the status of the HGDS server with the Server Administration tool.
NMDR_<name> -- Lost HGDS connection.	Fatal. Check the status of the HGDS server with the Server Administration tool.
NMDR_<name> -- Error response from HGDS.	Fatal. Check the HGDS.cfg file for invalid entries.
NMDR_<name> -- connection to FDTM failed... retry in 2 minutes.	Non-fatal. Check the status of the FDTM server with the Server Administration tool. Restart FDTM, if necessary.
NMDR_<name> -- error from FDTM: <error>.	Fatal. Check the status of the FDTM server with the Server Administration tool.
NMDR_<name> -- comm error with FDTM.	Fatal. Check the status of the FDTM server with the Server Administration tool. Try restarting the FDTM server.
(Sheet 1 of 5)	

Table 45 (Continued)
Error messages for the NMDR server

Error message	Meaning and action
NMDR_<name> -- lost connection to FDTM.	Fatal. Check the status of the FDTM server with the Server Administration tool. Restart FDTM, if necessary.
NMDR_<name> -- could not connect to FDTR: <name>	Fatal. Check the logs for the FDTR problem.
NMDR_<name> -- could not create session with FDTR: <name>	Fatal. Check the logs for the FDTR problem.
NMDR_<name> -- is now connected to <fdtr>.	Message only.
NMDR_<name> -- lost connection to <fdtr>.	Fatal. Check the logs for the FDTR problem.
NMDR_<name> -- error response from <fdtr>: <error>.	Fatal. Check the logs for the FDTR problem.
NMDR_<name> -- unknown response type from <fdtr>.	Fatal. Check the logs for the FDTR problem.
NMDR_<name> -- error response from <fdtr>.	Fatal. Check the logs for the FDTR problem.
NMDR_<name> -- error response on state walk for <component>: <error>.	Non-fatal. The state walk has been discontinued for this component. Check the FDTR logs for additional information.
NMDR_<name> -- Unable to find NMDR_[ppc|ppe|legacy].cfg file.	Fatal. Ensure that there is an NMDR_[ppc|ppe|legacy].cfg file in directory /opt/MagellanNMS/lib/cfg.
NMDR_<name> -- Unable to find /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. NMDR has been started with circuit monitoring enabled but the circuit monitoring cfg file does not exist. Without the file, no components can be circuit monitored. Create the file; it can be empty.
NMDR_<name> -- no interval specified at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
(Sheet 2 of 5)	

Table 45 (Continued)
Error messages for the NMDR server

Error message	Meaning and action
NMDR_<name> -- Unrecognized label at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
NMDR_<name> -- <instance> is not a legal type for circuit monitoring at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
NMDR_<name> -- Interval is not a positive integer at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
NMDR_<name> -- no instance specified at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
NMDR_<name> -- Second INSTANCE definition at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
NMDR_<name> -- Second INTERVAL definition at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
NMDR_<name> -- Invalid instance type: <instance> at line <x> in file /opt/MagellanNMS/cfg/NMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger NMDR to reparse and restart circuit monitoring.
(Sheet 3 of 5)	

Table 45 (Continued)
Error messages for the NMDR server

Error message	Meaning and action
NMDR_<name> -- EM <nodename> is not a member of this group.	Non-fatal. The NMDR server is reporting on module EM/<nodename> which is not defined in group <name> in /opt/MagellanNMS/cfg/HGDS.cfg. This is likely caused by a mismatch between the name provisioned on a node and the name entered in the HGDS.cfg file. Correct the HGDS.cfg file to match the name provisioned and restart HGDS, FDTM, and NMDR_<name> servers.
NMDR <name> -- EM <module name>: Either the module name on switch does not match the module name in the HGDS server or the on switch boot activation is not complete.	Non-fatal. If the name given to the node does not match the name that is specified in the /opt/MagellanNMS/cfg/HGDS.cfg file, update the HGDS.cfg file, and then stop and restart the NMDR server. If the names match, a state walk will be done automatically once the node has fully activated.
NMDR <name> -- error response on state walk for <instance> APPLICATION ERROR 1120 Fdtr received an invalid response from device Sw. Component is owned by another customer.	Non-fatal. NMDR is using the -M command line option with a value different than 0. NMDR cannot discover the Passport family type (see "Detection of Passport family" (page 392)). NMDR will use the default schemas for state walk, and it will not be able to retrieve any active alarm list.
NMDR <name> -- Unable to retrieve the Active Alarm List from module <instance>, proxy alarms will be generated.	Non-fatal. The node supports the Active Alarm List feature, but NMDR cannot obtain the active alarms because too many NMDRs requested the alarms at the same time. Proxy alarms are generated to represent the out-of-service components.
NMDR <name> -- The Active Alarm List from node <instance> was retrieved.	Non-fatal. The previous log message was generated. NMDR completed two state walks on the module, and it retrieved the active alarm list from the node.
NMDR <name> -- State walk started for every module of the group.	Non-fatal. One state walk has started for each module of the NMDR group.
NMDR <name> -- State walk started on module EM <module name>.	Non-fatal. One state walk has started for the module.
(Sheet 4 of 5)	

Table 45 (Continued)
Error messages for the NMDR server

Error message	Meaning and action
NMDR <name> -- State walk on module EM <module name> cancelled.	Non-fatal. The state walk in progress on the module was cancelled. The log is displayed for the following reasons: - loss of connectivity to the node while a state walk is in progress - manually triggering a Reset Database of NMDR, using GMDR Admin, while a state walk is in progress on that node - using GMDR Admin to delete a module while a state walk is in progress on that node - the name of the node does not match the name of the /opt/MagellanNMS/cfg/HGDS.cfg file.
NMDR <name> -- Walk already in progress or pending for module EM <module name>.	Non-fatal. A resynchronization was triggered on the module while it is currently being state walked.
NMDR <name> -- The babbler option (-B) was specified for this NMDR. This option is no longer supported and will be ignored.	Non-fatal. The option is ignored. NMDR execution continues. ***Note:*** The command line option -B (babbler) is ignored in this release. If any NMDR entry in the Server Administration tool uses this option, remove this option. For more information about the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.
NMDR <name> -- There was at least one query failure during the nodal state walk because of the following errors: <reason>. The reason could be 1121, 1122, 1137 and their combinations.	Non-fatal. Check the error summary for the FDTM server to diagnose the problem.
(Sheet 5 of 5)	

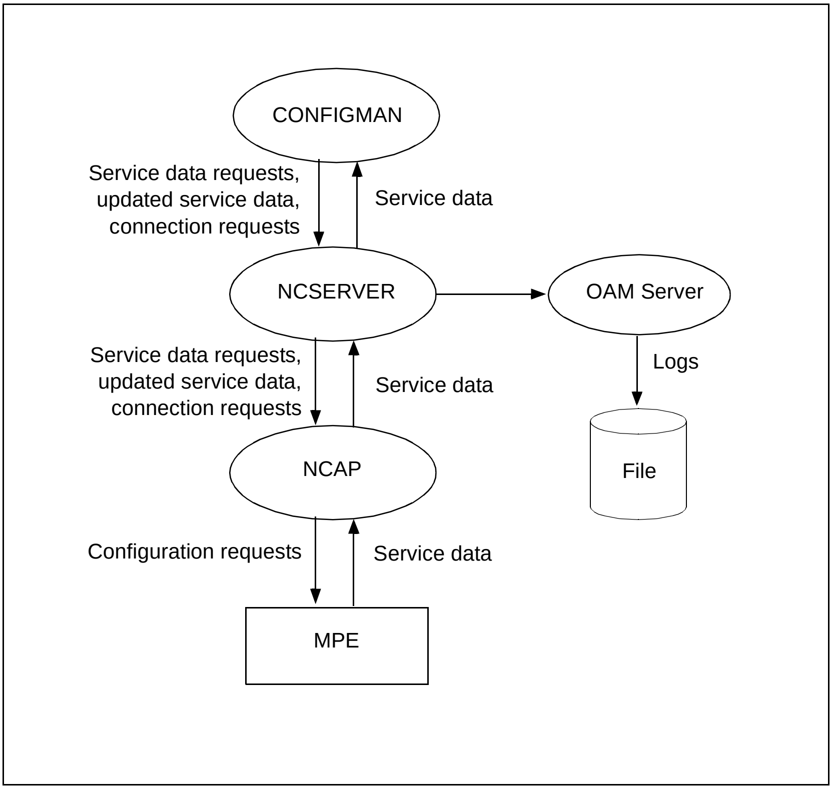
Chapter 37
MPE Nodal Provisioning Configuration Server (NCSERVER)

This section contains information on the Nortel Networks Multiservice Provider Edge (MPE) Nodal Provisioning Configuration server (NCSERVER). See the following topics for more information:

About the NCSERVER

The MPE Nodal Provisioning Configuration server (NCSERVER) is started and managed by the MDM Server Administrator (SVM) tool. This server is used by the Nodal Provisioning tool. It creates and manages the MPE Configuration Access Providers (NCAP) to provide configuration access to MPE network elements. For each open connection request received from the Configuration Manager, NCSERVER creates a new provider to handle configuration requests for a particular MPE. For more information on the Configuration Manager, see “Nodal Provisioning Configuration Manager (CONFIGMAN)” (page 331).

Figure 30
NCSERVER data flow and logging diagram


CONFIGMAN
Service data requests, updated service data, connection requests
Service data
NCSERVER
OAM Server
Logs
File
Service data requests, updated service data, connection requests
Service data
NCAP
Configuration requests
Service data
MPE

Managing the NCSERVER

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the NCSERVER. Any changes you make to the startup command or options take effect when the NCSERVER is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 301)
- “Startup command” (page 301)

Suggested name in Server Administration

The recommended name for the NCSERVER is MPE NP Config Server.

Configuring the NCSERVER with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

Use the following command to start the NCSERVER.

```
/opt/MagellanNMS/bin/ncserver [-h]
```

where:

`-h` displays information about the ncserver command.

Exit codes

Exit codes for when the NCSERVER terminates under its own control are shown in the following table. The NCSERVER can terminate with other exit codes if it aborts or is stopped by the Server Administration tool.

Table 46
Exit codes for the NCSERVER

Exit code	Description
0	Successful exit
51	Memory resource error
53	Failed to register service
55	Invalid argument
61	Exit signal received
62	Terminated due to licensing problem
100	Other unclassified errors

MPE Configuration Access Provider

The MPE Nodal Provisioning Configuration server (NCSERVER) creates and manages the MPE Configuration Access Provider (NCAP). NCAP is a client/server process that communicates with a specific MPE and provides all the functions related to configuration including loading, modifying, or saving the configuration view. This process is automatically started by NCSERVER when a configuration to an MPE is required.

Startup command

The NCAP process runs automatically when required. If the process stops and you need to manually restart, use the following command:

```
/opt/MagellanNMS/bin/ncap [-h]
```

where:

`-h` displays information about the ncap command.

Exit codes

Exit codes for when NCAP terminates under its own control are shown in the following table. NCAP can terminate with other exit codes if it aborts or is stopped by the Server Administration tool. The exit codes for NCAP are the same as those for the NCSERVER and are listed in the table “Exit codes for the NCSERVER” (page 301).

Chapter 38
Network Model Coordinator (DNMNMC)

This section contains information on the Network Model Coordinator (DNMNMC). See the following topics for more information:

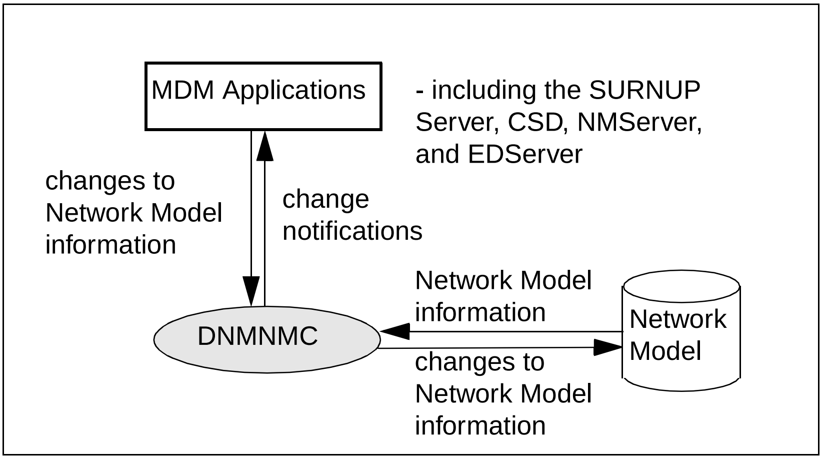
- “About the DNMNMC server” (page 303)
- “Managing the DNMNMC server” (page 304)
- “Configuration” (page 305)
- “Interdependencies” (page 305)
- “Exit codes” (page 305)
- “Error messages” (page 306)

About the DNMNMC server

The DNMNMC is the fault server responsible for coordinating access to the Network Model. The Network Model is stored in a shared memory segment that is used by a number of applications. The DNMNMC server ensures that applications that connect to the shared memory are informed when other applications make changes to the model, so that they may take appropriate action.

For example, before a network population tool reads in a new model, it must obtain a lock from the DNMNMC server. Fault tools will be informed of the lock, so that they do not attempt to read the model while it is being read in to shared memory. The DNMNMC server also provides a change notification service, so that if one application makes a change to the model, for example adding or deleting a component, the other applications can take appropriate action, for example, updating their displays to include or remove the component, respectively.

Figure 31
DNMNMC data flow diagram


MDM Applications
- including the SURNUP Server, CSD, NMServer, and EDServer
changes to Network Model information
change notifications
Network Model information
DNMNMC
Network Model
changes to Network Model information

Managing the DNMNMC server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for DNMNMC. Any changes you make to the startup command or options take effect when DNMNMC is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

See the following:

- “Suggested name in Server Administration” (page 304)
- “Startup command” (page 305)

Suggested name in Server Administration

The recommended name for the DNMNMC server is DNMNMC.

Configuring DMNNMC with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the DNMNMC server has the following syntax:

```
/opt/MagellanNMS/bin/dnmnmc [-s <shm size in MB>]
```

where:

`-s <shm size in MB>` is the maximum amount of shared memory to reserve. This number must be less than or equal to the maximum shared memory segment size configured in the workstation kernel.

If this option is not specified, the DNMNMC server reserves the largest possible shared memory segment available up to 24 Mbyte. To allocate a segment that is larger than 24 Mbyte, you need to specify the -s option.

Configuration

For information on configuring shared memory, see 241-6001-303 *Preside MDM Administrator Guide*.

Interdependencies

The DNMNMC server is responsible for initializing the shared memory segment for a Network Model. Therefore the DNMNMC server must be started before any other application that uses the model. This is ensured by positioning its entry in the file /opt/MagellanNMS/cfg/SVMList.cfg (Network Model Coordinator) before the other servers that use the model (DPN Network Model Updater, Network Model Server and Network Model Editing Server). Stopping the DNMNMC server using the Server Administration tool will cause all Network Model applications and servers to exit.

Exit codes

Exit codes for the DNMNMC server are shown in the following table.

Table 47
Exit codes for the DNMNMC server

Exit code	Description
1	Could not reserve the shared memory segment.
2	Model could not be loaded.
55	Invalid command line argument.

Error messages

Error messages for the DNMNMC server are shown in the following table.

Table 48
Error messages for the DNMNMC server

Error message	Meaning and action
Invalid command line argument.	Fatal, invalid command line argument provided. Revise the server configuration with the Server Administration tool.
Unable to initialize shared memory.	Fatal, unable to reserve the shared memory segment. The segment may be owned by another user (root).
NM Coordinator: Model is suspect. Exiting.	Fatal, model load failed and is assumed to be corrupted. Verify the model configuration.

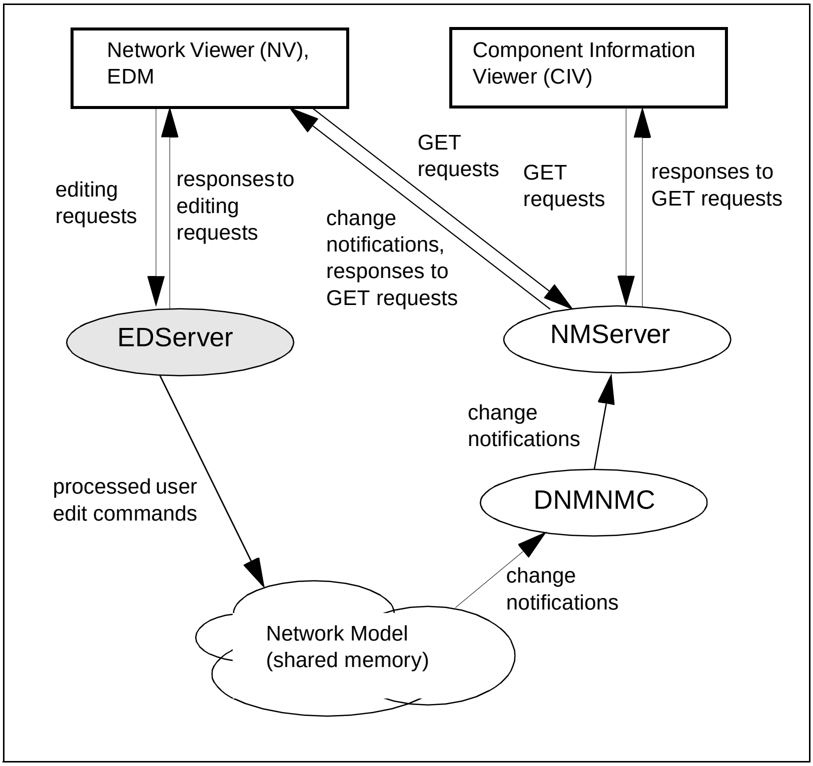
Chapter 39
Network Model Editing server (EDSERVER)

This section contains information on the Network Model Editing server (EDESERVER). See the following topics for more information:

About the EDESERVER

The EDSERVER lets you edit Network Models from the Network Viewer (NV). The EDSERVER, along with the NMSERVER, must be running to use NV in edit mode. Security must also be configured and there are multiple levels of security possibilities.

Figure 32
EDSERVER data flow diagram


Network Viewer (NV), EDM
Component Information Viewer (CIV)
editing requests
responses to editing requests
GET requests
change notifications, responses to GET requests
GET requests
responses to GET requests
EDServer
NMServer
change notifications
processed user edit commands
DNMNMC
change notifications
Network Model (shared memory)

Managing the EDESERVER

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the EDSERVER. Any changes you make to the server startup command or options take effect when the EDSERVER is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 309)
- “Startup command” (page 309)

Suggested name in Server Administration

The recommended name for the EDSERVER is NM Edit Server.

Configuring EDSERVER with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the EDSERVER has the following syntax:

```
/opt/MagellanNMS/bin/edserver [-p <password>] [-L] [-
h]
```

Variable definitions

Variable	Definition
`-p <password>`	is a password, or the full path of the file that contains the encrypted password. Password files are stored in the directory /opt/MagellanNMS/cfg/private. The password must be provided to NV when entering edit mode. If set, you must provide the specified password to be allowed to edit the model.
-L	specifies local model editing only. Only users running NV on the same workstation as the EDSERVER can edit Network Models.
-h	displays the help menu

The default values for the EDSERVER are remote access permission without a password.

Interdependencies

The EDSERVER relies on the DNMNMC server, and is always set up to run with the NMSERVER.

Exit codes

Exit codes for the EDSERVER are shown in the following table.

Table 49
Exit codes for the EDSERVER

Exit code	Description
1	Lost or no DNMNMC
2	Network Model load failed
3	Could not initialize the IPC environment
55	Invalid command line

Error messages

The error messages for the EDSERVER are shown in the following table.

Table 50
Error messages for the EDSERVER

Error message	Meaning and action
Lost connection to Network Model Coordinator	Fatal, lost DNMNMC. Make sure the server is running. The original cause for this may be an invalid network model being loaded or insufficient shared memory space.
Could not register with Network Model Coordinator - or - No Network Model Coordinator	Fatal, could not connect to DNMNMC. Make sure the server is running.
Invalid Network Model	Fatal, a corrupt Network Model was loaded. Verify the model configuration.

Chapter 40
Network Model Server (NMSERVER)

This section contains information on the Network Model server (NMSERVER). See the following topics for more information:

- “About the NMSERVER” (page 311)
- “Managing the NMSERVER” (page 312)
- “Interdependencies” (page 313)
- “Exit codes” (page 314)
- “Error messages” (page 314)

About the NMSERVER

The NMSERVER is the Preside Multiservice Data Manager server responsible for handling the API requests from the Network Model Provider. It also provides Network Model information to the fault tools. The NMSERVER is designed to give access to information about the active surveillance Network Model. The NMSERVER communicates with the Network Model Provider through the IPI interface, which is an IPC version of the API protocol as documented in the ESI primer.

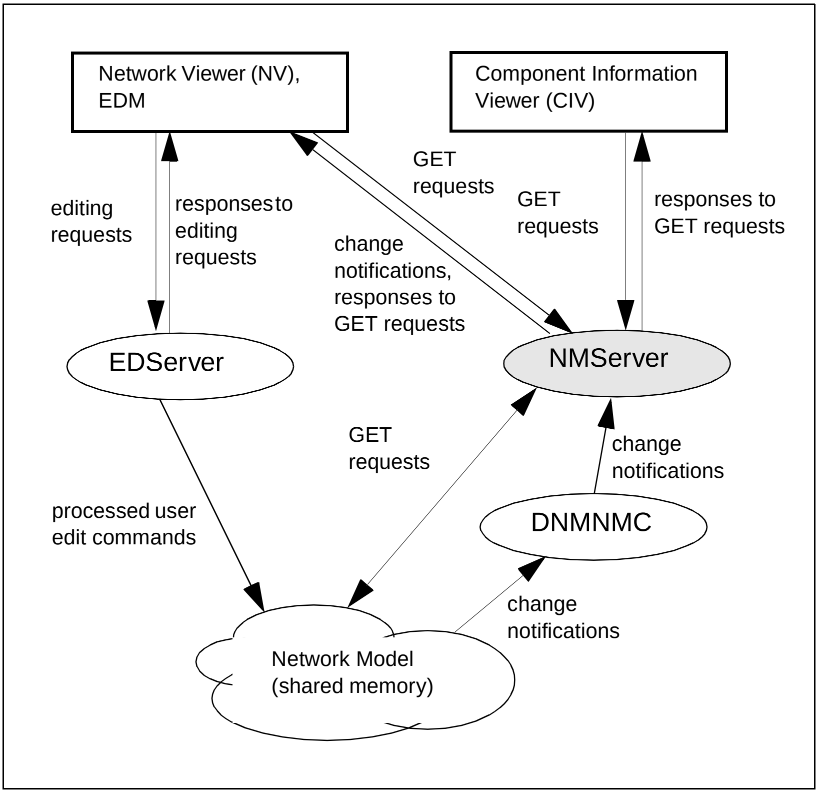
The NMSERVER provides different kinds of information about the active surveillance model such as:

- the model topology
- the raw and network states of the network components
- the network and raw state notifications of the network components

- the component attributes and their values

  Note: The maximum number of sieves that can be handled by the NMSERVER is fixed at 50.

Figure 33
NMSERVER data flow diagram


Network Viewer (NV), EDM
Component Information Viewer (CIV)
GET requests
editing requests
responses to editing requests
change notifications, responses to GET requests
GET requests
responses to GET requests
EDServer
NMServer
GET requests
change notifications
processed user edit commands
DNMNMC
change notifications
Network Model (shared memory)

Managing the NMSERVER

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the NMSERVER. Any changes you make to the server startup command or options take effect when the NMSERVER is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions on using the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 313)
- “Startup command” (page 313)

Suggested name in Server Administration

The recommended name for the NMSERVER is NM Server.

Configuring NMSERVER with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the NMSERVER has the following syntax:

```
/opt/MagellanNMS/bin/nmserver \
[-t <number of replies per thread pass>] \
[-c <message congestion threshold>]
```

where:

`-t <number of replies per thread pass>` is the thread concurrence factor in terms of numbers of replies forwarded before the next thread takes over (default 0).

`-c <message congestion threshold>` is the number of messages buffered due to congestion before the client connection is automatically dropped (default 3000).

Interdependencies

The NMSERVER relies on the Network Model Coordinator (DNMNMC) server to control access to the active surveillance model. The DNMNMC server must run on the same workstation as the NMSERVER. The DNMNMC server must be started before the NMSERVER is started.

The NMSERVER also relies on the Surveillance Network Model Updater (SURNUP) to provide its state notifications services. The NMSERVER must run on the same workstation as the SURNUP server. Therefore, Nortel Networks recommends that the SURNUP server be started before the NMSERVER.

The NMSERVER is used by the following items:

- Network Model API
- Network Viewer
- Component Information Viewer
- RNCS Alarm
- Accounting Gateway
- Network Status Bar
- Component Status Display

Exit codes

The exit codes for the NMSERVER are shown in the following table.

Table 51
Exit codes for the NMSERVER

Exit code	Description
1	lost or no DNMNMC
2	Network Model load failed
3	could not initialize the IPC environment
55	invalid command line

Error messages

The error messages for the NMSERVER are shown in the following table.

Table 52
Error messages for the NMSERVER

Error message	Meaning and action
Lost connection to Network Model Coordinator	Fatal, lost DNMNMC. Make sure the server is running. The original cause for this may be an invalid network model being loaded or insufficient shared memory space.
Could not register with Network Model Coordinator - or - No Network Model Coordinator	Fatal, could not connect to DNMNMC. Make sure the server is running.
Invalid Network Model	Fatal, a corrupt Network Model was loaded. Verify the model configuration

Chapter 41
Network Model Surveillance Updater (SURNUP)

This section contains information on the Surveillance Network Model Updater. See the following topics for more information:

About the SURNUP server

The SURNUP server is the Preside Multiservice Data Manager server responsible for keeping the active Network Model up to date with current component state information. The SURNUP server receives component state information from a GMDR server that maps states, component status, and alarms from the network into a raw component state. The SURNUP server takes the raw state value for a component and determines an overall state that takes into account the criticality of the component and the states of its related components. Whenever a component’s overall state changes, the SURNUP server propagates the new state to related components and recomputes their overall states if necessary.

The SURNUP server is also responsible for adding new components to the Network Model when state notifications arrive for components that do not exist in the model. This is called auto-population. Components of certain types are added to the model permanently. These are the same components that may be created in the model using the Network Viewer. Other components are added to the model only temporarily, for the period of time that they are experiencing a problem. These are called dynamic components. For more information on components and attributes, see 241-6001-015 *Preside MDM Network Model Administrator Guide* for a list of all component types, indicating which are dynamic.

Whenever the SURNUP server makes a change to the model, such as a state update or a component addition or deletion, it informs the Network Model surveillance display applications (Component Information Viewer, and Component Status Display) so that they may update their displays.

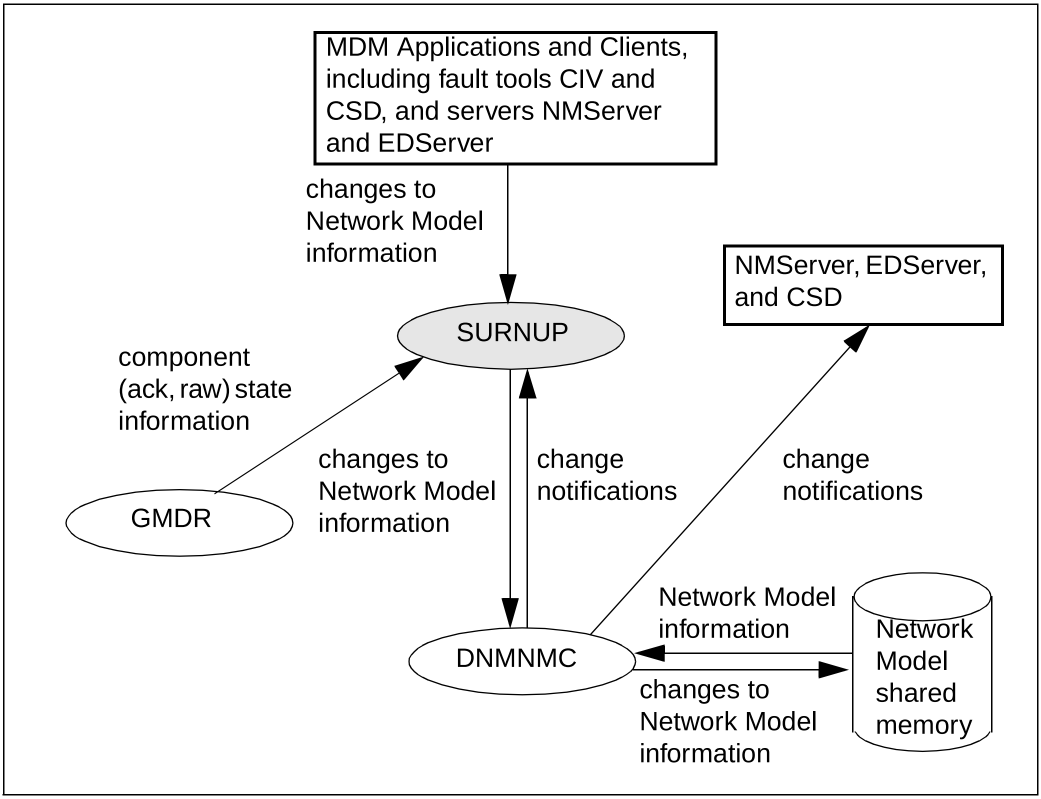
The SURNUP server receives new ackStateChange notifications from the GMDR server when components become acknowledged or unacknowledged. The SURNUP server propagates these notifications and updates the Network Model. The NV, CSD and NSB receive the required notifications from the Network Model. The SURNUP server maps the notifications to the action of manually acknowledging or unacknowledging a network state from one of the state-based surveillance tools such as NV or CSD.

The SURNUP server also has an option that allows obsolete components to be removed from the network model. A command line option provides the flexibility to decide on the level of component deletion (see “Obsolete component deletion options” (page 321)). In order to receive component delete notifications, SURNUP asks for server reset notifications from GMDR when it connects to GMDR. When a server reset component delete notification is received, the component delete type attribute value is extracted. If the type is permanent, then the component is removed from the network model, unless it is overridden by the command line option. If the attribute does not exists, then the delete notification is treated as transient, and the component is not removed from the network model.

Components are automatically deleted from the network model when a disconnected GMDR is reconnected - this is referred to as resynchronization-based automatic component deletion. For GMDR disconnects that are the result of administrator action, the components are deleted on any downstream GMDRs and are also deleted from the network model and fault applications using the fault API or EPI. For GMDR disconnects that are the result of communication failures, the components stay in the unknown state until the component indicates no alarms; the component is then eligible for deletion.

While an editing session or operation is being performed on the network model, component deletions are not applied. The events are queued and applied when the editing session or operation is finished.

Figure 34
SURNUP data flow diagram


MDM Applications and Clients, including fault tools CIV and CSD, and servers NMServer and EDServer
changes to Network Model information
NMServer, EDServer, and CSD
SURNUP
component (ack, raw) state information
changes to Network Model information
change notifications
change notifications
GMDR
Network Model information
Network Model shared memory
DNMNMC
changes to Network Model information

Startup options

You can use the startup command to specify the following options:

- “Auto-population options” (page 320)
- “State-determination options” (page 320)

See “Startup command” (page 322) for a detailed description of the startup options.

Auto-population options

Whenever the SURNUP server receives state information for a component that is not in the Network Model, it automatically adds the component to the model. Auto-population can be controlled by the following options.

- -A [all]
- -A b[ackbone]
- -A n[one]
- -A s[ubcomp]

These options do not affect dynamic components, which will continue to be temporarily added to the model while they are in the out-of-service or in-service-troubled states.

See “Startup command” (page 322) for a detailed description of the auto-population options.

State-determination options

State-determination can be controlled by the following options.

- -N
- -C
- -V

The -N option specifies how the SURNUP server determines overall state for organizational nodes (for example, regions and sites) for the case in which the node has an out-of-service subcomponent.

The -C option specifies customer network manager (CNM) mode. In this mode, unknown subcomponents are forced to the in-service state if one or more of their parents is set to the in-service state (For example, at initial population). The -C option also turns auto-population off, but auto-population can be turned back on by including the -A all option in the start up command.

See “Startup command” (page 322) for a detailed description of the -N, and -C options.

Obsolete component deletion options

Whenever the SURNUP server receives a component delete notification from GMDR, it can automatically delete the component from the network model. Obsolete component deletion has the following options:

- -D [all]
- -D all, nl
- -D n[one]
- -D s[ubcomp]
- -D s[ubcomp], nl
- -S

Component deletion in SURNUP and the network model applies to all types of components in the network model: ordinary, optional, endpoint, optional endpoint, and dynamic.

See “Startup command” (page 322) for a detailed description of the obsolete component delete options.

Managing the Surveillance Network Model Updater

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for SURNUP. Any changes you make to the server startup command or options take effect when SURNUP is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 322)
- “Startup command” (page 322)

Suggested name in Server Administration

The recommended name for the Network Model Surveillance Updater is SURNUP.

Configuring SURNUP with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command syntax for the SURNUP server has the following syntax:

```
/opt/MagellanNMS/bin/surnup [-C] [-U] [-V] [-N] \
[-A [all] |-A n[one] |-A b[ackbone] |-A s[ubcomp] \
[-D all [nl] |-D n[one] |-D s[ubcomp] [nl] \
[-H <GMDR host name>]] [-d <NM files directory> \
|-i <instances directory> \
[-t <types directory>]] [-I <GMDR service name>] [-P] \
[-T <criticality threshold>] \
[-l] [-q] [-e] [-S]
```

where:

`-C` selects the customer network management (CNM) mode. Autopopulation is turned off, and the initial states for subcomponents are set to in service when their parent component is up. If this option is not used, the initial states for these subcomponents are *unknown.*

To use the -C parameter with autopopulation turned on, type in parameter -A immediately after the -C parameter.

`-U` stops the SURNUP server from producing state change notifications to UNKNOWN that are the result of losing a connection to a network node. This option should be specified in networks that contain very large nodes (that have in the order of thousands of subcomponents) to reduce communications loads.

CAUTION

Risk of displaying inaccurate component states

Do not specify the -U option if Network Viewers from software releases of 10.1 or lower have been used to monitor the current Network Model. If you do, the Network Viewer will display inaccurate subcomponent states.

`-V` selects the customer network management (CNM) mode. Autopopulation is turned on, and the initial states for subcomponents are set to in service when their parent component is up. If this option is not used, the initial states for these subcomponents are *unknown.*

This option is the equivalent of entering the -C option along with the -A[all] option.

`-N` changes the way the SURNUP server determines the overall state for organizational nodes (for example, regions and sites) when the node has an out-of-service subcomponent. It allows operators that are monitoring at a high level to be made directly aware of sites and regions that contain modules that are out of service.

If you do not enter the -N option and an organizational node has at least one subcomponent that is out-of-service, the state of the node is set to in-service trouble. The criticality (severity) of the trouble is the highest criticality of all troubled subcomponents.

If you enter the -N option and an organizational node has at least one subcomponent that is out of service (OOS), the state of the node is OOS. The criticality of the outage is the highest criticality of all OOS subcomponents.

`-A [all]` directs the SURNUP server to auto-populate components of any component type. This is the default.

`-A n[one]` directs the SURNUP server not to auto-populate components of any type

`-A b[ackbone]` directs the SURNUP server to auto-populate only backbone subcomponents and network links and trunks. No module level components will be auto-populated when this option is turned on. Backbone subcomponents are defined with the BACKBONE flag in the Network Model schema.

Example:

PM: subcomponents down to and including PI.

NM: subcomponents down to and including SCAN.

EM: subcomponents such as shelves, cards, trunks, and DPN gateways

This includes dynamic subcomponents at the same levels as the above backbone subcomponents; for example, REDS and OFFICE under PM, and VOLUME and FQ under NM.

`-A s[ubcomp]` directs the SURNUP server to only autopopulate subcomponents of existing modules.

`-D [all]` directs the SURNUP server to automatically delete components from the network model on receiving a component deletion notification with the permanent delete attribute. The extent of deletion depends on the level of component deletion notification, as follows:

- on modules, a hierarchical delete including all links is performed
- on a module level link, the deletion is performed
- on subcomponents, a hierarchical delete including all link end-points is performed
- on a subcomponent link, the deletion is performed

`-D all, nl` directs the SURNUP server to automatically delete components from the network model on receiving a component deletion notification with the permanent delete attribute. Links and their end-points are not deleted. The extent of deletion depends on the level of component deletion notification, as follows:

- on modules, a hierarchical delete including all links is performed
- on a module level link, the link and end-points are not deleted
- on subcomponents, a hierarchical delete is performed except for link end-points and the parent components of the end-points
- on a subcomponent link, the link and end-points are not deleted

`-D n[one]` directs the SURNUP server not to delete components of any type (including modules, subcomponents, and links).

> ***Note:*** This does not include dynamic components deletion when the state changes to INSV.

`-D s[ubcomp]` directs the SURNUP server to automatically delete only components representing subcomponents and links from the network model on receiving a component deletion notification with the permanent delete attribute. This is the default component deletion option. The extent of deletion depends on the level of component deletion notification, as follows:

- on modules, the deletion is not performed
- on a module level link, the link and end-points are not deleted
- on subcomponents, a hierarchical delete including all links is performed
- on a subcomponent link, the link and end-points are not deleted

> ***Note:*** Modules are never automatically deleted from the network model.

`-D s[ubcomp], nl` directs the SURNUP server to automatically delete only components representing subcomponents from the network model on receiving a component deletion notification with the permanent delete attribute. Links and their end-points are not deleted. The extent of deletion depends on the level of component deletion notification, as follows:

- on modules, the deletion is not performed

- on a module level link, the link and end-points are not deleted
- on subcomponents, a hierarchical delete is performed except for link end-points and the parent components of the end-points
- on a subcomponent link, the link and end-points are not deleted

Note: Modules are never automatically deleted from the network model.

`-H` if specified, names the workstation that runs the GMDR server to be used as a data source. If this option is not specified, the workstation-wide service selection for surveillance is used.

`-d <NM files directory>` directs the SURNUP server to use network model types and instances files in <NM files directory>

`-i <instances directory>` directs the SURNUP server to use network model instances files in <instances directory>

`-t <types directory>` directs the SURNUP server to use network model types files in <types directory>

`-I <GMDR Service name>` specifies the GMDR service name to connect to

`-T <criticality threshold>` specifies the minimum component criticality value that a component must have if the GMDR server is to supply SURNUP with state change information for that component. Only information about components with a GMDR component criticality value higher or equal to the threshold are reported to SURNUP, and are therefore managed by SURNUP. The GMDR component criticality specified with this option is not the same as the Network Model criticality.

`-l` generates all logs for missing components (default off)

`-q` generates the first log for a missing component (default off)

`-e` prints all error logs on standard output

`-S` any component GMDR does not know about is deleted from SURNUP. This only occurs if GMDR has heard from the device which contains the UNK component. If GMDR has not heard from the device, it will not delete any subcomponents.

Interdependencies

The SURNUP server relies on the Network Model Coordinator (DNMNMC) server to control access to the active Network Model and broadcast state change notifications. The DNMNMC server must run on the same workstation as the SURNUP server. The DNMNMC server must be started before the SURNUP server is started.

The SURNUP server relies on the General Management Data Router (GMDR) server to provide it with raw state information and component deletion information. Note that the GMDR server does not have to run on the same workstation as the SURNUP server. Workstation-wide service selection allows the SURNUP server to communicate with GMDR servers that run on other workstations in a LAN, unless the -H option is used. For details on establishing level 2 MNS domains, see the section on configuring the Multi-nodal Naming Service domains in 241-6001-303 *Preside MDM Administrator Guide*. For details on selecting remote services, see the section on using the Service Selection tool in 241-6001-303 *Preside MDM Administrator Guide*.

When the SURNUP server starts and there is an active network model, it automatically loads the committed model unless either the -i or -d option is specified, in which case the corresponding model is loaded instead.

Configuration

Configuring the SURNUP server consists solely of optional parameters and the presence of valid Network Model instance files. Network Model files are expected to reside in the file system directory /opt/MagellanNMS/data/model/nmf.

Run-time options are specified using the Server Administration tool. For more information, see 241-6001-303 *Preside MDM Administrator Guide*.

Exit codes

Exit codes for the SURNUP server are shown in the following table.

Table 53
Exit codes for the SURNUP server

Exit code	Description
0	Help invoked
1	Lost or no DNMNMC.
2	Could not load the Network model.
3	Unlikely, problem with Network Model.
6	An input/output error has occurred because the program is trying to do something illegal with the hardware, for example, trying to free a chunk of memory more than once.This may result in the pointer referencing an invalid memory address and generating an exit code 11. The pointer could also reference a valid memory address that was already used for something else. This results in a memory corruption, resulting in almost any kind of error, such as exit code 6.
11	A segmentation error has occurred because a program has tried to access an invalid memory address.
50	Another SURNUP server is already running. SURNUP aborts with a core dump when it runs out of memory. A log containing the following message is generated before SURNUP aborts: SURNUP: Ran out of memory.
55	Invalid arguments
62	Terminated due to licencing problem (see MDM Logs). Look at the System Log Display tool for logs about licencing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.

Error messages

Error messages for the SURNUP server are shown in the following table.

Table 54
Error messages for the SURNUP server

Error message	Meaning and action
SURNUP: Could not connect to IPI Service “<service>” on host “<host>”	Not fatal, SURNUP could not connect to its data server (For example, GMDR). Make sure the server is running and host/service selection is setup correctly
SURNUP: Problem Manager Interface: SURNUP cannot connect to EMNMSERV	Not fatal. Make sure the server is running and host/service selection is setup correctly.
SURNUP: Lost connection to Network Model Coordinator.	Fatal, SURNUP lost connection DNMNMC. Make sure the server is running. The original cause for this may be an invalid network model being loaded or insufficient shared memory space.
SURNUP: Invalid state received from data manager/IPI server for <component> : <state>	Unlikely, can be ignored, not fatal.
SURNUP: Invalid state received from problem manager for <component> : <state>	Unlikely, can be ignored, not fatal.
SURNUP: Ran out of memory.	The server will restart abort (w core dump) and restart.
SURNUP: Model load warning. Object not created: <Network Model warning>	Not fatal, problems while loading the Model. Investigate the message and ignore if not major.
SURNUP: Failed to register ipc service.	Fatal, another SURNUP is probably running or MNSD is down. Revise the server configuration.
SURNUP: Invalid startup option	Fatal, bad command line option. Revise the server configuration with the Server Administration tool.
SURNUP: Failed to connect to NM Coordinator.	Fatal, the Network Model Coordinator is down. Revise the server configuration with the Server Administration tool.
(Sheet 1 of 2)	

Table 54 (Continued)
Error messages for the SURNUP server

Error message	Meaning and action
SURNUP: Failed to initialize model data.	Fatal, the model could not be loaded. Check the model configuration
SURNUP: Invalid data file directory: <type directory> or <instance directory>	Unlikely, fatal, invalid run time environment.
SURNUP: Invalid component id: <component>	Not fatal, unrecognized component name received from GMDR. Generated only if -l or -q provided on the command line.
SURNUP: Component not found in NW model: <component>	Not fatal, parent component not in model. Generated only if -l or -q provided on the command line.
SURNUP - Unable to establish a licensing context	The SURNUP server licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
SURNUP - License request failed: <reason>	A run-time license cannot be allocated to the SURNUP server for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running SURNUP.
SURNUP - License warning: <reason>	The license required to run SURNUP is about to expire. ContactNortel Networks immediately.
(Sheet 2 of 2)	

Chapter 42

Nodal Provisioning Configuration Manager (CONFIGMAN)

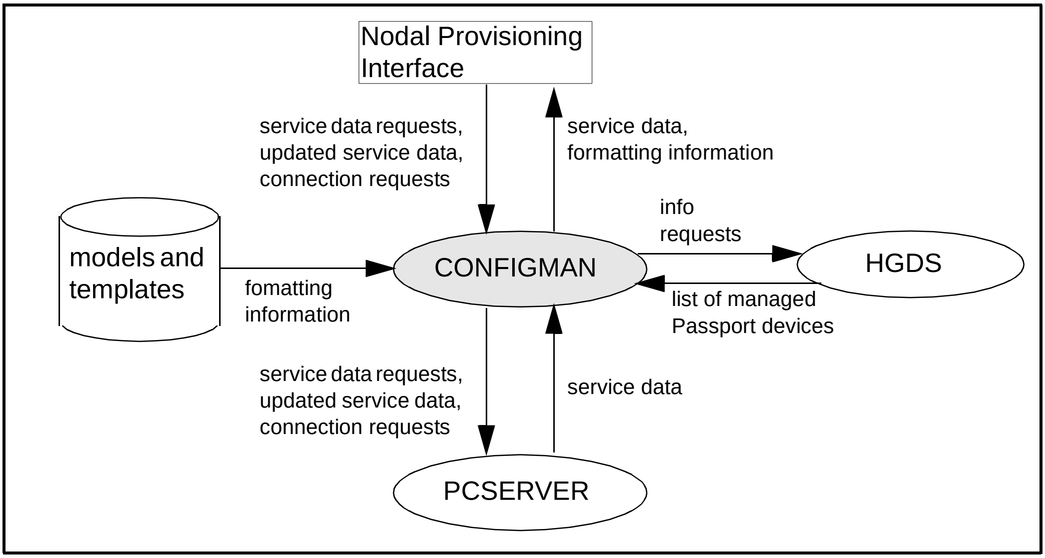
This section contains information on the Nodal Provisioning Configuration Manager (CONFIGMAN), and contains the following topics:

About the Configuration Manager

The Configuration Manager (CONFIGMAN) provides configuration services for the nodal provisioning interface. The interface sends requests and updated service data through CONFIGMAN and PCSERVER to the Passport device. See “Passport Nodal Provisioning Configuration Server (PCSERVER)” (page 415) for more information. CONFIGMAN also communicates with the Host Group Directory server (HGDS) to obtain a list of managed devices.

When you create or modify a service template, you can specify the software version range applicable to the connected device. By specifying the range, CONFIGMAN uses the nodal provisioning version checking tool to filter out non-applicable service templates. Based on the deviceVersion node element specified in the service template XML file, COMFIGMAN sends only valid templates to be displayed in the Nodal Provisioning window.

Figure 35
CONFIGMAN data flow diagram


Nodal Provisioning Interface
service data requests, updated service data, connection requests
service data, formatting information
info requests
models and templates
fomatting information
CONFIGMAN
HGDS
list of managed Passport devices
service data requests, updated service data, connection requests
service data
PCSERVER

Managing the CONFIGMAN server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for CONFIGMAN. Any changes you make to the startup command or options take effect when CONFIGMAN is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

See the following information:

- “Suggested name in Server Administration” (page 332)
- “Startup command” (page 333)

Suggested name in Server Administration

The recommended name for the CONFIGMAN server is NP Config Manager.

Configuring the Configuration Manager with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

Use the following command to start CONFIGMAN.

```
/opt/MagellanNMS/bin/configman
  [-Dcshost=<IP address>|<host name>]
  [-Dlog=<log filename>] [-Dcfg=<config filename>]
  [-Dlogview=true] [-Dverbose=true] [-Dhelp=true]
  [-port=<port number>]
```

where:

`-Dcshost=<IP address>|<host name>` specifies either the IP address or the host name of the Passport Configuration server (PCSERVER)

`-Dlog=<log filename>` lets you redirect errors to a log file

`-Dcfg=<config filename>` specifies the name of a configuration file that has default values for CONFIGMAN parameters and user preferences

`-Dlogview=true` opens the Configuration Manager logging application, which is a graphical window that displays all generated messages and lets you save the messages to a file

`-Dverbose=true` causes all logging messages to be written to STDOUT

`-Dhelp=true` displays information about the configman command

`-port=<port number>` specifies the port number assigned to the server

Exit codes

Exit codes for when CONFIGMAN terminates under its own control are shown in the following table. CONFIGMAN can terminate with other exit codes if it crashes or is stopped by the Server Administration tool.

Table 55
Exit codes for CONFIGMAN

Exit code	Description
0	Successful exit
50	Failed to initialize
51	Memory resource error
55	Invalid argument
56	Unable to locate JRE path
61	Exit signal received
65	context server (CTXSVR) unavailable and unable to register the port with MNSD. Ensure that MNSD and MNSD Agent are running.
100	Other unclassified errors

Chapter 43

Passport 4400 Backup Provider (PBCKPP4400)

This section contains information about the Passport 4400 Backup Provider (PBCKPP4400). See the following sections for information about this server:

- “About the PBCKPP4400” (page 335)
- “Managing the PBCKPP4400” (page 337)
- “Interdependencies” (page 339)

About the PBCKPP4400

The PBCKPP4400 sends requests to a Passport 4400 device to obtain a backup image in response to a request from the Backup Server (a backup controller).

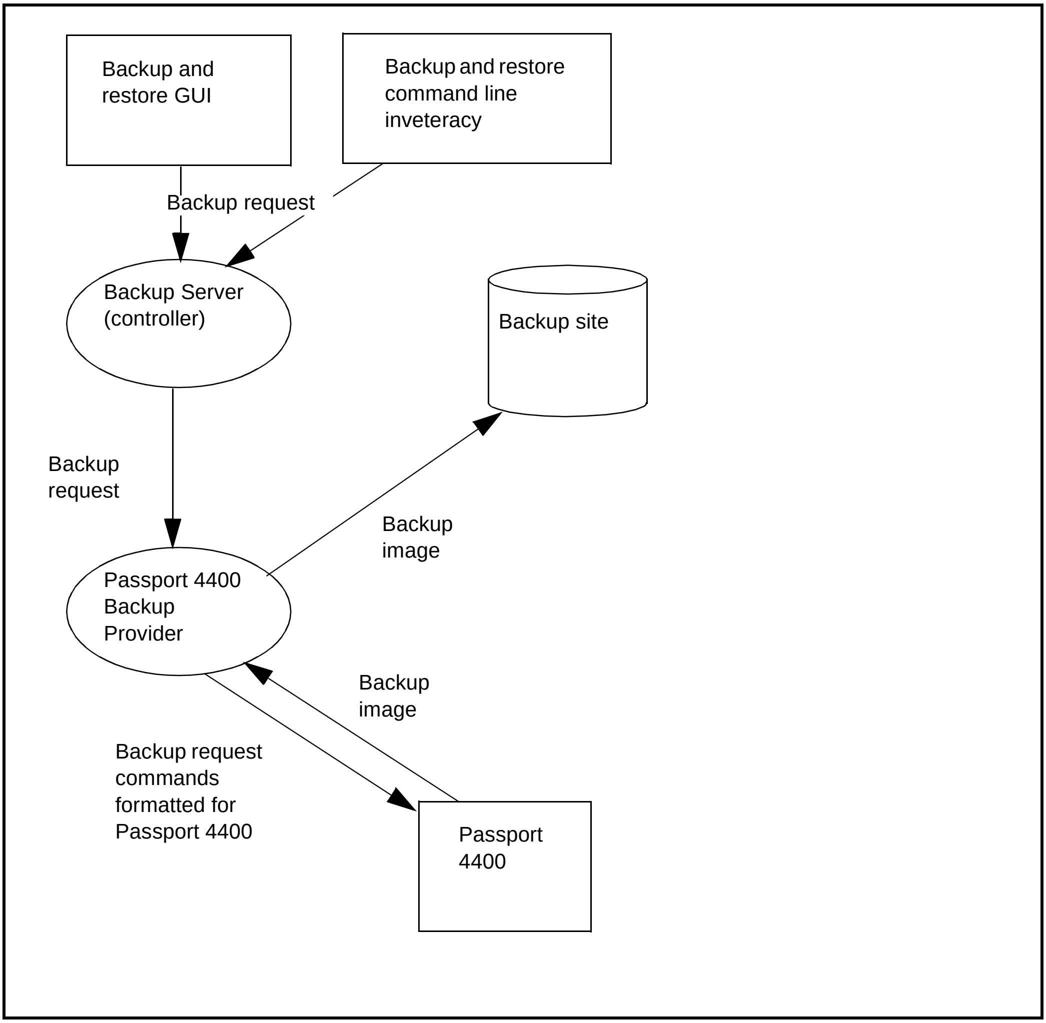
See Figure 36 for a dataflow diagram that shows the items related to the PBCKPP4400.

Here is an abridged version of the backup process:

1. A user sets up a request to perform a backup using the Passport/SNMP Backup and Restore tool or the command line interface. The tool or command line sends this request to the Backup Server (a controller)
2. The Backup Server sends a request to a backup provider. If the node being backed up is a Passport 4400, the controller sends the request to a PBCKPP4400.
3. The PBCKPP4400 formats the request for the Passport 4400 and forwards it to the Passport 4400.

4 The Passport 4400 provides access to the image, and the Backup Provider routes a copy of the image to the backup site specified by the Backup Server.

Figure 36
Dataflow diagram for the PBCKPP4400


Backup and restore GUI
Backup and restore command line inveteracy
Backup request
Backup Server (controller)
Backup site
Backup request
Backup image
Passport 4400 Backup Provider
Backup image
Backup request commands formatted for Passport 4400
Passport 4400

Managing the PBCKPP4400

See the following sections for information about managing the PBCKPP4400:

- “Configuration” (page 337)
- “Suggested name in Server Administration” (page 337)
- “Start-up command” (page 337)

Configuration

Configuring the Passport 4400 Backup Provider is part of a much larger task: Configuring Passport/SNMP Backup and Restore. For the instructions to do this, see 241-6001-807 *Preside MDM Network Backup and Restore*.

Suggested name in Server Administration

The recommended name to enter in the Server Manager Administration tool for the PBCKPP4400 is PP4400 Backup Provider.

Configuring the PBCKPP4400 with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the PBCKPP4400 has the following syntax:

```
/opt/MagellanNMS/bin/pbckpp4400
  [-m <interface_mapping_file>] [-p <port_no>]
```

where:

`interface_mapping_file` is the name of the interface mapping file. This optional file is required when your system has multiple network interfaces. The default interface mapping file is /opt/MagellanNMS/cfg/ifmap.cfg.

If the file ifmap.cfg exists in directory /opt/MagellanNMS/cfg and is populated with valid mapping information, it is not necessary to specify the `-m` option. This file is used by default.

`port_no` is the port number that the Provider uses.
The default port for the PBCKPP4400 is port 5030.

Interface mapping file (ifmap.cfg)

The Provider host machine can have multiple network interfaces. For example, the system can have one interface to the LAN and other interfaces to the WAN where the Passport 4400/4460 devices reside. In this configuration the Passport 4400/4460 devices cannot see the IP address of the interface to the LAN. In this situation you need to configure the Provider to use the correct interface address for the TFTP connection. You do this by creating an interface mapping file.

The Passport 4400 and Passport 4460 Providers can use the interface mapping file to determine the IP address used as the TFTP server address. The Passport 4400/4460 device connects to the TFTP server address for TFTP file transfers. Each line in the interface mapping file defines the mapping of the host address and the Passport 4400/4460 address or addresses. The host address is the TFTP server address.

The interface mapping file is named ifmap.cfg.

The format of ifmap.cfg is

```
<interface IP address> <device IP address(es)>
```

where:

`interface IP address` is the IP address of the interface to be used and has the format n.n.n.n.

`device IP address(es)` is the IP addresses of the devices and has the format n.n.n.n. You can match a group of devices by using the wildcard character (*).

You can include comments in the interface mapping file by inserting an octothorpe (#) at the beginning of the line. You can also include blank lines.

Example

The following example shows an interface mapping file for a host machine that has three network interfaces: one connects to the LAN and the other two are connected to separate networks of Passport 4400/4460 devices.

```
# Interface mapping file
# The last entry is the interface address to the LAN,
# which needs to map to any device address.
# This last entry can be omitted.

131.147.0.1 131.147.*
131.148.0.1 131.148.*
32.123.1.1  *
```

Interdependencies

Preside Multiservice Data Manager software must be installed and configured, and the Backup Server (controller) must be configured and running.

Chapter 44

Passport 4400 Restore Provider (PRSTPP4400)

This section contains information about the Passport 4400 Restore Provider (PRSTPP4400). See the following sections for information about this server:

- “About the PRSTPP4400” (page 341)
- “Managing the PRSTPP4400” (page 342)
- “Interdependencies” (page 345)

About the PRSTPP4400

The PRSTPP4400 sends a backup image to a Passport 4400 image in response to a request from the Restore Controller.

See “Dataflow diagram for the PRSTPP4400” (page 342) for a dataflow diagram that shows the items related to the PRSTPP4400.

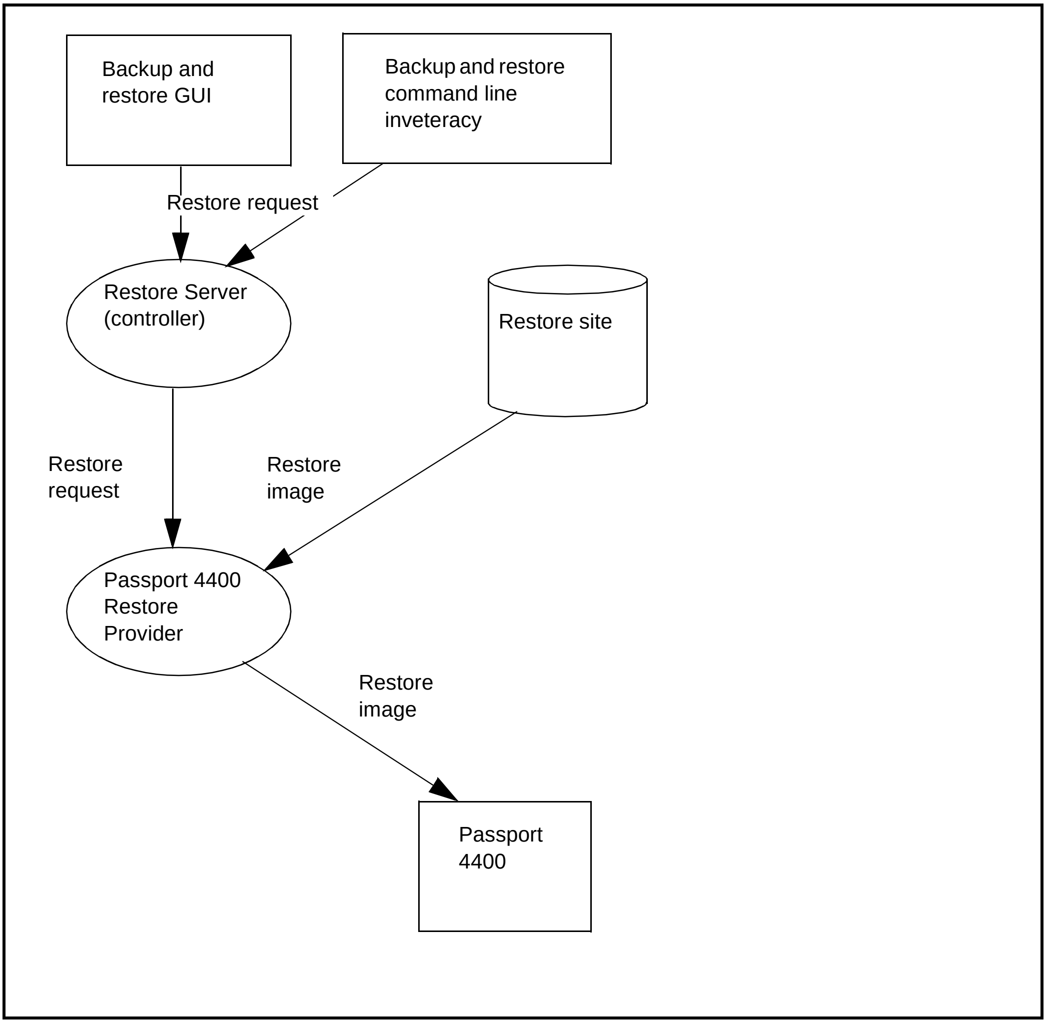
Here is an abridged version of the restore process:

1 A user sets up a request to perform a restore using the Passport/SNMP Backup and Restore tool or the command line interface. The tool or command line sends this request to the Restore Controller.

2 The Restore Controller sends a request to a Restore Provider. If the node being restored up is a Passport 4400, the controller sends the request to a PRSTPP4400.

3 The PRSTPP4400 obtains the backup image from the distribution site and forwards it to the Passport 4400.

Figure 37
Dataflow diagram for the PRSTPP4400



Managing the PRSTPP4400

See the following sections for information about managing the PRSTPP4400:

- “Configuration” (page 343)
- “Suggested name in Server Administration” (page 343)

- “Start-up command” (page 343)

Configuration

Configuring the Passport 4400 Restore Provider is part of a much larger task: Configuring Passport/SNMP Backup and Restore. For the instructions to do this, see 241-6001-807 *Preside MDM Network Backup and Restore*.

Suggested name in Server Administration

The recommended name to enter in the Server Manager Administration tool for the Passport 4400 Restore Provider is PP4400 Restore Provider.

Configuring the PRSTPP4400 with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the PRSTPP4400 has the following syntax:

```
/opt/MagellanNMS/bin/prstpp4400
  [-m <interface_mapping_file>] [-p <port_no>]
```

where:

`interface_mapping_file` is the name of the interface mapping file. This optional file is required when your system has multiple network interfaces. The default interface mapping file is /opt/MagellanNMS/cfg/ifmap.cfg.

If the file ifmap.cfg exists in directory /opt/MagellanNMS/cfg and is populated with valid mapping information, it is not necessary to specify the `-m` option. This file is used by default.

`port_no` is the port number that the Provider uses.
The default port for the PRSTPP4400 is port 5031.

Interface mapping file (ifmap.cfg)

The Provider host machine can have multiple network interfaces. For example, the system can have one interface to the LAN and other interfaces to the WAN where the Passport 4400/4460 devices reside. In this configuration the Passport 4400/4460 devices cannot see the IP address of the interface to the LAN. In this situation you need to configure the Provider to use the correct interface address for the TFTP connection. You do this by creating an interface mapping file.

The Passport 4400 and Passport 4460 Providers can use the interface mapping file to determine the IP address used as the TFTP server address. The Passport 4400/4460 device connects to the TFTP server address for TFTP file transfers. Each line in the interface mapping file defines the mapping of the host address and the Passport 4400/4460 address or addresses. The host address is the TFTP server address.

The interface mapping file is named ifmap.cfg.

The format of ifmap.cfg is

```
<interface IP address> <device IP address(es)>
```

where:

`interface IP address` is the IP address of the interface to be used and has the format n.n.n.n.

`device IP address(es)` is the IP addresses of the devices and has the format n.n.n.n. You can match a group of devices by using the wildcard character (*).

You can include comments in the interface mapping file by inserting an octothorpe (#) at the beginning of the line. You can also include blank lines.

Example

The following example shows an interface mapping file for a host machine that has 3 network interfaces: one connects to the LAN and the other two are connected to separate networks of Passport 4400/4460 devices.

```
# Interface mapping file
# The last entry is the interface address to the LAN,
# which needs to map to any device address.
# This last entry can be omitted.
```

```
131.147.0.1 131.147.*
131.148.0.1 131.148.*
32.123.1.1  *
```

Interdependencies

The Preside Multiservice Data Manager software must be installed and configured, and the Restore Controller must be configured and running.

Chapter 45

Passport 4460 Backup Provider (PBCKPP4460)

This section contains information about the Passport 4460 Backup Provider (PBCKPP4460). See the following sections for information about this server:

- “About the PBCKPP4460” (page 347)
- “Managing the PBCKPP4460” (page 349)
- “Interdependencies” (page 351)

About the PBCKPP4460

The PBCKPP4460 sends requests to a Passport 4460 device to obtain a backup image in response to a request from the Backup Server (a backup controller).

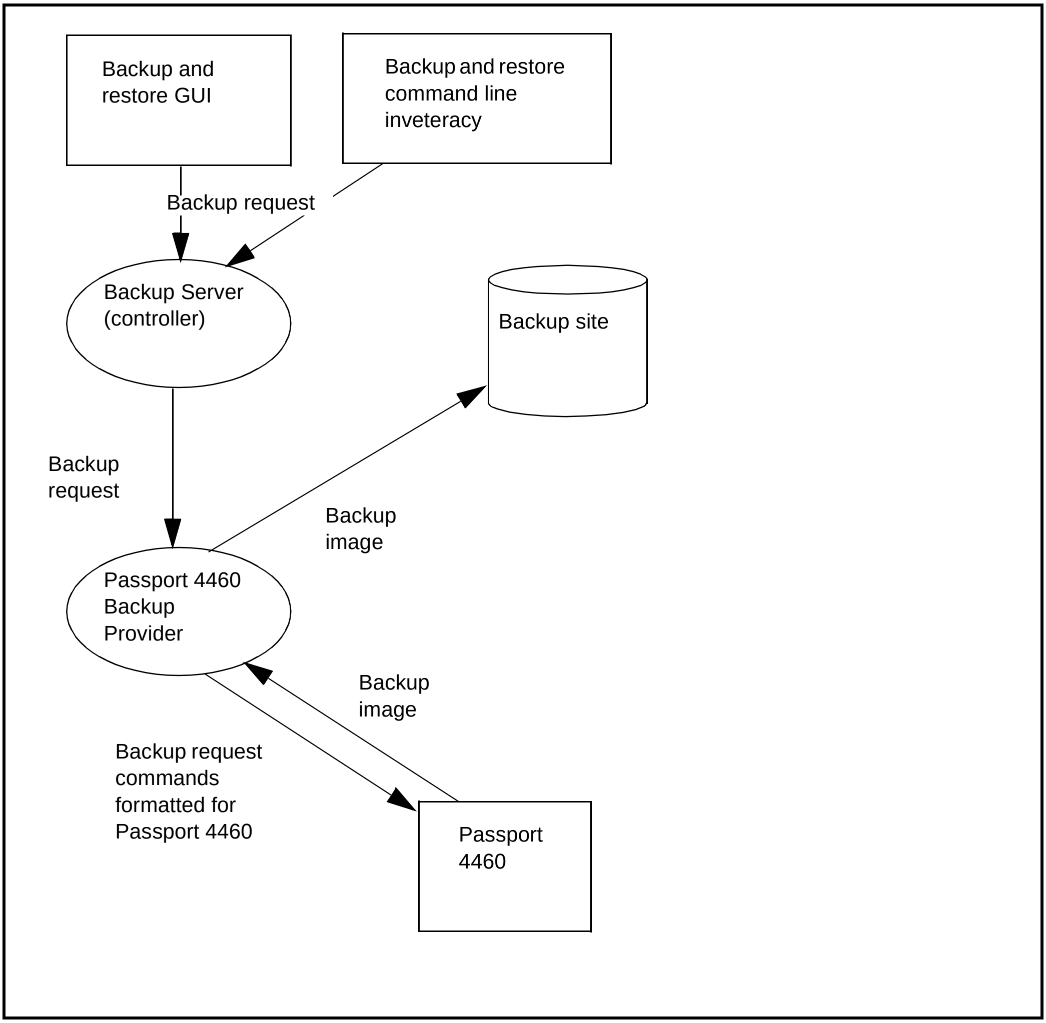
See “Dataflow diagram for the PBCKPP4460” (page 348) that shows the items related to the PBCKPP4460.

Here is an abridged version of the backup process:

1. A user sets up a request to perform a backup using the Passport/SNMP Backup and Restore tool or the command line interface. The tool or command line sends this request to the Backup Server (a controller).
2. The Backup Server sends a request to a backup provider. If the node being backed up is a Passport 4460, the controller sends the request to a PBCKPP4460.
3. The PBCKPP4460 formats the request for the Passport 4460 and forwards it to the Passport 4460.

4 The Passport 4460 provides access to the image, and the Backup Provider routes a copy of the image to backup site specified by the Backup Server

Figure 38
Dataflow diagram for the PBCKPP4460


Backup and restore GUI
Backup and restore command line inveteracy
Backup request
Backup Server (controller)
Backup site
Backup request
Backup image
Passport 4460 Backup Provider
Backup image
Backup request commands formatted for Passport 4460
Passport 4460

Managing the PBCKPP4460

See the following sections for information about managing the PBCKPP4460:

- "Configuration" (page 349)
- "Suggested name in Server Administration" (page 349)
- "Start-up command" (page 349)

Configuration

Configuring the PBCKPP4460 is part of a much larger task: Configuring Passport/SNMP Backup and Restore. For the instructions to do this, see 241-6001-807 *Preside MDM Network Backup and Restore*.

Suggested name in Server Administration

The recommended name to enter in the Server Manager Administration tool for the Passport 4460 Backup Provider is PP4460 Backup Provider.

Configuring the PBCKPP4460 with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the PBCKPP4460 has the following syntax:

```
/opt/MagellanNMS/bin/pbckpp4460
  [-m <interface_mapping_file>] [-p <port_no>]
```

where:

`interface_mapping_file` is the name of the interface mapping file. This optional file is required when your system has multiple network interfaces. The default interface mapping file is /opt/MagellanNMS/cfg/ifmap.cfg.

If the file ifmap.cfg exists in directory /opt/MagellanNMS/cfg and is populated with valid mapping information, it is not necessary to specify the `-m` option. This file is used by default.

`port_no` is the port number that the Provider uses.
The default port for the PBCKPP4460 is port 5040.

Interface mapping file (ifmap.cfg)

The Provider host machine can have multiple network interfaces. For example, the system can have one interface to the LAN and other interfaces to the WAN where the Passport 4400/4460 devices reside. In this configuration the Passport 4400/4460 devices cannot see the IP address of the interface to the LAN. In this situation you need to configure the Provider to use the correct interface address for the TFTP connection. You do this by creating an interface mapping file.

The Passport 4400 and Passport 4460 Providers can use the interface mapping file to determine the IP address used as the TFTP server address. The Passport 4400/4460 device connects to the TFTP server address for TFTP file transfers. Each line in the interface mapping file defines the mapping of the host address and the Passport 4400/4460 address or addresses. The host address is the TFTP server address.

The interface mapping file is named ifmap.cfg.

The format of ifmap.cfg is

```
<interface IP address> <device IP address(es)>
```

where:

`interface IP address` is the IP address of the interface to be used and has the format n.n.n.n.

`device IP address(es)` is the IP addresses of the devices and has the format n.n.n.n. You can match a group of devices by using the wildcard character (*).

You can include comments in the interface mapping file by inserting an octothorpe (#) at the beginning of the line. You can also include blank lines.

Example

The following example shows an interface mapping file for a host machine that has 3 network interfaces: one connects to the LAN and the other two are connected to separate networks of Passport 4400/4460 devices.

```
# Interface mapping file
# The last entry is the interface address to the LAN,
# which needs to map to any device address.
# This last entry can be omitted.

131.147.0.1 131.147.*
131.148.0.1 131.148.*
32.123.1.1  *
```

Interdependencies

The Preside Multiservice Data Manager software must be installed and configured, and the Backup Server (controller) must be configured and running.

Chapter 46

Passport 4460 Restore Provider (PRSTPP4460)

This section contains information about the Passport 4460 Restore Provider (PRSTPP4460). See the following sections for information about this server:

- “About the PRSTPP4460” (page 353)
- “Managing the PRSTPP4460” (page 354)
- “Interdependencies” (page 357)

About the PRSTPP4460

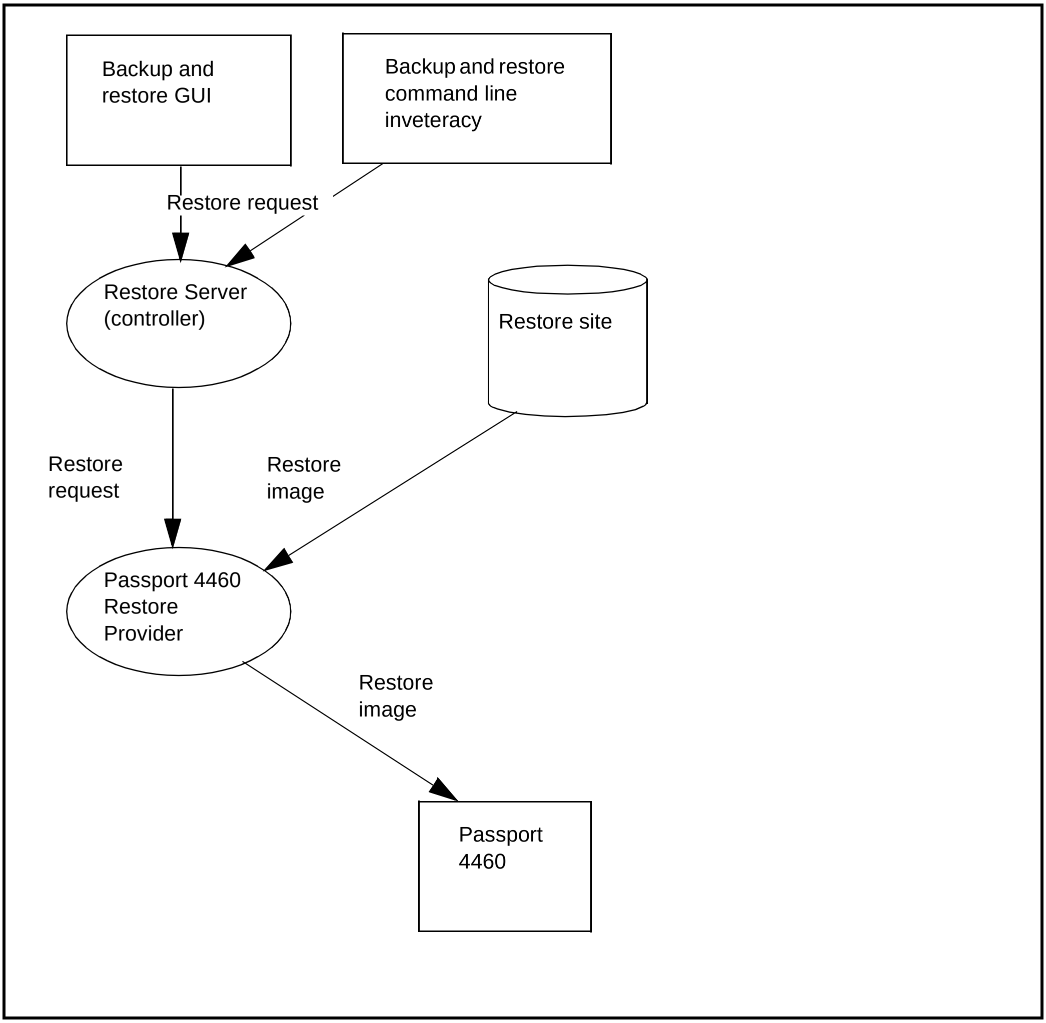
The PRSTPP4460 sends a backup image to a Passport 4460 in response to a request from the Restore Controller.

The “Dataflow diagram for the PRSTPP4460” (page 354) shows the items related to the PRSTPP4460.

Here is an abridged version of the restore process:

1. A user sets up a request to perform a restore using the Passport/SNMP Backup and Restore tool or the command line interface. The tool or command line sends this request to the Restore controller.
2. The Restore Controller sends a request to a Restore Provider. If the node being restored up is a Passport 4460, the controller sends the request to a PRSTPP4460.
3. The PRSTPP4460 obtains the backup image from the distribution site and forwards it to the Passport 4460.

Figure 39
Dataflow diagram for the PRSTPP4460



Managing the PRSTPP4460

See the following sections for information about managing the PRSTPP4460:

- “Configuration” (page 355)
- “Suggested name in Server Administration” (page 355)

- “Start-up command” (page 355)

Configuration

Configuring the Passport 4460 Restore Provider is part of a much larger task: Configuring Passport/SNMP Backup and Restore. For the instructions to do this, see 241-6001-807 *Preside MDM Network Backup and Restore*.

Suggested name in Server Administration

The recommended name to enter in the Server Manager Administration tool for the Passport 4460 Restore Provider is PP4460 Restore Provider.

Configuring the PRSTPP4460 with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the PRSTPP4460 has the following syntax:

```
/opt/MagellanNMS/bin/pprstp4460
  [-m <interface_mapping_file>] [-p <port_no>]
```

where:

`interface_mapping_file` is the name of the interface mapping file. This optional file is required when your system has multiple network interfaces. The default interface mapping file is /opt/MagellanNMS/cfg/ifmap.cfg.

If the file ifmap.cfg exists in directory /opt/MagellanNMS/cfg and is populated with valid mapping information, it is not necessary to specify the `-m` option. This file is used by default.

`port_no` is the port number that the Provider uses.
The default port for the PRSTPP4460 is port 5041.

Interface mapping file (ifmap.cfg)

The Provider host machine can have multiple network interfaces. For example, the system can have one interface to the LAN and other interfaces to the WAN where the Passport 4400/4460 devices reside. In this configuration the Passport 4400/4460 devices cannot see the IP address of the interface to the LAN. In this situation you need to configure the Provider to use the correct interface address for the TFTP connection. You do this by creating an interface mapping file.

The Passport 4400 and Passport 4460 Providers can use the interface mapping file to determine the IP address used as the TFTP server address. The Passport 4400/4460 device connects to the TFTP server address for TFTP file transfers. Each line in the interface mapping file defines the mapping of the host address and the Passport 4400/4460 address or addresses. The host address is the TFTP server address.

The interface mapping file is named ifmap.cfg.

The format of ifmap.cfg is

```
<interface IP address> <device IP address(es)>
```

where:

`interface IP address` is the IP address of the interface to be used and has the format n.n.n.n.

`device IP address(es)` is the IP addresses of the devices and has the format n.n.n.n. You can match a group of devices by using the wildcard character (*).

You can include comments in the interface mapping file by inserting an octothorpe (#) at the beginning of the line. You can also include blank lines.

Example

The following example shows an interface mapping file for a host machine that has 3 network interfaces: one connects to the LAN and the other two are connected to separate networks of Passport 4400/4460 devices.

```
# Interface mapping file
# The last entry is the interface address to the LAN,
# which needs to map to any device address.
# This last entry can be omitted.
```

```
131.147.0.1 131.147.*
131.148.0.1 131.148.*
32.123.1.1  *
```

Interdependencies

Preside Multiservice Data Manager software must be installed and configured, and the Restore Controller must be configured and running.

Chapter 47
Passport Backup Provider (PBCKPP)

This section contains information on the Passport Backup Provider (PBCKPP) . See the following topics for more information:

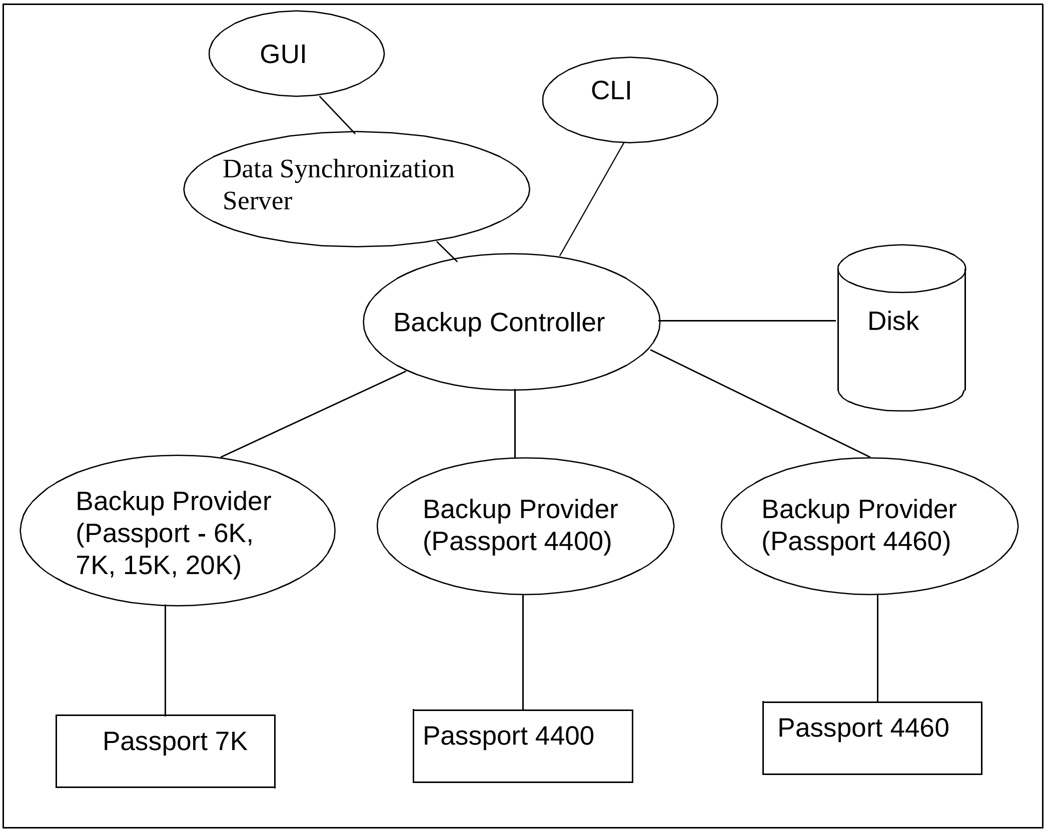
- “About the Passport Backup Provider” (page 359)
- “Managing the PBCKPP” (page 360)
- “Interdependencies” (page 361)

About the Passport Backup Provider

The Backup Controller communicates to the Passport Backup Provider, which sends a request to the Passport device to back up the most current view and/or journal files. The new files are stored on the backup site.

See 241-6001-807 *Preside MDM Network Backup and Restore*.

Figure 40
Passport Backup Provider dataflow diagram


GUI
CLI
Data Synchronization Server
Backup Controller
Disk
Backup Provider (Passport - 6K, 7K, 15K, 20K)
Backup Provider (Passport 4400)
Backup Provider (Passport 4460)
Passport 7K
Passport 4400
Passport 4460

Managing the PBCKPP

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 361)
- “Startup command” (page 361)

Suggested name in Server Administration

The recommended name is PP Backup Provider.

Configuring the PBCKPP with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the PBCKPP is as follows:

```
/opt/MagellanNMS/bin/pbckpp [-p <port_no>]
```

Use the following table to substitute command parameters:

Parameter	Definition
-p port_no	is the port number the Restore Provider uses. The port number is dynamically assigned but you can also specify it as port number 5020.

Interdependencies

The PBCKPP must be started before the Backup Controller when using the “-notification” option, for example:

```
/opt/MagellanNMS/bin/pbckpp
/opt/MagellanNMS/bin/nsctlbck -notification
```

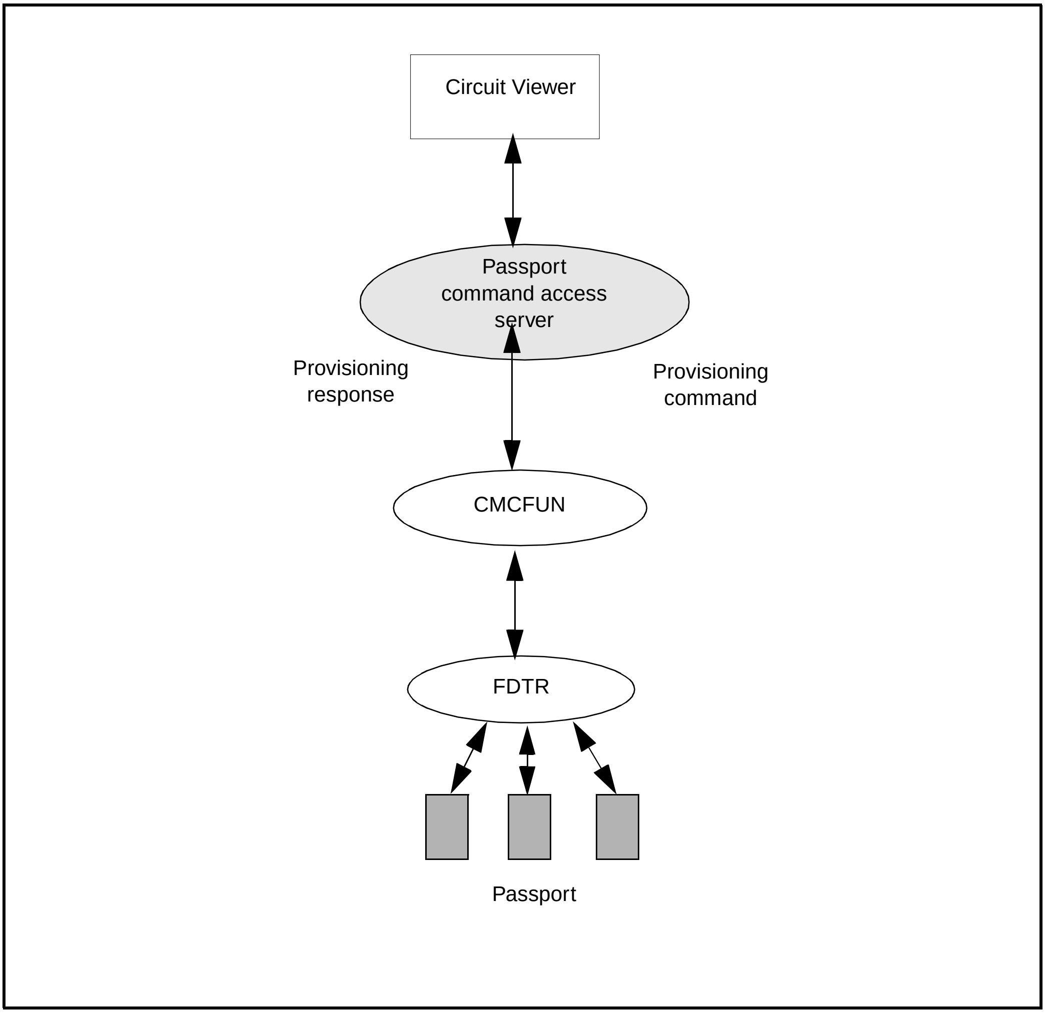
Chapter 48
Passport Command Access Server (PPAccessServer)

This section contains information on the Passport Command Access server (PPAccessServer). See the following topics for more information:

About the PPAccessServer

The PPAccessServer acts as an intermediary between Circuit Viewer and the Preside Multiservice Data Manager Command Console Functional Process (CMCFUN) server.

Figure 41
ATM data flow diagram



Managing the PPAccessServer

Use the Server Administration tool to enter or to edit the startup command and to start, stop, and configure this server to start automatically when the workstation is rebooted. Any changes you make to the startup command or to the configuration file become active whenever the server is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

For more information, see the following:

- “Suggested name in Server Administration” (page 365)
- “Startup command” (page 365)

Suggested name in Server Administration

The recommended name for the server is PP Command Access Svr.

Configuring PPAccessServer with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for PPAccessServer is as follows:

```
/opt/MagellanNMS/bin/eteserver -n PPAccessServer
-p <portno>
```

where:

-n this mandatory parameter specifies the server name

`-p <portno>` this mandatory parameter specifies the TCP port number for monitoring incoming requests. The default port value is 6601.

If you use a number other than the default, ensure that the alternate port number is not used by any other processes. Also ensure that the port number matches the port number specified in the Circuit Viewer configuration file /opt/nortel/CMT/cfg/CtViewer.cfg.

Interdependencies

The PPAccessServer depends on the HGDS, CM, CMCFUN, and ETESERVER servers. The Circuit Viewer tool depends on PPAccessServer.

Exit codes

Exit codes for the PPAccessServer are shown in the following table.

Table 56
Exit codes for the PPAccessServer

Exit code	Description
51	Out of memory.
55	Bad argument on command line.
59	Could not initialize IPC system.
60	Could not register service. (Is the server running?)

Chapter 49
Passport Communications Manager (FDTM)

This section contains information on the Passport Communications Manager (FDTM). See the following topics for more information:

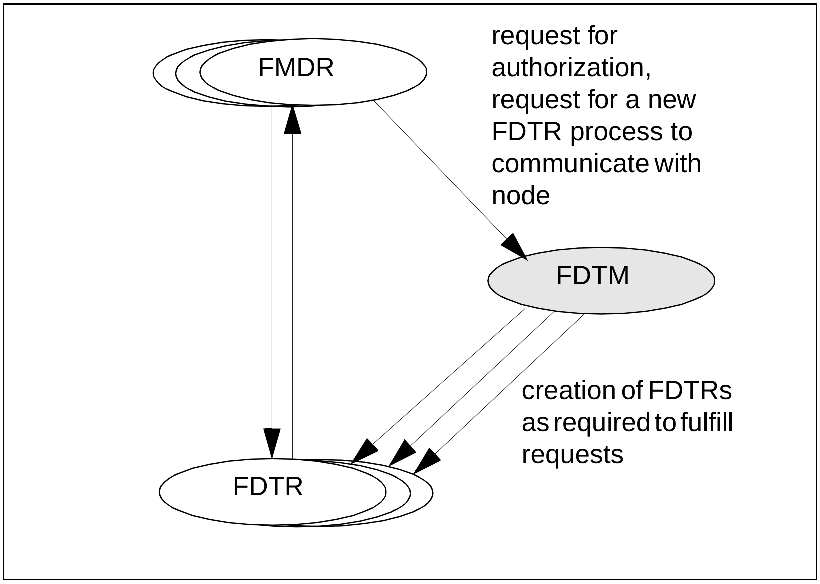
About the FDTM server

In networks that only contain DPN switches, this server is not used and should not be started.

The FDTM server creates and manages the data translation (FDTR) processes. The FDTR process allows the workstation to communicate with a Passport node. The FDTM server also provides user ID and password authentication service when users log in to a node through the Connection Manager. This service is not visible to the user.

FDTM performs command line option validation. If FDTM incurs an invalid option, it issues a log message and then terminates. Correct any invalid command line options and restart the FDTM server.

Figure 42
FDTM data flow diagram


FMDR
request for authorization, request for a new FDTR process to communicate with node
FDTM
creation of FDTRs as required to fulfill requests
FDTR

Managing the FDTM server

After software installation, the FDTM server is in the Server Administration tool's server list, but it is not started automatically by the server daemon (SVMDMN). Use the Server Administration tool to edit the FDTM server entry, enter the startup command, set the server options, and specify that the server should be started automatically at reboot time. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

After editing the server parameters, you can use the Server Administration tool to manually start the server, or you can edit the parameters for all of the node servers. See the section on configuring servers for Passport nodes in 241-6001-303 *Preside MDM Administrator Guide*. Then, reboot the workstation to start the servers automatically.

For more information, see the following:

- "Effective management of node connections" (page 369)

- “Suggested name in Server Administration” (page 370)
- “Startup command” (page 370)

Effective management of node connections

Passport nodes allow a limited number of connections for data and command access. Because of this, they are a valuable resource to operations personnel who manage the network. This is especially true for cases in which a network is global in scope and is managed from several different operations centers. In such cases, operators tend to start tools such as the Command Console or the Data Viewer then leave the tools running around the clock. However, this method can quickly exhaust all available connections and make it impossible to access a particular node at a time when that access is vital.

To alleviate this situation, the FDTM server has an argument in its startup command that allows an administrator to specify a connection timeout (-connTimeOut). This argument specifies the length of time that an idle connection to anode remains active before it is dropped. This scheme allows the FDTM servers to terminate connections when they are not in use, thereby ensuring that connections to nodes are always available when they are needed.

The connection timeout works as follows. If, for example, a user starts the Command Console tool, logs on to a group and sends a series of commands to Node X; and after awhile shifts focus to Node Y. If we assume the value of -connTimeOut to be 15 minutes, the user’s connection to Node X drops if no further commands are sent to the node within 15 minutes. If, after spending time on other nodes the user again sends commands to Node X, the connection is re-established. In the interim, other users have the opportunity to gain access to Node X. Terminating and re-establishing the connection is transparent to the user.

The default value for the -connTimeOut argument is 0 minutes. This value means that the connection never times out. If users are experiencing difficulty in obtaining connections, consider setting the -connTimeOut argument to a value other than the default (0).

If you have added a new node to a new group or to an existing group, it is not necessary for you to restart the FDTM server to update the server with the new group information. Instead, enter the following command to run the passport.kick program and update the server with the new group information:

```
/opt/MagellanNMS/bin/passport.kick
```

Suggested name in Server Administration

The recommended name for the FDTM server is PP Comms Manager.

Configuring FDTM with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

Enter the command in the Startup command field of the Server Administration dialogs in the Server Administration tool. See the section on editing a server in 241-6001-303 *Preside MDM Administrator Guide* for a description of how to enter the server startup command.

The command to start the FDTM server has the following syntax:

```
/opt/MagellanNMS/bin/fdtm \
[-connTimeOut <timeout_length>] [-h] \
[-numAuthNodes] <number_of_nodes>] \
[-numAuthRetries <number_of_retries>] \
[-offset <offset_time> [-positiveOffset]] \
[-keyBase <base_value>] \
[-numModels <number_of_models>] \
[-segBase <base_address>] \
[-segSize <segment_size>] \
[-connHeartBeatTime <interval_time>] \
[-msgTransTime <waiting_time>] \
[-ping <ping_time>]
[-maxPP <power of 2>]
[-congestBuffer <buffer_size>]
[-reConnTime <reconnect_timer>]
```

where:

- -connTimeOut, -h, -numAuthNodes, -numAuthRetries, -offset, and -maxPP are for normal usage
- -keyBase, -numModels, -segBase, and -segSize are for model loading; See “Loading Passport models in shared memory” (page 373) for more information.
- -connHeartBeatTime, -msgTransTime, -ping, and -congestBuffer are for network engineering

`[-connTimeOut <timeout_length>]` is the length of time that a connection used by the Command Console or the Data Viewer remains active between successive requests sent to a node. The timeout_length range is 0 to 90 minutes, and the default is 0. Specifying a value of 0 means that connections will not time out.

`[-h]` displays information about the startup command

`-numAuthNodes <number_of_nodes>` specifies the number of nodes that the FDTM server uses to authenticate the user ID and password supplied to it through the FMDR server. The software chooses the nodes at random from within the group. The number_of_nodes range is 1 to 10 nodes, and the default value is 3.

`-numAuthRetries <number_of_retries>]` specifies the maximum number of attempts that the FDTM server makes to authenticate the user ID and password supplied in the FDTM server’s startup command. At each attempt, the FDTM server selects a different set of nodes from within the group. The number_of_retries range is 0 to 90, and the default value is 3.

`[-offset <offset_time>]` is the number of minutes from the network time. A value between -720 and 720 represents the range of -12 hours to +12 hours. A time offset between 0 and 720 minutes (+12 hours) represents a time ahead of UTC (or east of the prime meridian). A time offset value between 0 and -720 minutes (-12 hours) represents a time behind UTC (or west of the prime meridian).

The following examples assume the network time is UTC:

- If you are in a location that uses UTC, set the time zone offset to 0 (the default value).
- If you are in a country east of the prime meridian, add 60 minutes to the minimum time zone offset (0) for each hour you are ahead of UTC. For example, if your time zone is two hours ahead of UTC, set the time zone offset to 120.
- If you are in a country west of the prime meridian (for example, the United States), subtract 60 from the minimum time zone offset (0) for each hour you are behind UTC. For example, if your time zone is four hours behind UTC, set the time zone offset to -240.

For backwards compatibility, <offset_time> has the range of -720 to 1440 minutes.

If you are in a location with a time zone greater than 12 hours ahead of UTC, see the option [-positiveOffset]. This option indicates that the value specified as <offset_time> for the -offset option is to be considered a valid time zone offset ahead of UTC even though it is greater than 720. Use [-positiveOffset] for Australia (GMT +13) and New Zealand (GMT +14). For example, if you are in a country east of the prime meridian, 13 hours ahead of UTC, set the time zone offset to 780 (13x60), and include the [-positiveOffset] on the command line to indicate it is a positive offset.

`[-keyBase <base_value>]` is the base value for keys, which are assigned to memory segments as the segments are created. There is potential for key conflicts between different applications creating shared memory segments. Therefore, you need to manage the key ranges of the applications that are on the same workstation.

`[-numModels <number_of_models>]` is the maximum number of models that you can load simultaneously. The default value for this parameter is 9.

`[-segBase <base_address>]` is the base address in FDTR process memory to which the segment is mapped. The default value for this parameter is 0xD0000000. This parameter is rarely used; you only need to use it if you cannot use the default value.

`[-segSize <segment_size>]` is the size, in megabytes, of each memory segment. The default value for this parameter is 20.

`[-connHeartBeatTime <interval_time>]` is the number of seconds between each heartbeat. The default value for this parameter is 30.

`[-msgTransTime <waiting_time>]` is the number of seconds the workstation waits for a response from a node. The default value for this parameter is 20.

`[-ping <ping_time>]` is the number of seconds used to determine if a node is reachable. The ping time is used when connecting to a group. Its range is 1 to 60 seconds, and the default is 2 seconds.

`[-maxPP <power of 2>]` is the maximum number of nodes allowed by a group in the HGDS.cfg file. Specify this number as a power of 2, for example, 4, 8, 16, 32, and so on. If you do not set this option in the fdtm startup command, or if you specify an invalid number, then a default value of 2048 is used.

`[-congestBuffer <buffer size>]` sets the number of messages that will be stored in the outgoing message queue to a client. When this queue overflows, FDTR tears down the connection to the client. The default is 2000.

`[-reConnTime <reconnect_timer>]` specifies the time interval for FDTRs to try to reconnect with the device when the connection is lost. A value between 2 and 120 represents the range of 2 seconds to 120 seconds. The default value is 120 seconds.

Loading Passport models in shared memory

A Passport model is a representation of the node component hierarchy. Applications use Passport models for message translation. The Passport model library resides in shared memory and lets you have one model used by multiple applications. Passport models are updated dynamically in shared memory as upgraded Passport software appears in the network. This ability lets you manage updated nodes without stopping FDTM and all FDTRs. Therefore, there is no service interruption on the other nodes in the network.

More than one model can be loaded in shared memory at the same time. See “Startup command” (page 370) for the command line parameters that are associated with Passport models.

Using shared memory for Passport models reduces memory resources. In addition, while a model remains in shared memory, you do not have to wait for it to rebuild when any application needs to use it again. The first time that a model is parsed and loaded, an image of the model is created in a file that has a naming convention of sursdd<xxx>.act.img. This image file is placed in /opt/MagellanNMS/cfg/PassportSchema. Subsequent loads of this model are very fast because the image file is used when it exists.

When an application that needs a new model is started, the application exits if the number of loaded models equals the number of models allowed in shared memory. You need to terminate all tools that are using any one model so that the model has no users. Then when you invoke the required application again, the old model with no users is deleted and the required model is loaded in its place.

Interdependencies

If you are using a LAN configuration, you need to run the FDTM server on the LAN-selected workstation. The FDTM server must also be running on any workstation on which you want to run the FMDR server.

Exit codes

FDTM also outputs useful diagnostic messages to the MDM Log Display. Exit codes for the FDTM server are shown in the following table.

Table 57
Exit codes for the FDTM server

Exit code	Description
0	Successful exit. Do not restart the server.
1	Failure. Restart the server.
50	Do not restart (for backward compatibility).
(Sheet 1 of 2)	

Table 57 (Continued)
Exit codes for the FDTM server

Exit code	Description
51	Memory resource error (For example, not enough memory). No restart.
52	Disk resource error (For example, not enough space, no file). No restart.
53	Communication resource error (For example, cannot connect or register). No restart.
54	Timeout/deadlock/congestion (For example, too much congestion). No restart.
55	Bad command line arguments. No restart.
56	Bad configuration file or environment. No restart.
57	Fork exec failure. No restart.
58	Manual server shutdown (admin?). No restart.
59	ipc_init unsuccessful. No restart.
60	ipc service cannot be registered. No restart.
61	Exit signal received. No restart.
62	Terminated due to licencing problem (see MDM Logs). Look at the System Log Display tool for logs about licencing problems, then take the corrective action for the log, as recommended in the Error Messages section for this server.
100	Other unclassified errors. Restart the server.
(Sheet 2 of 2)	

Error messages

Error messages for the FDTM server are shown in the following table.

Table 58
Error messages for the FDTM server

Error message	Meaning and action
FDTM - Cannot allocate a licensing context	The FDTM server licensing context initialization failed. Verify that files LIClicenses.cfg and LICcustName.cfg exist in directory /etc/opt/Magellan and that they can be read.
FDTM - License refused: <reason>	A run-time license cannot be allocated to the FDTM server for the product, release, and customer indicated. Verify that file /etc/opt/Magellan/LIClicenses.cfg contains a valid license key for running FDTM.
FDTM - License warning: <reason>	The license required to run FDTM is about to expire. Contact Nortel Networks immediately.
FDTM - License not confirmed: <reason>	There is a problem with the license required to run FDTM, it has most likely expired. Contact Nortel Networks immediately.
FDTM Invalid offset parameter in command line (0-1440) FDTM Invalid ping parameter in command line (1-60) FDTM invalid directory on watch option on command line	Fatal, invalid command line argument. Revise configuration with the Server Administration tool.
FDTM failed to set descriptor limit, using default FDTM failed to get descriptor limit, using default	Unlikely, non fatal, could not augment file descriptor limit.
FDTM service already exists	Fatal, could not register the service name. Server may already be running. Revise server configuration with the Server Administration tool.
FDTM terminated unable to access anymore memory	Fatal, memory allocation problems. Restart the server.
(Sheet 1 of 5)	

Table 58 (Continued)
Error messages for the FDTM server

Error message	Meaning and action
FDTR failed to set descriptor limit, using default	Unlikely, non fatal, could not augment file descriptor limit.
FDTR service already exists	Contact your System Administrator.
FDTR cannot access anymore memory	Memory exhausted, fatal. If this is occurring frequently, you probably don't have enough RAM to run Preside Multiservice Data Manager.
FDTR destroyed connection to client, congestion problem	An FDTR client wasn't handling FDTR output fast enough and was dropped by the server. This can occur if a user "locks" or otherwise freezes a display that is being dynamically updated with data from a node connection. For example, cmccmd output.
Get failed because no value for identifier Get failed because attribute is not a fedUnsigned_c Get failed because component name is empty	If the problem persists, contact your System Administrator support.
Failed to allocate shared memory segment - errno: 22	The default workstation shared memory segment size is smaller than the size of the segment being created by Preside Multiservice Data Manager. Check the default segment size for the workstation in /etc/system, for example, set shmsys:shminfo_shmmax=10485760. Either restart FDTM and use the -segSize option to decrease the segment size that you are creating, or use a utility to increase the default segment size. See the section on setting the amount of shared memory in the kernel in 241-6001-303 *Preside MDM Administrator Guide* for information on the utility.
(Sheet 2 of 5)	

Table 58 (Continued)
Error messages for the FDTM server

Error message	Meaning and action
FDTM: Fdtm could not load the Passport model. Check segment size or number of models in use	If accompanied by the previous message, follow the previous procedure. Otherwise, it means that all available segments are in use. Either free an existing model and launch the application again, or restart FDTM and use the -numModels option to increase the number of models that can be in use at the same time.
<Passport SDD Model error>	Problem with the Passport SDD file.
<Sdd filename> could not be parsed.	Either there is an error in the sursdd<version>.act.file or the parser cannot parse a new feature.
ERROR: syntax error at or near line <line number> on token *123* in <sdd filename>	Gives the location of an error in the sdd file.
FDTM: Model loader exited abnormally. Model not loaded. Check sdd file <sdd filename>	Parsing error.
APPLICATION_ERROR 1111 FDTR cannot connect to <host>	The specified node is out of service or not reachable for some reason. Verify the state of the node and take corrective actions if required.
APPLICATION_ERROR 1112 FDTR lost connection to	FDTR has lost connection to client - you should already have seen a problem somewhere else in the system.
APPLICATION_ERROR 1114 FDTR cannot connect to the Group Directory Service	Check that the Host Group Directory Server has been configured to run by means of the Server Administration tool.
APPLICATION_ERROR 1117 FDTR only supports one client	Contact your System Administrator.
APPLICATION_ERROR 1118 FDTR received an invalid logon from client	Contact your System Administrator.
(Sheet 3 of 5)	

Table 58 (Continued)
Error messages for the FDTM server

Error message	Meaning and action
APPLICATION_ERROR 1119 FDTR received an invalid request from client	Somewhat unusual, non fatal. If seen frequently, report to your System Administrator
APPLICATION_ERROR 1120 FDTR received an invalid response from device	Can occur when the latest version of the *sursdd* file hasn't been installed in /opt/MagellanNMS/cfg/PassportSchema. Ensure that the file is at the highest release level of all Passports running in the network.
APPLICATION_ERROR 1121 FDTR communication congestion to device <host>	The FDTR process has had to wait an unreasonable length of time for the node to respond to a command; this usually indicates that the node is heavily loaded for some reason. If you believe this to be unusual, investigate the node to discover the source of the heavy load.
APPLICATION_ERROR 1122 FMIP connection locked or down, command terminated	Can occur if FMIP connections are locked or otherwise disabled on the node. Verify that no one inadvertently disabled FMIP connections.
APPLICATION_ERROR 1123 FDTR client is already registered	Contact your System Administrator.
APPLICATION_ERROR 1125 FDTR is unable to read the schema	Verify that files in the /opt/MagellanNMS/cfg/PassportSchema directory are readable by anyone.
APPLICATION_ERROR 1128 FDTR Group does not contain device	Can occur when requests are made for nodes that aren't in a particular group. Ensure that node-related servers are restarted whenever groups are modified, or nodes are removed from groups.
APPLICATION_ERROR 1132 FDTR - Invalid event type in alarm	Contact your System Administrator.
(Sheet 4 of 5)	

Table 58 (Continued)
Error messages for the FDTM server

Error message	Meaning and action
APPLICATION_ERROR 1134 FDTR - Invalid scn event from device	Contact your System Administrator.
APPLICATION_ERROR 1136 FDTR - Invalid group requested	Can occur when changes have been made to the group configuration and node servers haven't been restarted. Ensure that node-related servers are restarted whenever groups are modified.
APPLICATION_ERROR 1137 FMIP connection not responding	If the problem persists, contact your System Administrator.
APPLICATION_ERROR 1138 FMIP authentication failed on specified device	Ensure that the user IDS/passwords are consistent across all nodes in your defined groups.
APPLICATION_ERROR 1139 FDTR received an invalid component name from client	Contact your System Administrator.
APPLICATION_ERROR 1143 FDTR - can't authenticate on nodes: <X><state> <Y> <state> <Z> <state>... Attempted authentication on <A> of <B> available modules.	The FDTR process is unable to authenticate on one or more nodes. where: <X>, <Y>, and <Z> are the host names of the nodes on which authentication failed <state> is the state of the connection when authentication failed. This state is one of: unpingable, no FMIP, bad UID/pw (user ID/password), can't connect, or lost connection. <A> represents the number of nodes on which authentication was attempted <B> represents the number of nodes in the group.
(Sheet 5 of 5)	

Chapter 50
Passport Configuration Model Server (PCMS)

This section contains information about the Passport Configuration Model Server (PCMS). See the following topics for more information:

About the PCMS server

The PCMS server centralizes the control of the Passport provisioning models. This server loads the Passport models into shared memory so that other Passport provisioning server (FPS) can access the same models. The PCMS performs the following services:

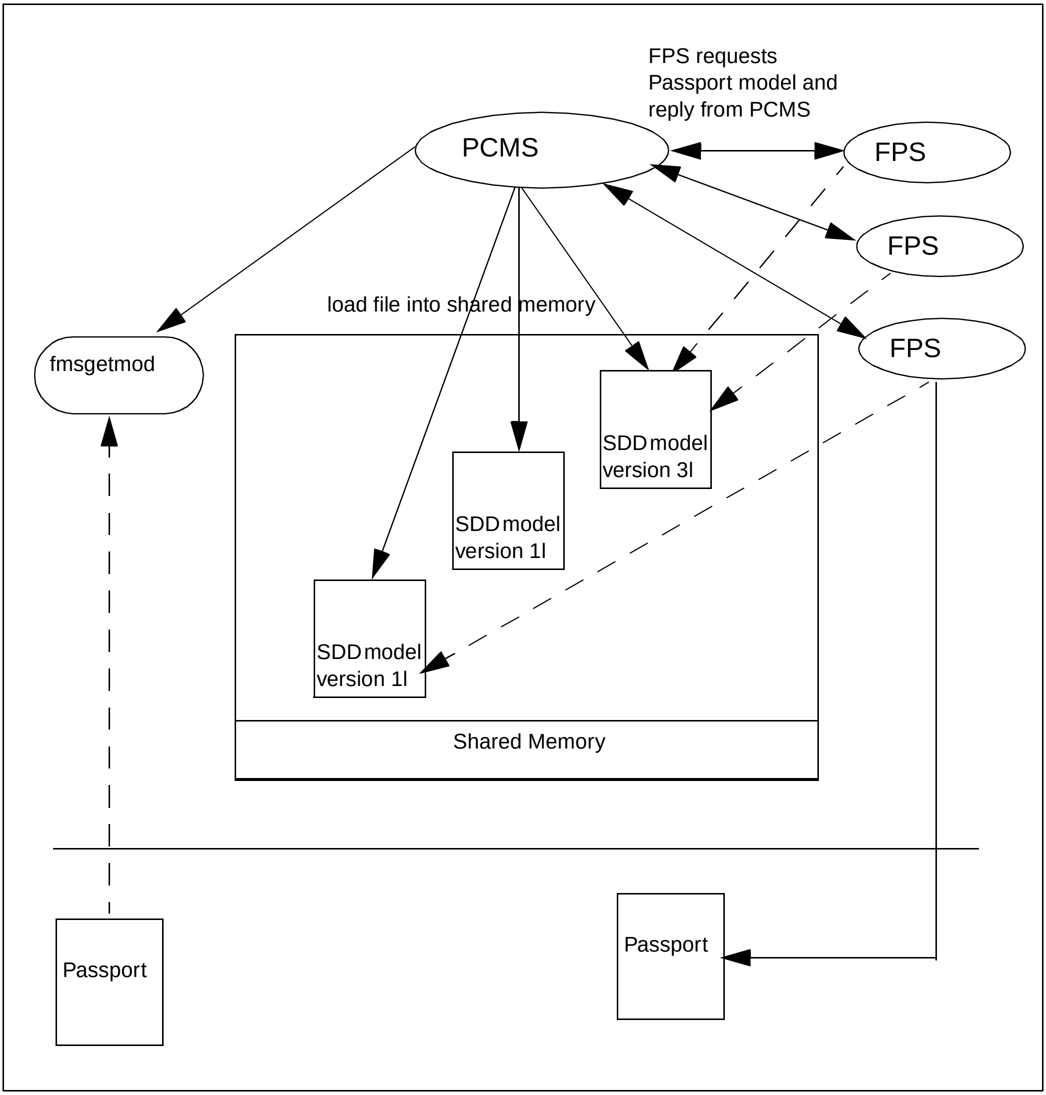
- allocates the shared memory segments in the desired size
- loads multiple Passport provisioning models into shared memory
- centralizes control of the shared models. Retrieves model files (SDD and CDL) from Passport nodes when the models for the requested versions are not present on the Preside Multiservice Data Manager (MDM) workstation.
- generates all model files required by MDM applications from SDD and CDL

- lets FPS retrieve the information from the shared memory and shared model. This enables FPS to use this information to attach to and access the shared model.
- supports suspense mode. The PCMS postpones the PCMS's termination and prevents new FPSs accessing PCMS. The PCMS only terminates when all the FPSs release the shared models.
- displays the logs using the Log Display tool. See 241-6001-303 *Preside MDM Administrator Guide*.

For each Passport model request, one of the following occurs:

- If the model is loaded in shared memory, the shared memory ID is returned in the reply. FPS attaches to the shared memory and the Passport model information can be accessed.
- If the Passport model is not available, PCMS searches for the model file from the disk. If it is found, it is loaded into shared memory.
- If the Passport model is not available, and cannot be found on the disk, PCMS invokes fmsgetmod to upload the SDD and CDL model from the node. This generates model files that can be used by different applications and loads the model required by FPS in shared memory.
- If the file cannot be loaded due to heap exhausted, FPS loads the model into its local memory.

Figure 43
PCMS data flow diagram



Managing PCMS

Use the Server Administration tool to enter or edit the startup command and to start and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- "Suggested name in Server Administration" (page 384)
- "Startup command" (page 384)

Suggested name in Server Administration

The recommended name for the PCMS name is PP Config Model Server.

Configuring PCMS with the Server Administration tool requires you to enter the server name in the **Descriptive name** field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the PCMS is as follows:

```
/opt/MagellanNMS/bin/pcms [-numOfModels <n>]
[-modelSize <s>] [-preloadModels <v1><v2>...]
```

Variable definitions

Variable	Definition
n	is the number of models to be loaded. The default is 2.
s	is the model size in Mbyte. The default is 20 Mbyte.
v1, v2	is the node version number

Stopping PCMS and retrieving shared memory

The pcmsmon utility terminates the PCMS or retrieves the shared memory available and the shared model's information. The command is as follows:

```
/opt/MagellanNMS/bin/pcmsmon [-stop] [-query]
```

Stop terminates the PCMS:

```
/opt/MagellanNMS/bin/pcmsmon -stop
```

Query retrieves the shared memory status and model information:

```
/opt/MagellanNMS/bin/pcmsmon [-query]
```

An example of the query command output is as follows:

```
Passport Configuration Model Server state: RUNNING
Shared heap available: 6 Mb

Number of models loaded: 2
Version CD0106A - Size: 17 Mb Current access: 2 Total
access: 5
Version CD0222A - Size: 17 Mb Current access: 1 Total
access: 20
```

PCMS postpones the termination until all attached models are released. The Server Administration tool automatically restarts PCMS.

Interdependencies

You can configure the PCMS server and shared memory size using Quickstart. PCMS interacts with the Passport provisioning stack (FPS).

Exit codes

Exit codes for the PCMS server are shown in the following table.

Table 59
Exit codes for the PCMS

Exit code	Description
0	Successful exit
51	Memory resource error. Failed to allocate shared memory.
(Sheet 1 of 2)	

Table 59 (Continued)
Exit codes for the PCMS

Exit code	Description
52	Unable to remove an existing shared memory resource.
55	Invalid argument
59	Unable to initialize interprocess communication.
60	Unable to register its service. The PCMS may already be running.
61	Exit signal received
62	Terminated due to invalid Preside Multiservice Data Manager license key.
100	Other unclassified errors. Restart the server.
(Sheet 2 of 2)	

Error messages

The following PCMS error messages are logged to the Preside Multiservice Data Manager System Log display.

Table 60
Error messages for PCMS

Error message	Meaning
Cannot upload/generate model version <version>. <Reason>.	The PCMS cannot retrieve the Passport model file or the models used by Preside Multiservice Data Manager applications cannot be generated. For this PCMS error, the provisioning application will load the model in its local memory space.
Unable to start. Failed to register with the MDM name server. Another Passport Configuration Model Server is probably running.	The PCMS cannot register with Preside Multiservice Data Manager name server. The likely cause is that another PCMS server is running.
(Sheet 1 of 2)	

Table 60
Error messages for PCMS

Error message	Meaning
Cannot activate model version <version>. Not enough memory left in the reserved shared memory segment. Only <size> Mb left out of <size> Mb reserved.	The PCMS cannot activate the model version into its shared memory segment. There is not enough room for the additional model in the reserved shared memory segment. Stop and restart the PCMS server to release the memory taken by old models using the pcmsmon utility. You can also increase the memory size reserved for PCMS by running PCMS with the "-numOfModels" and "-modelSize" options on the server command line.
The Passport Configuration Model Server has been suspended. It no longer accepts any requests.	The PCMS server received a termination request from the administrator. The PCMS server waits for all existing Passport provisioning server (FPS) connections to terminate before exiting. In the mean time, the PCMS no longer accepts any requests. For this PCMS error, the provisioning application loads the model in its local memory space.
Unable to start. Failed to allocate/initialize <size> Mb of shared memory.	The requested size is probably larger than the shared memory segment sized defined in the file /etc/system.
(Sheet 2 of 2)	

When the shared model is not available, the following warning messages will be displayed by Nodal Provisioning or Embedded Nodal Provisioning when opening a view. The following Passport provisioning stack (FPS) error messages are also generated:

Table 61
Error messages for FPS

Error message	Meaning
Cannot connect to the Passport Configuration Model Server.	The model version <version> is loaded into the local memory.
The Passport Configuration Model Server is not running. The FPS application will load the model into its local memory	The PCMS server is not running. The FPS application will load the model into its memory.
Cannot activate model version <version>. Not enough memory left in the reserved shared memory segment. Only <size> Mb left out of <size> Mb reserved. The model version <version> is loaded into local memory.	There is not enough room for the additional model in the reserved shared memory segment. The FPS application will load the model into its memory. Stop and restart the PCMS server to release the memory taken by old models by using the pcmsmon utility. You can also increase the memory size reserved for PCMS by running PCMS with the "-numOfModels" and "-modelSize" options on the server command line.

Chapter 51 Passport Management Data Router (FMDR)

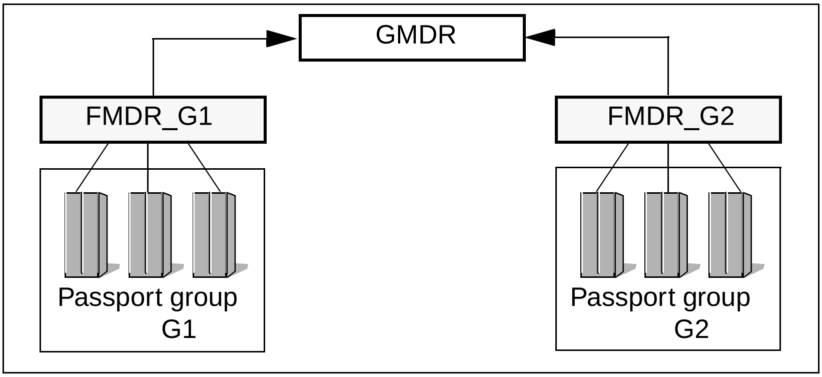
This section contains information on the Passport Management Data Router (FMDR). See the following topics for more information:

About the FMDR server

In networks that do not contain Passport devices, the FMDR server is not used and should not be started.

The FMDR server is responsible for routing alarm and state change notification event reports from a group of Passport nodes to GMDR server(s). A separate FMDR server is required for each surveillance group in the network. It is good practice to have at least two FMDR servers on different workstations or network access paths running for each surveillance group in the network. In this way, data retrieval is maintained if connectivity to the network is lost by one of the servers. FMDR uses the group specified in the FMDR server startup command. See the figure “FMDR server configuration” (page 390).

Figure 44
FMDR server configuration



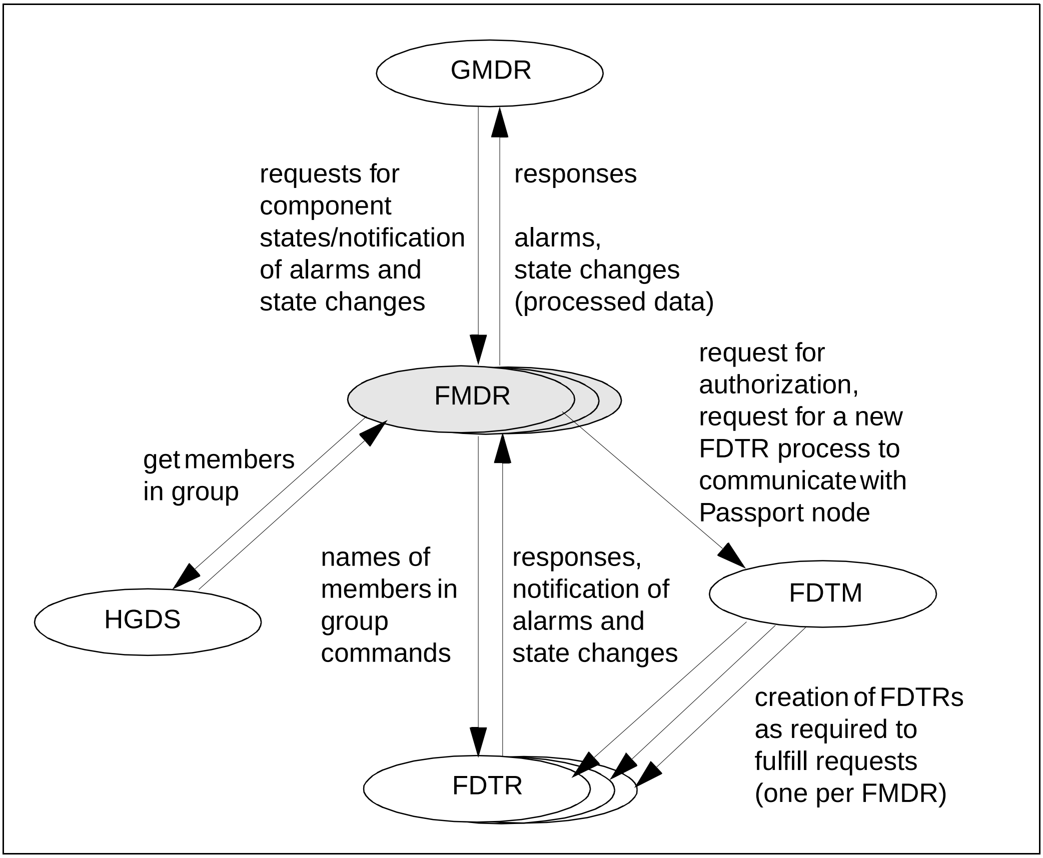
FMDR functionality

FMDR performs the following functions:

- connects to a node group for surveillance
- collects surveillance information from the nodes
- stores the surveillance information for the nodes
- supplies the surveillance information to its GMDR clients
- detects new or obsolete nodes components and notifies its GMDR clients
- supports automatic deletion of dynamic node components and notifies its GMDR clients
- applies customer network management identifier (CNMID) filtering of surveillance information
- supports get requests, manual alarm clear requests, database reset requests, resynch requests, component delete requests, and query property requests from its GMDR clients

The figure “FMDR data flow diagram” (page 391) illustrates some of these functions.

Figure 45
FMDR data flow diagram


GMDR
requests for component states/notification of alarms and state changes
responses
alarms, state changes (processed data)
FMDR
request for authorization, request for a new FDTR process to communicate with Passport node
get members in group
HGDS
names of members in group commands
responses, notification of alarms and state changes
FDTM
creation of FDTRs as required to fulfill requests (one per FMDR)
FDTR

Node access

A surveillance group is defined solely to provide access to the nodes connected to the FMDR server. Surveillance groups are defined in the Host Group Directory Server information file. For more information on configuring servers for Passport nodes, see 241-6001-303 *Preside MDM Administrator Guide*.

FMDR servers automatically log on to a node group during system initialization using the user ID and password specified in the server startup command. The FMDR server manages each node in the group, using the group member information supplied by the HGDS server.

FMDR does not talk directly to the node; it asks the FDTM server for an FDTr process for access (see “Passport Communications Manager (FDTM)” (page 367) for more information). The FDTr process also issues a lost connectivity proxy alarm when the connection to a node is down for more than one minute. FMDR and FDTr automatically attempt to re-establish any failed connections.

⚠ **CAUTION**

Inability to connect to nodes

Do not define groups for surveillance access that contain more than 60 nodes. Doing so may cause difficulty in connecting to all of the nodes in the group to obtain surveillance information.

Detection of Passport family

In order to use the correct Passport SDD file for data translation, the FDTR process associated with an FMDR server determines the Passport family of each node in the group. FMDR also determines the family of each node in the group in order to use the correct state walk configuration files and alarm exceptions files. Family is determined by the version name of the Passport software loaded on the node.

This allows FMDR to manage nodes of different families within the same group. FMDR/FDTr also adapts automatically to new Passport software versions as they appear in the network.

Surveillance events and state walk

Nodes provide the following surveillance information to FMDR:

- an alarm notification flow
- an OSI state change notification flow
- explicit OSI state information on components

FMDR initializes or synchronizes its surveillance database through an explicit state walk of the nodes in its group. By default, FMDR will state walk up to 10 nodes in its group in parallel. The maximum number of concurrent state walks can be overridden through startup options (see “Startup command” (page 398) for more information). The state walk is driven by the state walk configuration files. See “FMDR state walk configuration” (page 401) for more information.

The state walk allows FMDR to learn about component existence, to extract current component OSI state, to extract other information attributes like customer ID, and to extract the active alarms.

> ***Note:*** Active alarms may be obtained from the node only if the Active Alarm List feature (Passport release PCR 5.1) is installed and provisioned on the node.

Detection of obsolete components, that is, components in the FMDR database that no longer exist in the network, is done after the state walk completes. Obsolete components are deleted from the FMDR database and notification of the deletion is sent to all clients. Depending on the startup options specified for the server SURNUP, the obsolete component deletions may be reflected in the active network model (see “Startup options” (page 320).

Detection of dynamic components that should be deleted occurs after the state walk is completed (see “Dynamic components” (page 395). These components are deleted from the FMDR database and notification of the deletion is sent to all clients.

A state walk is triggered automatically by the following events:

- initial connection to a node by the FMDR
- reconnection to a node by the FMDR
- when a Confirm Provisioning CLR alarm (sent after activation of new provisioning is confirmed) is received from a node

> ***Note:*** By default, post-activation state walks will not be triggered less than 15 minutes after the last walk on the target node in order to avoid over-stressing it. The post-activation state walk delay can be overridden through the startup options (see "Startup command" (page 398) for more information).

Although the automatic statewalks are sufficient to properly initialize the surveillance database, a state walk is also triggered through the following administrative actions available from the GMDR administration tool:

- synchronization action on the FMDR server or a specific module
- reset subserver database on the FMDR server
- delete component action on a Passport module

For more information, see the GMDR Administration tool, as described in 241-6001-303 *Preside MDM Administrator Guide*

> ***Note:*** The synchronization action triggers a state walk on all the nodes in a group, unless there is already a state walk in progress for the node.

FMDR may be configured such that a state walk on a particular node or the whole group is performed at regular intervals. See "Additional configuration for recurrent state walk" (page 403) for more information.

> ***Note:*** By default, there are no recurrent state walks.

FMDR may be configured to use a polling mechanism to monitor circuit level components. With circuit monitoring, a state walk is performed only on the specified circuit components and only at specified polling times. For more information, see "Additional configuration for circuit monitoring" (page 405).

> ***Note:*** By default, there is no circuit monitoring.

Between state walks FMDR relies on the OSI SCN notification flow and alarm notification flow from the node to accurately maintain state and alarm information for the nodes in its group. These events are forwarded on to the clients of FMDR. In addition, FMDR will automatically generate proxy alarms to represent out of service (OOS) or troubled components without sufficient device alarms.

FMDR may be configured to handle specific alarms in a special way, either by ignoring an alarm, or by changing the severity of a SET or MSG alarm. See “Additional configuration for alarm exception handling” (page 404) for more information. In this way, an alarm that is considered to be more important than is indicated through it severity may be mapped to a more critical state or vice versa.

Dynamic components

In FMDR, node components are classified as ordinary or dynamic. A component that is classified as ordinary is never deleted due to its current raw state. A component that is classified as dynamic is deleted when it does not have a troubled raw state and has no links or subcomponents associated with it.

Components are classified in the following way:

- all links and modules in FMDR are always ordinary
- a subcomponent that is discovered through a state walk is ordinary by default unless classified explicitly as dynamic in the FMDR configuration file (see “FMDR state walk configuration” (page 401))
- a subcomponent that is discovered through circuit polling is ordinary by default unless classified explicitly as dynamic in the FMDR configuration file (see “FMDR state walk configuration” (page 401))
- a subcomponent that is discovered only through events is dynamic

Link autopopulation and deletion

FMDR can create and maintain the state for link components based on information from alarms and SCNs, and information gathered during a state walk. From information on the following endpoints, TRK, DPNGATE and MPANL, FMDR creates the following link types:

- Passport DPRS/PORS Trunks (PTK)
- DPNGate Trunks (TK)
- Network Links (NL)

- DPRS Dial Backup Network Links (DBNL)
- DPRS Bandwidth on Demand Links (BWOD)
- DPRS Dial Network Links (DNL)
- Multiservice Passport Access Network Links (MPANL)

FMDR can also be configured through server startup options to perform ATM link autopopulation. The link is created based on the ATMIF's "remoteAtmInterfaceLabel" attribute value on the node with a link type specified on the server startup command line. See "Startup command" (page 398) for more information.

> ***Note:*** The ATMIF component's "remoteAtmInterfaceLabel" attribute is expected to have been given the full component name of the peer ATMIF component in Preside Multiservice Data Manager format, for example, EM/<RemoteName> ATMIF/<instance>.

If FMDR detects and creates a new link with an endpoint that is already an endpoint of an existing link, then the existing link is considered obsolete. As with obsolete component detected after a state walk, obsolete links are deleted from the FMDR database and notification of the deletion is sent to all clients.

> ***Note:*** Link autopopulation can be disabled through command line options (see "Startup command" (page 398) for more information).

Data filtering by Customer Network Management Identifier (CNMID)

The FMDR server filters out alarm events based on the CNMID obtained from the node during authentication of the user ID and password. Therefore, the GMDR server only receives alarms that are valid against the CNMID that is returned from the node for the specified user ID and password.

Administrative actions

The FMDR server supports the following administrative actions.

- reset database
- delete component

- trigger synchronization

They are available to the administrator through GMDR Administration tool, as described in 241-6001-303 *Preside MDM Administrator Guide*.

Managing FMDR servers

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for FMDR. Any changes you make to the startup command or options take effect when FMDR is restarted. For instructions to use the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

If you have removed a node from a surveillance group, you must stop and restart the Preside Multiservice Data Manager servers for Passport nodes, including the FMDR server for the modified surveillance group.

If you have added a new node to a new group or to an existing group, you do not need to restart the FMDR server to update the server with the new group information. Instead, you can invoke the following command to run the passport.kick utility and update the server with the new group information:

```
/opt/MagellanNMS/bin/passport.kick
```

For more information on configuring servers, see 241-6001-303 *Preside MDM Administrator Guide*.

For the GMDR server to use the services of the FMDR server, it must send the user ID and password for the FMDR group to the FMDR server. For more information about supplying user IDs and passwords, see the section describing the GMDR Administration tool in 241-6001-303 *Preside MDM Administrator Guide*.

FMDR configuration

This section covers:

- “Suggested name in Server Administration” (page 398)
- “Startup command” (page 398)
- “FMDR state walk configuration” (page 401)

For additional configuration information, see the following sections:

- “Additional configuration for recurrent state walk” (page 403)
- “Additional configuration for alarm exception handling” (page 404)
- “Additional configuration for circuit monitoring” (page 405)

Suggested name in Server Administration

The recommended service name for FMDR is FMDR_<group_name> where <group_name> is the surveillance group name specified on the command line. An example of an entry for an FMDR server is FMDR_G1 where G1 is the group from which the FMDR server will retrieve data.

Configuring FMDR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

The service name also needs to be specified when configuring a GMDR server to access the FMDR server.

Startup command

The command to start an FMDR server has the following syntax:

```
/opt/MagellanNMS/bin/fmdr \
-g <GroupName> -u <userid> \
-p <password> \
[-l <nameOfAtmLink>]\
[-b <size of the message buffer>] \
[-c <state walk configuration file name>] \
[-e <alarm exceptions file name>] \
[-h] \
[-s] \
[-w <maximum number of concurrent state walks>] \
[-L] \
[-S <timer interval>] \
[-T <post-activation state walk delay in seconds>] \
[-X <maximum number of sieves>] \
[-M <CID>]... \
[-O]
```

where:

`-g <GroupName>` is the name of the FMDR group that the FMDR server is managing

`-u <userid>` is the user ID used to log on to the nodes defined in the group. At minimum, the user ID must have passive impact on the nodes, and a customer ID of 0. You must also have a scope of device or higher to obtain active alarms. Active alarms may be obtained from the node only if the Active Alarm List feature (Passport release PCR 5.1) is installed and provisioned on the node.

`-p <password>` is the password used to log on to the nodes defined in the group or the full path name of the file that contains the encrypted password. Password files are stored in the directory /opt/MagellanNMS/cfg/private.

`-l <nameOfAtmLink>` turns on automatic generation of ATM topology (links). The argument nameOfAtmLink is an arbitrary name to represent the links. Just as TRK is used to represent Trunks, here you need to specify your own value. “AL” or “ATK” are the recommended values.

`-b <size of the message buffer>` specifies the number of congested replies before the client is cut off. If this parameter is not specified, the default value of 5,000 is used. The maximum allowable is 50,000.

`-c <state walk configuration file name>` specifies the full pathname of a state walk configuration file.

FMDR selects the appropriate state walk configuration files for the Passport families. If you use this option to load a particular configuration file, you will override the FMDR-selected configuration files. As a consequence, FMDR will use the specific configuration file that you have loaded for all nodes in the group.

`-e <alarm exceptions file name>` specifies the full pathname of a file specifying the alarm exceptions.

FMDR selects the appropriate alarm exceptions files for the Passport families. If you use this option to load a particular alarm exceptions file, you will override the FMDR-selected files. As a consequence, FMDR will use the specific alarm exceptions file that you have loaded for all the nodes in the group.

`-h` displays the help menu

`-s` enables circuit monitoring. With this parameter, the timer waits 3 minutes before checking whether a circuit should be polled. Use the -S parameter instead of the -s parameter when you want the timer interval to be a value other than 3 minutes. See “Additional configuration for circuit monitoring” (page 405) for an example of how the timer interval influences how often polling occurs.

`-w` is the maximum number of concurrent state walks. The default is 10 state walks.

`-L` turns off automatic generation of link information.

`-X <maximum number of sieves>` specifies the maximum number of sieves. If not specified, the FMDR server defaults to 360.

`-S <timer interval>` enables circuit monitoring. The timer interval is an integer value that represents how long (in minutes) the timer waits before checking whether a circuit should be polled. Use the -S parameter instead of the -s parameter when you want the timer interval to be different from its default value of 3 minutes. See “Additional configuration for circuit monitoring” (page 405) for an example of how the timer interval influences how often polling occurs.

`-T <post-activation state walk delay in seconds>` specifies a delay, in seconds, between state walks after an activation-confirm alarm. The default delay is 900 seconds (15 minutes). A new state walk will not occur if one has already occurred within the specified post-configuration state walk delay time. To change the default, specify a new value (in seconds). If you specify a zero, the post-configuration statewalks are disabled (no state walks after an activation). If you specify one (1), the post configuration state walks occur immediately. If you specify x>= 300, the post configuration throttle is x seconds. If you specify a negative value or between 2 to 299 inclusively, the values are invalid and the FMDR server will not restart.

`-M <CID>` causes the FMDR server to use customer identifier 0 for filtering purposes for a specified CID. When a client authenticates with FMDR, the node returns a CID. If the CID returned matches a CID specified with the -M option, FMDR allows filters on CID 0 for that client. The filtering is not of the wild-card type where information about all components is passed through, as is done for the netman user. The filtering on CID 0 is explicit; only information for components whose CID has explicitly assigned as CID 0 will be passed through. You can specify more than one -M option in the startup command.

`-O` disables the automatic deletion of dynamic components. All components act as ordinary components if this option is used. Redundant FMDRs associated with the same node must have this option set consistently, that is, all FMDRs must enable the feature, or all FMDRs must disable the feature.

> ***Note:*** The command line option -B (babbler) is ignored in this release. If any FMDR entry in the Server Administration tool uses this option, remove this option. For more information about the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

FMDR state walk configuration

Files /opt/MagellanNMS/lib/cfg/FMDR_[ppc|ppe|legacy].cfg specify the node components for which information will be retrieved when FDMR performs a state walk. A record in this file has the form:

```
<tag name format>:   <component name format>
ISDYNAMIC:
QUERY: <query type>
.
.
RETRIEVE: <attribute identifier>
.
.
RETRIEVE: <attribute identifier>
```

where:

`tag name format` specifies the state walk behaviour of components and can be one of the following:

- `OBJECT` is for normal state walk behaviour.
- `OBJECT_DYN` is only for endpoint components. If the state comes back in-service, a state walk is performed only on the component specified in the component name format, and its children.
- `OBJECT_CIRCUIT` identifies a circuit on which you can perform circuit monitoring. See "Additional configuration for circuit monitoring" (page 405) for more information.

`component name format` specifies the component types in the <full component name>, and whether the component is named. For example, EM SHELF CARD * specifies the full name of a card component. In this example, the asterisk (*) indicates that the component may have a name; for example, EM A1 SHELF CARD 0. A dollar sign ($) indicates that the component is not named; for example, EM A1 SHELF.

`ISDYNAMIC` specifies that instances of this component class that are discovered during a statewalk behave as dynamic components. If this tag is not specified in a record, then the components behave as ordinary components.

`query type` specifies any special query properties and can be one of the following:

- `GET_NEXT_COMP` is used for the components that can take advantage of more efficient state walks using get_next capability. Do not edit the information in these records.
- `NO_COMPRESS` is used for the components that cannot take advantage of more efficient state walks using a compressed query. Do not edit the information in these records.

`attribute identifier` is one of the following:

- `OSI_STATE` retrieves all OSI states - operational, administrative, and usage
- `OSI_STATUS` retrieves all OSI status - alarm, procedural, unknown, availability, control, and standby

- `CUSTOMER_ID` retrieves the customer id
- `INTERFACE_NAME_ATM` retrieves the interface name (LP) for an ATM interface
- `INTERFACE_NAME_FRAMER` retrieves the interface name (LP) for a Framer
- `LOGICAL_PROCESSOR` retrieves the LP for an application
- `CARD_TYPE` retrieves the card type for a card
- `INSERTED_CARD_TYPE` retrieves the inserted card type for a card

A comment line starts with an asterisk (*).

If you need to customize this file, first copy it into file /opt/MagellanNMS/cfg/FMDR_[ppc|ppe|legacy].cfg, then modify the file.

> ***Note:*** Is is recommended that you do not customize these files.

Additional configuration for recurrent state walk

File /opt/MagellanNMS/cfg/FMDRStateWalk_<group>.cfg can be configured to schedule the triggering of automatic state walks. As part of its name, this file uses the name of the Passport group that the FMDR server manages. For example, for server FMDR_WESTREGION, the filename is FMDRStateWalk_WESTREGION.cfg. You need to use a separate file for each FMDR server that is to run automatic state walks.

This file is optional, and can contain multiple entries to trigger state walks. (The existence of the file on startup of the FMDR enables recurrent state walks.) Each entry in the file must be separated by a blank line. The format of an entry in this file is as follows:

```
START_TIME: [<day>] [<hh:min>]
INTERVAL: [DAILY|WEEKLY|<hours>]
MODULE: [ALL |<module name>]
```

where:

`<day>` is the day of the week at which the state walks are to begin.Values are Sun, Mon, Tue, Wed, Thu, Fri, and Sat. If this parameter is omitted, the current day is used as the day to begin triggering statewalks.

`<hh:min>` is the time of day at which state walks are to begin. Values are from 00:00 (midnight) to 23:59 (one minute before midnight). If this parameter is omitted, state walks begin at 00:00.

`DAILY|WEEKLY|<hours>` is the frequency at which the state walks are to be triggered once the START_TIME is reached. Values are DAILY, WEEKLY, or the number of hours, for example, 2. If this parameter is omitted, state walks are triggered DAILY.

`ALL |<module name>` specifies the module on which the state walk is to be performed. Values are ALL or the name of a provisioned module. If this parameter is omitted the state walk is performed on all modules managed by the FMDR server. More than one module can be specified, but each module name must be entered on a different line, and must be preceded by MODULE:

Example

To trigger a state walk every two hours and obtain the states of modules EM NODEA03 and EM NODEA04 starting at 00:00 hours of the current day, the file contains the following entries:

```
START: 00:00
INTERVAL: 2
MODULE: EM NODEA03
MODULE: EM NODEA04
```

Note: By default, post-activation and recurrent state walks will not be triggered less than 15 minutes after the last walk on the target node in order to avoid over-stressing it.

Additional configuration for alarm exception handling

The handling of alarm events in FMDR can be controlled by the FMDR alarm exceptions configuration file. Options in this file can be set to ignore proxy alarms, but this file cannot be used to modify the severity of proxy alarms.

File /opt/MagellanNMS/cfg/FMDRAlarmExcep_[ppc|ppe].cfg specifies the actions that are to be taken when certain alarms are received.

Note: If the file does not exist in /opt/MagellanNMS/cfg/, then copy /opt/MagellanNMS/lib/cfg/FMDRAlarmExcep_[ppc|ppe].cfg to /opt/MagellanNMS/cfg/FMDRAlarmExcep_[ppc|ppe].cfg and modify it.

A record in the file has the form:

```
<event type> <fault code> <action>
```

where:

`event type` is MSG, SET, or CLR

`fault code` is the 8-digit number that identifies the type of alarm

`action` is a code identifying the special action that is to be taken for this alarm. The possible action codes are as follows:

CODE	MEANING
0	no action (not an exception); that is, allow the alarm to pass
1	ignore alarm; that is, throw the alarm away
30	set severity to CRITICAL
31	set severity to MAJOR
32	set severity to MINOR
33	set severity to WARNING

Note: There are other action codes that are used internally by FMDR.

A comment line starts with an exclamation mark (!).

Additional configuration for circuit monitoring

FMDR optionally uses a polling mechanism to perform circuit monitoring. With circuit monitoring, a state walk is performed only on the specified circuit components and only at specified polling times.

To enable circuit monitoring, use either the -s or -S option when you start the FMDR server. See “Startup command” (page 398) for more information.

> ***Note:*** Do not monitor too many circuits from the same FMDR process because circuit monitoring can overload the FMDR process and associated FDTR process. The number of circuits you can monitor depends on the characteristics of your installation, such as the number of nodes and the number of circuits per node. If you use circuit monitoring, it is recommended that you start monitoring a small number of circuits and gradually increase the number.

You can only perform circuit monitoring on certain components. If you request circuit monitoring for other components, no information is shown for those components.

You can perform circuit monitoring on the following components:

- EM MPANL DLCI
- EM FRUNI DLCI
- EM FRNNI DLCI
- EM FRMUX DLCI
- EM FRATM DLCI
- EM FRDTE DLCI
- EM GVCIF DLCI
- EM APPN DLCI
- EM ATMIF VCC
- EM ATMIF VPC
- EM ATMIF VPT
- EM ATMIF VPT VCC
- EM VR IP BGP
- EM VR IP BGP PEER
- EM VR IP NHRP

File /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg contains information for performing circuit monitoring state walks. This file contains a polling list that describes the subcomponent types or the specific subcomponent instances that are in a circuit. The polling list also includes the minimum time interval that elapses between query responses.

When you change the contents of this file, you do not need to restart the FMDR server to update the server with the new information. Instead, you can enter the following command to run the passport.kick program and update the server with the new information:

```
/opt/MagellanNMS/bin/passport.kick
```

The format of an entry in this file is as follows:

where:

```
INSTANCE: <component/instance format>
INTERVAL: <polling interval>
```

`component/instance format` specifies the component name and instance. You can specify a specific instance or use the wildcard character (*) to specify a set of instances.

`polling interval` specifies the minimum interval, in minutes, between query responses. The exact polling interval is determined by this parameter, the timer interval (which defaults to 3 minutes), and the response time of the node. You can assign a different value to the timer interval with the -S <timer interval> parameter in the FMDR startup command. See “Startup command” (page 398).

See the following example of FMDR_pollingSurveillance.cfg for more information on how the timing of polling requests is determined.

Example

```
INSTANCE: EM NODEA FRUNI 10 DLCI *
INTERVAL: 10

INSTANCE: EM NODEA FRUNI 11 DLCI 30
INTERVAL: 15

INSTANCE: EM NODEB ATMIF 1 VCC *
INTERVAL: 5
```

In this example, the first two instances use the same timer because they are on the same node. Assume that the timer uses the default interval of 3 minutes.

In the first instance, the timer starts at 10 minutes and checks every 3 minutes to see whether it is time for polling to occur. If it isn’t time, the value of INTERVAL is reduced by 3. In this example the timer checks for polling at 3 minute intervals and sets INTERVAL to 7, 4, and 1 without polling. After the next 3 minute interval, INTERVAL is set to -2 and a polling request is sent to NODEA, if possible. “If possible” means that a full state walk is not in progress. If a full state walk is in progress, circuit monitoring is postponed until the state walk is complete. When a response to the polling request is received, INTERVAL is set back to 10 and the process begins again.

In the second instance, INTERVAL changes to 12, 9, 6, 3, and 0. When INTERVAL becomes 0, a polling request is sent to NODEA, if possible.

Interdependencies

The FMDR server relies on the HGDS server for host and group information. FMDR also relies on the FDTM server to provide an FDTr process, and relies on the FDTr to provide connection and data translation management. For more information on the servers required to support Passport network access, surveillance, and provisioning access, see 241-6001-303 *Preside MDM Administrator Guide*.

Exit codes

Exit codes for the FMDR server are shown in the following table.

Table 62
Exit codes for the FMDR server

Exit code	Description
0	Successful exit. Do not restart the server.
1	Failure. Restart the server.
50	Do not restart (for backward compatibility)
51	Memory resource error (For example, not enough memory). No restart.
(Sheet 1 of 2)	

Table 62 (Continued)
Exit codes for the FMDR server

Exit code	Description
52	Disk resource error (For example, not enough space, no file). No restart.
53	Communication resource error (For example, cannot connect or register). No restart.
54	Timeout/deadlock/congestion (For example, too much congestion). No restart.
55	Bad command line arguments. No restart.
56	Bad configuration file or environment. No restart.
57	Fork exec failure. No restart.
58	Manual server shutdown (admin?). No restart.
59	ipc_init unsuccessful. No restart.
60	ipc service cannot be registered. No restart.
61	Exit signal received. No restart.Other unclassified errors. Restart the server.
100	Query problem to HGDS.
(Sheet 2 of 2)	

Error messages

Error messages output to the System Log Display for the FMDR server are shown in the following table:

Table 63
Error messages for the FMDR server

Error message	Meaning and action
FMDR_<name> -- Invalid command line argument.	Fatal. Correct the command line arguments in the definition for the server with the Server Administration tool.
FMDR_<name> -- Missing mandatory argument	Fatal. One of the following is missing from the command line: -g <passport group> -u <user id> -p <password> Correct the command line arguments in the definition of the server with the Server Administration tool.
FMDR_<name> -- Options -a and -B are incompatible.	Fatal. Correct the command line arguments in the definition of the server with the Server Administration tool.
FMDR_<name> -- Killed client <client> connection due to congestion.	Non-fatal. Check the logs for the problems relating to <client>.
FMDR_<name> -- Unexpectedly lost client <client> connection; error code: <code>	Non-fatal. Check the logs for problems relating to <client>.
FMDR_<name> -- Failed to connect to HGDS.	Fatal. Check the status of the HGDS server with the Server Administration tool.
FMDR_<name> -- Lost HGDS connection.	Fatal. Check the status of the HGDS server with the Server Administration tool.
FMDR_<name> -- Error response from HGDS.	Fatal. Check the HGDS.cfg file for invalid entries.
FMDR_<name> -- connection to FDTM failed... retry in 2 minutes.	Non-fatal. Check the status of the FDTM server with the Server Administration tool. Restart FDTM, if necessary.
FMDR_<name> -- error from FDTM: <error>.	Fatal. Check the status of the FDTM server with the Server Administration tool.
FMDR_<name> -- comm error with FDTM.	Fatal. Check the status of the FDTM server with the Server Administration tool. Try restarting the FDTM server.
(Sheet 1 of 5)	

Table 63 (Continued)
Error messages for the FMDR server

Error message	Meaning and action
FMDR_<name> -- lost connection to FDTM.	Fatal. Check the status of the FDTM server with the Server Administration tool. Restart FDTM, if necessary.
FMDR_<name> -- could not connect to FDTR: <name>	Fatal. Check the logs for the FDTR problem.
FMDR_<name> -- could not create session with FDTR: <name>	Fatal. Check the logs for the FDTR problem.
FMDR_<name> -- is now connected to <fdtr>.	Message only.
FMDR_<name> -- lost connection to <fdtr>.	Fatal. Check the logs for the FDTR problem.
FMDR_<name> -- error response from <fdtr>: <error>.	Fatal. Check the logs for the FDTR problem.
FMDR_<name> -- unknown response type from <fdtr>.	Fatal. Check the logs for the FDTR problem.
FMDR_<name> -- error response from <fdtr>.	Fatal. Check the logs for the FDTR problem.
FMDR_<name> -- error response on state walk for <component>: <error>.	Non-fatal. The state walk has been discontinued for this component. Check the FDTR logs for additional information.
FMDR_<name> -- Unable to find FMDR_[ppc|ppe|legacy].cfg file.	Fatal. Ensure that there is an FMDR_[ppc|ppe|legacy].cfg file in directory /opt/MagellanNMS/lib/cfg.
FMDR_<name> -- Unable to find /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. FMDR has been started with circuit monitoring enabled but the circuit monitoring cfg file does not exist. Without the file, no components can be circuit monitored. Create the file; it can be empty.
FMDR_<name> -- no interval specified at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
(Sheet 2 of 5)	

Table 63 (Continued)
Error messages for the FMDR server

Error message	Meaning and action
FMDR_<name> -- Unrecognized label at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
FMDR_<name> -- <instance> is not a legal type for circuit monitoring at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
FMDR_<name> -- Interval is not a positive integer at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
FMDR_<name> -- no instance specified at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
FMDR_<name> -- Second INSTANCE definition at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
FMDR_<name> -- Second INTERVAL definition at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
FMDR_<name> -- Invalid instance type: <instance> at line <x> in file /opt/MagellanNMS/cfg/FMDR_pollingSurveillance.cfg	Non-fatal. The entry is skipped due to the error found when parsing the entry in the file. Correct the entry and use the passport.kick utility to trigger FMDR to reparse and restart circuit monitoring.
(Sheet 3 of 5)	

Table 63 (Continued)
Error messages for the FMDR server

Error message	Meaning and action
FMDR_<name> -- EM <nodename> is not a member of this group.	Non-fatal. The FMDR server is reporting on module EM/<nodename> which is not defined in group <name> in /opt/MagellanNMS/cfg/HGDS.cfg. This is likely caused by a mismatch between the name provisioned on a node and the name entered in the HGDS.cfg file. Correct the HGDS.cfg file to match the name provisioned and restart HGDS, FDTM, and FMDR_<name> servers.
FMDR <name> -- EM <module name>: Either the module name on switch does not match the module name in the HGDS server or the on switch boot activation is not complete.	Non-fatal. If the name given to the node does not match the name that is specified in the /opt/MagellanNMS/cfg/HGDS.cfg file, update the HGDS.cfg file, and then stop and restart the FMDR server. If the names match, a state walk will be done automatically once the node has fully activated.
FMDR <name> -- error response on state walk for <instance> APPLICATION ERROR 1120 Fdtr received an invalid response from device Sw. Component is owned by another customer.	Non-fatal. FMDR is using the -M command line option with a value different than 0. FMDR cannot discover the Passport family type (see “Detection of Passport family” (page 392)). FMDR will use the default schemas for state walk, and it will not be able to retrieve any active alarm list.
FMDR <name> -- Unable to retrieve the Active Alarm List from module <instance>, proxy alarms will be generated.	Non-fatal. The node supports the Active Alarm List feature, but FMDR cannot obtain the active alarms because too many FMDRs requested the alarms at the same time. Proxy alarms are generated to represent the out-of-service components.
FMDR <name> -- The Active Alarm List from node <instance> was retrieved.	Non-fatal. The previous log message was generated. FMDR completed two state walks on the module, and it retrieved the active alarm list from the node.
FMDR <name> -- State walk started for every module of the group.	Non-fatal. One state walk has started for each module of the FMDR group.
FMDR <name> -- State walk started on module EM <module name>.	Non-fatal. One state walk has started for the module.
(Sheet 4 of 5)	

Table 63 (Continued)
Error messages for the FMDR server

Error message	Meaning and action
FMDR <name> -- State walk on module EM <module name> cancelled.	Non-fatal. The state walk in progress on the module was cancelled. The log is displayed for the following reasons: - loss of connectivity to the node while a state walk is in progress - manually triggering a Reset Database of FMDR, using GMDR Admin, while a state walk is in progress on that node - using GMDR Admin to delete a module while a state walk is in progress on that node - the name of the node does not match the name of the /opt/MagellanNMS/cfg/HGDS.cfg file.
FMDR <name> -- Walk already in progress or pending for module EM <module name>.	Non-fatal. A resynchronization was triggered on the module while it is currently being state walked.
FMDR <name> -- The babbler option (-B) was specified for this FMDR. This option is no longer supported and will be ignored.	Non-fatal. The option is ignored. FMDR execution continues. ***Note:*** The command line option -B (babbler) is ignored in this release. If any FMDR entry in the Server Administration tool uses this option, remove this option. For more information about the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.
FMDR <name> -- There was at least one query failure during the nodal state walk because of the following errors: <reason>. The reason could be 1121, 1122, 1137 and their combinations.	Non-fatal. Check the error summary for the FDTM server to diagnose the problem.
(Sheet 5 of 5)	

Chapter 52

Passport Nodal Provisioning Configuration Server (PCSERVER)

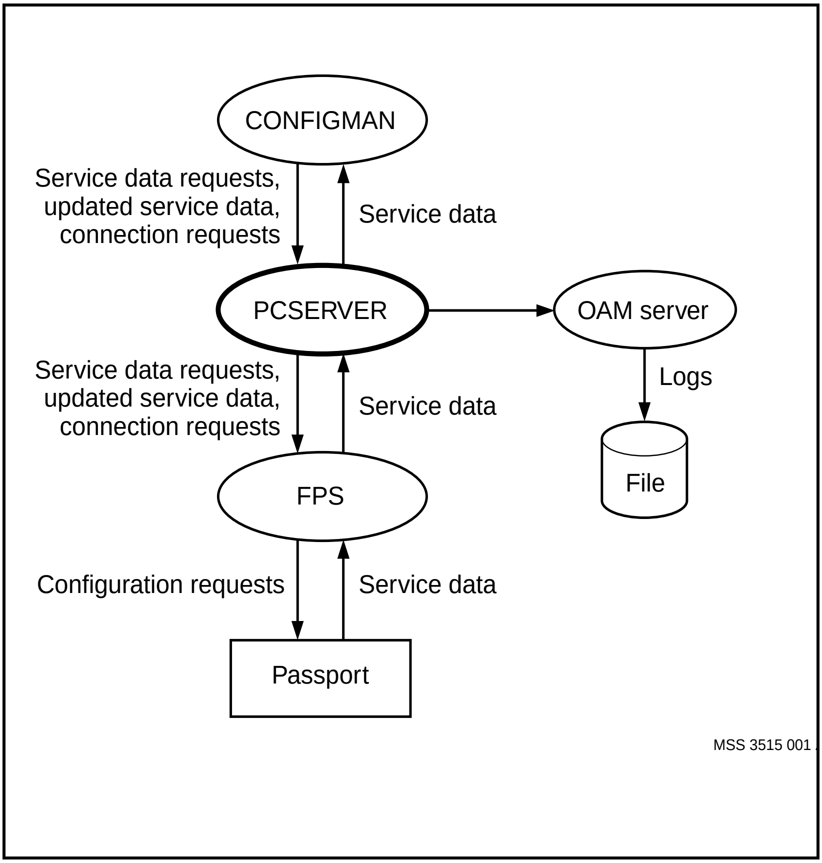
This section contains information on the Passport Configuration server (PCSERVER). See the following topics for more information:

- “About the PCSERVER” (page 415)
- “Managing the PCSERVER” (page 416)
- “Exit codes” (page 418)

About the PCSERVER

The PCSERVER is used by the Nodal Provisioning interface. The PCSERVER creates and manages the Passport Configuration Providers (FPS), which provide configuration access to Passport nodes. For each open connection request received from the Configuration Manager, the PCSERVER creates a new FPS to handle configuration requests for a particular node. For more information on the Configuration Manager, see “Nodal Provisioning Configuration Manager (CONFIGMAN)” (page 331).

Figure 46
PCSERVER data flow and logging diagram


CONFIGMAN
Service data requests, updated service data, connection requests
Service data
PCSERVER
OAM server
Logs
File
Service data requests, updated service data, connection requests
Service data
FPS
Configuration requests
Service data
Passport
MSS 3515 001

Managing the PCSERVER

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the PCSERVER. Any changes you make to the startup command or options take effect when the PCSERVER is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 417)

- “Startup command” (page 417)

Suggested name in Server Administration

The recommended name for the PCSERVER is PP NP Config Server.

Configuring the PCSERVER with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

Use the following command to start the PCSERVER.

```
/opt/MagellanNMS/bin/pcserver [-port <port number>]
[-bckdir <backup directory>] [-h]
```

where:

`-port <port number>` specifies the port number that the PCSERVER uses. If you specify this parameter, the port number you specify must be the same as the configuration server port number in the CONFIGMAN startup command. For more information, see the -Dcsport parameter in “Startup command” (page 333). If you do not specify this parameter, PCSERVER dynamically sets up the port based on the specifications in the configuration file /opt/MagellanNMS/cfg/private/IPCNameMap.cfg . If there is no configuration file, PCSERVER determines the appropriate port number.

`-bckdir <backup_dir>` specifies the backup directory that FPS accesses to load view and journal files. The default backup directory is /opt/MagellanNMS/data/Backup_Data/PASSPORT. If you do not specify a backup directory, FPS uploads the view and journal files from the node

`-h` displays information about the pcserver command.

Example

In the following example, errors are not directed to a log file and the port number is obtained from a configuration file.

```
/opt/MagellanNMS/bin/pcserver
```

Exit codes

Exit codes for when the PCSERVER terminates under its own control are shown in the following table. The PCSERVER can terminate with other exit codes if it crashes or is stopped by the Server Administration tool.

Table 64
Exit codes for the PCSERVER

Exit code	Description
0	Successful exit
50	Unable to connect to Connection Manager
51	Memory error
53	Communication error - port already used
55	Invalid argument
59	Unable to initialize IPC
61	Exit signal received
62	Terminated due to licensing problem
100	Other unclassified errors

Chapter 53 Passport Restore Provider (PRSTPP)

This section contains information on the Passport Restore Provider (PRSTPP). See the following topics for more information:

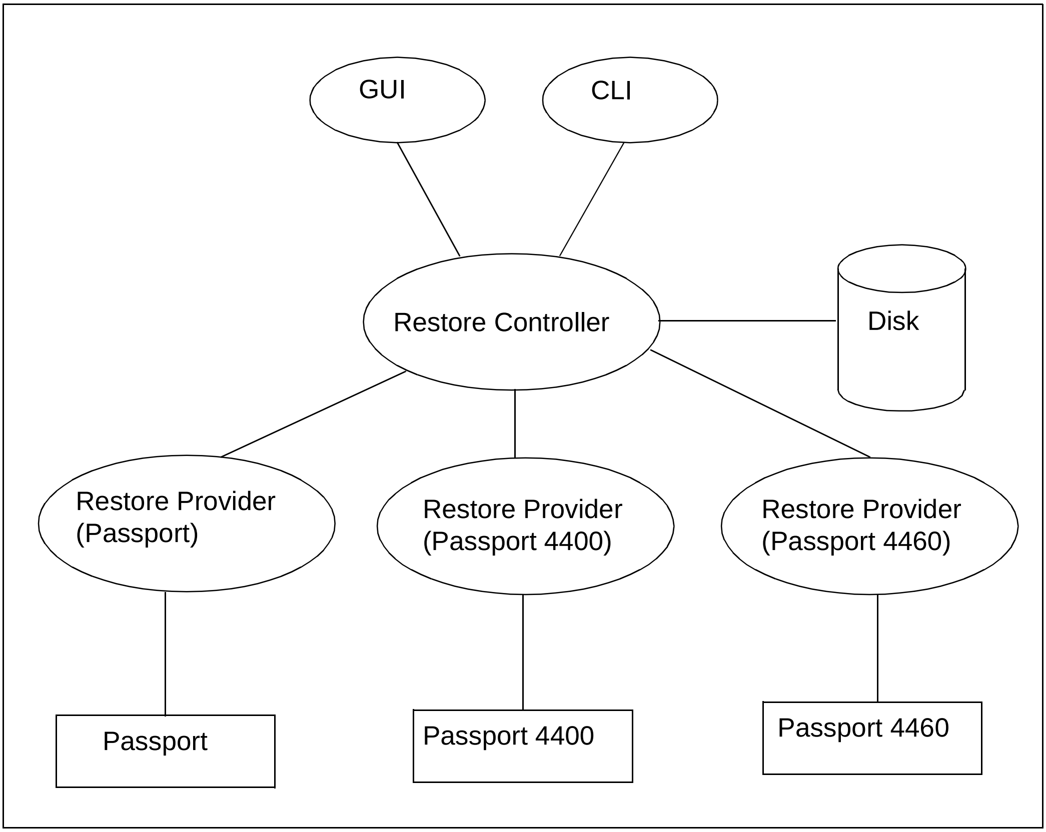
- “About the Restore Provider” (page 419)
- “Managing the Restore Provider” (page 420)

About the Restore Provider

The Restore Controller and Restore Provider are responsible for restoring the backed up views to the Passport node.

See 241-6001-807 *Preside MDM Network Backup and Restore*.

Figure 47
Restore Provider data flow diagram


GUI
CLI
Restore Controller
Disk
Restore Provider (Passport)
Restore Provider (Passport 4400)
Restore Provider (Passport 4460)
Passport
Passport 4400
Passport 4460

Managing the Restore Provider

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 421)
- “Startup command” (page 421)

Suggested name in Server Administration

The recommended name is Restore Provider.

Configuring the Restore Provider with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the Restore Provider is as follows:

```
/opt/MagellanNMS/bin/prstpp [-p <port_no>]
```

Use the following table to substitute command parameters:

Parameter	Definition
port_no	is the port number the Restore Provider uses. The port number is dynamically assigned but you can also specify it as port number. The default port is port 5021.

Chapter 54
Fault Device Access Agent (PSVAGENT)

This section contains information about the Fault Device Access Agent (PSVAGENT). See the following topics for more information:

- "About the PSVAGENT" (page 423)
- "Suggested name in Server Administration" (page 424)
- "Startup command" (page 425)
- "PSVAGENT schema files" (page 426)
- "Error messages" (page 427)

About the PSVAGENT

The PSVAGENT is a server that gathers and sends node information to the Passport Shelf View, MPE 9500 Shelf View, Command Console (Java version), and Operator Commands. The PSVAGENT provides an interface to Preside Multiservice Data Manager (MDM) servers that send commands to devices in the network and receive responses to those commands.

The PSVAGENT gathers information from several sources including the MDM Network Model, the alarm stream for GMDR, and from the node through the Passport Command stack. Since PSVAGENT requires information from various sources, the following servers must be running:

- Passport Communications Manager (FDTM) (Passport Shelf View only)
- FMIP Management Data Router (FMDR) (Passport Shelf View only)
- General Management Data Router (GMDR)

- Host Group Directory Server (HGDS)
- Network Model Coordinator (DNMNMC)
- Surveillance Network Model Updater (SURNUP)
- Network Model Server (NMServer)
- NMS Log Collector (OAMC)
- NMS Context Server (CTXSVR)
- Multi-nodal Name Server (MNSD)
- Multi-nodal Name Server Agent (MNSDAGENT)
- MPE 9500 Communications Manager (NDTM)
- MPE 9500 Management Data Router (NMDR)

Using the Service Selection facility, the following minimum set of servers must always be local for the Passport Shelf Display tool to operate:

- Command Functional Process (CMCFUN) local to the session
- Connection Manager (CM) local to the session
- NMS Context Server (CTXSVR)
- Multi-nodal Name Server (MNSD)
- Multi-nodal Name Server Agent (MNSDAGENT)

Suggested name in Server Administration

The recommended name for the Fault Device Access Agent is Fault Device Access Agent.

Configuring PSVAGENT with the Server Administration tool requires you to enter the server name in the **Descriptive name** field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the PSVAGENT is as follows:

```
/opt/MagellanNMS/bin/psvagent \
[-h|?] \
[-walk <filename>] \
[-trigger <filename>] \
[-link <filename>] \
[-gmdr_host <hostname>] \
[-nm_host <hostname>] \
[-acc_host <hostname>] \
[-cc-time <timeout>]
[-srs_walk]
[-srs_model]
[-srs_trigger]
[-srs_link]
```

where:

`[-h|?]` displays the syntax and command options for the PSVAGENT’s startup command.

`[-walk <filename>]` walks the schema file (optional).

`[-trigger <filename>]` triggers the schema file (optional).

`[-link <filename>]` links the schema file (optional).

`[-gmdr_host <hostname>]` identifies the GMDR host to which you want to connect.

`[-nm_host <hostname>]` identifies the Network Model host to which you want to connect.

`[-acc_host <hostname>]` identifies the Passport Access host to which you want to connect.

`[-cc-time <timeout>]` sets the sleep timeout, in seconds, before attempting server connections. The default sleep time is 10 seconds.

`[-srs_walk]` walks the SRS_Walk.cfg file.

`[-srs_model]` uses the SRS_Model.cfg file for the model schema.

`[-srs_trigger]` triggers the SRS_Trigger.cfg file for the trigger schema.

`[-srs_link]` uses the SRS_Link.cfg file for the link schema.

PSVAGENT schema files

PSVAGENT uses the following schema files:

- PSV_Walk.cfg
  This configuration file identifies the node components and attributes required as input to the Passport Shelf Display tool.
- PSV_Trigger.cfg
  This file identifies which alarms trigger a new state walk of the node.
- PSV_Link.cfg
  This file specifies the links for node components in the component navigation panel of the Passport Shelf Display tool.
- SRS_Model.cfg
  This file describes the components that are modelled and the attributes that are stored.
- SRS_Walk.cfg
  This file specifies the walk requests made against the switch. A single walk request could gather information on several different components. The MDP_Model.cfg files specifies what information from the walk is filtered out.
- SRS_Trigger.cfg
  This file specifies the alarms that act as triggers and their scope.
- SRS_Link
  This file specifies the component attributes that contain link information. The Component Navigator shows links between components as hyperlinks in the Component Navigator tree.

The default copy of these configuration files are located in the /opt/MagellanNMS/lib/cfg directory. User customized copies of these files are located in the /opt/MagellanNMS/cfg directory.

Error messages

The following PSVAGENT error messages are logged to the System Log display.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Failed to register PSV service with IPC	The Fault Device Access Server is already running and does not need to be started again.
Failed to register CCAGENT service with IPC	An older version of ccagent is currently running.
The Fault Device Access Server could not connect to GMDR on <hostname>.	The Passport Shelf Display fails to connect to GMDR.
The Fault Device Access Server could not connect to the NM Server	The Passport Shelf Display fails to connect to the NM Server.
The psvagent could not connect to or lost its previous connection to the cmcfun session.	The Passport Shelf Display fails to connect to the cmcfun server. The psvagent only runs when an Preside Multiservice Data Manager session is active and, consequently, the session specific Passport Command stack processes are also running. This error usually indicates an invalid or nonexistent session.
An error was found in the PSV_Walk.cfg file. An error was found in the PSV_Trigger.cfg file. An error was found in the PSV_Link.cfg file.	An error occurred during the parsing of the configuration file.
The lock/unlock command failed.	The Lock or Unlock command fails to execute.
An invalid link schema file was given in the - srs_link command option line.	The file name given by the -srs_link parameter on the psvagent's command line does not exist.
NSV_Link.cfg not found. NSV_Model.cfg not found. NSV_Trigger.cfg not found NSV_Walk.cfg not found.	The file NSV_Link.cfg does not exist in the /opt/MagellanNMS/lib/cfg directory or the /opt/MagellanNMS/cfg directory and one of the following commands was not given in the command line: -srs_link, -srs_model, -srs_trigger, or -srs_walk override parameter.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Parsing Error: unrecognized record type: <string>. See line <line number> in file NSV_Link.cfg. See line <line number> in file NSV_Model.cfg. See line <line number> in file NSV_Trigger.cfg. See line <line number> in file NSV_Walk.cfg.	There is an error in the <filename>.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/ cfg/<filename>.cfg or the file specified by one of the following command line options: -srs_link, -srs_model, -srs_trigger, or -srs_walk
Parsing Error: Line too long. See line <link number> in NSV_Link.cfg.	There is an extremely long line in the NSV_Link.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/ cfg/NSV_Link.cfg or the file specified by the -srs_link command. line option.) The maximum line length is 1024 characters.
An invalid walk schema file was given in the -srs_model command line option.	The file name given by the -srs_model parameter on the psvagent's command line does not exist.
Parsing Error: bad state value: <string> See line <line number> in file NSV_Model.cfg.	There is an error in the NSV_Model.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/cfg/ NSV_Model.cfg or the file specified by the -srs_model command line option.)The only valid instance values are “$” and “*”.
Parsing Error: bad internal value: <string> See line <line number> in file NSV_Model.cfg.	There is an error in the NSV_Model.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/cfg/ NSV_Model.cfg or the file specified by the -srs_model command line option.)The only valid state values are “YES” and “NO”.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Parsing Error: GET has no attribute See line <line number> in file NSV_Model.cfg.	There is an error in the NSV_Model.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/cfg/ NSV_Model.cfg or the file specified by the -srs_model command line option.) The GET field requires an attribute value.
Parsing Error: no COMP_ID: in record See line <line number> in file NSV_Model.cfg.	There is an error in the NSV_Model.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/cfg/ NSV_Model.cfg or the file specified by the -srs_model command line option.) The COMP_ID field in a record is mandatory.
Parsing Error: no INSTANCE: in record See line <line number> in file NSV_Model.cfg.	There is an error in the NSV_Model.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/cfg/ NSV_Model.cfg or the file specified by the -srs_model command line option.) The INSTANCE field in a record is mandatory.
Parsing Error: Line too long See line <line number> in file NSV_Model.cfg.	There is an error in the NSV_Model.cfg file the psvagent uses. The user should check the data in the file (either the file location /opt/MagellanNMS/cfg/ NSV_Model.cfg or the file specified by the -srs_model command line option.) A line in this file can be at most 1024 characters long.
Error from cmcfun: <cmcfun error message>	The psvagent received an error from its cmcfun subserver, but this error is not severe enough to compromise the communication between the psvagent and cmcfun.
Lost connection to command server	The cmcfun server the psvagent was using is unavailable for an unknown reason.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Noncritical CMCFUN error: <cmcfun error message>	The psvagent received an error from its cmcfun subserver, but this error is not severe enough to compromise the communication between the psvagent and cmcfun.
An invalid trigger schema file was given in the -srs_trigger command line option.	The file name given by the -srs_trigger parameter on the psvagent's command line does not exist.
Parsing error: invalid fault code. See line <line number> in file NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). The value in the FAULT_CODE field is not a valid fault code or prefix to a fault code.
Parsing Error: SET should be YES or NO. See line <line number> in NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS /cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). The value of a SET field should be either "YES" or "NO".
Parsing Error: CLEAR record is not YES or NO. See line <line number> in file NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). The value of a CLEAR field should be "YES" or "NO".

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Parsing Error: MESSAGE missing YES or NO. See line <line number> in file NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). The value of a MESSAGE field should be "YES" or "NO".
Parsing Error: ACTIVATION missing YES or NO. See line <line number> in file NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). The value of a MESSAGE field should be "YES" or "NO".
Parsing Error: Schema Record not complete See line <line number> in NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). Not all of the mandatory fields for a trigger record were provided.
Parsing Error: Line too long. See line <line number> in NSV_Trigger.cfg.	There is an error in the NSV_Trigger.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Trigger.cfg, or the file specified by the -srs_trigger command line option). A line in the NSV_Trigger.cfg file cannot be more than 1024 characters long.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Parsing Error: Line too long. See line <line number> in NSV_Walk.cfg.	There is an error in the NSV_Walk.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Walk.cfg, or the file specified by the -srs_walk command line option). A line in the NSV_Walk.cfg file cannot be more than 1024 characters long.
An invalid walk schema file was given in the -srs_walk command line option.	The file name given by the -srs_walk parameter on the psvagent's command line does not exist.
Parsing Error: missing instance value. See line <line number> in file NSV_Walk.cfg.	There is an error in the NSV_Walk.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Walk.cfg, or the file specified by the -srs_walk command line option). The instance value should be "$" or "*".
Parsing Error: Schema Record invalid See line <line number> in NSV_Walk.cfg.	There is an error in the NSV_Walk.cfg file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/NSV_Walk.cfg, or the file specified by the -srs_walk command line option). The record should be a single "COMP_ID:" field. There should be at least one blank line between records.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
An invalid walk schema file was given in the -walk command line option An invalid trigger schema file was given in the -trigger command line option An invalid link schema file was given in the -link command line option An invalid walk schema file was given in the -msv_walk command line option. An invalid model schema file was given in the -srs_model command line option. An invalid trigger schema file was given in the -srs_trigger command line option. An invalid link schema file was given in the -srs_link command line option.	Check the command line parameters and make sure a valid filename follows one of the following applicable commands: -walk, -trigger, -link, -msv_walk, -srs_model, -srs_trigger, -srs_link. The filename cannot start with a '-'
An invalid host was given in the -gmdr_host command line option. An invalid host was given in the -nm_host command line option. An invalid host was given in the -acc_host command line option.	Check the command line parameters and make sure a valid workstation name follows one of the following command line options: -gmdr_host, -nm_host, -acc_host.
An invalid value was given for the -cc_time command line option. An invalid value was given for the -conn_check_time command line option.	Check the command line parameters and make sure a valid positive integer value follows one of the following command line options: -cc_time, -conn_check_time.
The following bad command line option was entered: <string>.	The psvagent does not recognize the command line option.
Software error with NS communications.	A critical internal error occurred on the protocol between the client and the server. Make sure that the server is not an older software version than the client.
Could not load config file <file name>.	The file at the specified name and location could not be found.

Table 65
Error messages for Fault Device Access Server

Error message	Meaning
Passport Shelf View has an exceptionally long string in the link configuration file <file name>.	There is an error in a schema file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/<file name>, or the file specified by the command line option). The maximum length for a line in a schema file is 1024 characters.
Parsing Error: <string> See line <line number> in file <file name>.	There is an error in a schema file the psvagent uses. The user should check validate the data in the file (either the file in the location /opt/MagellanNMS/cfg/<file name>, or the file specified by the command line option). Check the line of the file specified and fix the error.
Failed to register PSV service with IPC Failed to register NSVAGENT service with IPC	There is already a psvagent process running on this workstation, or mnsd is not running.

Chapter 55
Performance Measurement Stream Processor (PMSP)

The Performance Measurement Stream Processor (PMSP) server is used only in a Succession Network solution. If you have a Succession Network solution, see the following sections for information about the PMSP server:

About the PMSP server

The PMSP server collects and converts PM statistics from Passport processors and ATM interfaces. Performance measurement information is required for network planning and engineering.

> ***Note:*** To use this application, nodes require Passport software version PCR4.1, or later.

The process of collecting, transferring and converting PM statistics includes

1. At 5-minute intervals the nodes AtmCore, AtmNetworking, and PCS applications on processor cards and ATM interfaces collect PM statistics and data with the data type of rtStats that is generated by the Data Collection System (DCS).
2. Each node sends the PM statistics to the resident DCS where records of the *rtStats* data type are created for each application.

3 The Data Collection System (DCS) forwards collected PM statistics to the Preside Multiservice Data Manager (MDM) workstation as a stream of fast management information protocol (FMIP) encoded binary data. The PM statistics are transferred to the workstation at 5-minute intervals.

   5-minute intervals are synchronized to five minutes after the hour and beginning with the nearest 5-minute time increment (time of day on the node and the workstation must be synchronized).

   Note: PM statistics data cannot be spooled.

4 MDM converts the PM statistics to ASCII comma separated values (CSV) data format.

5 5-minute interval data can be stream transferred to SuperNode Data Manager (SDM)s in CSV format on registered network connections using TCP sockets.
   5-minute interval data can be written to CSV files on the workstation.

6 On the hour and half-hour, the data for all 5-minute intervals during the past 30-minute interval are aggregated.

7 30-minute interval data can be stream transferred to SDMs in CSV format on registered network connections using TCP sockets.
   30-minute interval data can be written to CSV files on the workstation.

This section includes descriptions of

- “5-minute interval data” (page 436)
- “30-minute interval data” (page 438)
- “Connectivity” (page 439)
- “Compensating for network time changes” (page 442)

5-minute interval data

This section describes several considerations with the 5-minute interval data and includes

- “Missing data” (page 437)
- “Records out of time synchronization” (page 437)

For detailed information on 5-minute interval data, refer to the customer-specific documentation.

Missing data

When values are missing from the attribute fields, there will be consecutive commas in the CSV-format record with no value in between. Missing values should be a rare occurrence, although it can occur if the number of interfaces being reported exceeds the engineering latency. That is, when growth scenarios cause more ATM interfaces to be added, if the initially configured PM Stream Processor latency is to small to handle the extra processing requirements, some raw information may not be processed in time. Of course, the latency configuration should be adjusted if this condition persists.

If no records are received during the data collection period for a particular ATM interface, a 5-minute interval ATM interface record is not generated for that ATM interface for that interval. If a record contains corrupted data, the PM Stream Processor writes the entire raw FMIP record to file /opt/MagellanNMS/data/pmsp/ppc_5min.err. This file can be used to recover partial data or to help determine the reason for the corruption.

Records out of time synchronization

If a record does not belong to the current interval, a 5-minute interval ATM interface record is not generated for that record as it has already missed its five-minute interval period.

If the record belongs to an interval for which 30-minute interval data processing is currently in progress, the record is incorporated into the respective 30-minute interval data. The PM Stream Processor issues a warning that indicates that a record from a previous 5-minute interval has been received from a particular node. This warning is displayed on the System Log Display tool and only one warning is issued for each node for each occurrence. See “Error messages” (page 445).

At the end of each 5-minute interval, MDM sets a timer according to the value of the -ppexpire command line option. This value is the latency time-out. Any records received from nodes during that time-out window are accepted into the 5-minute interval data. Any records received after the timer expires are discarded as the 5-minute interval data, and included in the 30-minute data for the current 30-minute interval. If the 30-minute interval CSV stream is complete, the record is counted in the next interval or possibly discarded and logged for the 30-minute interval.

If the record does not belong to the current interval and its 30-minute interval CSV data stream has already been completed, the record is discarded only if the -nosync option has not been specified.

30-minute interval data

The PM Stream Processor creates a single 30-minute interval data stream every half hour on the hour and the half hour. This data stream summarizes the PM statistics from the 5-minute interval ATM interface records on a per ATM interface basis. The 30-minute interval record can include data from six 5-minute intervals.

For detailed information on 30-minute interval data, refer to the customer-specific documentation.

Using the temporary autosave file

After it is compiled, the 30-minute interval record is written to the following temporary binary file

```
/opt/MagellanNMS/data/pmsp/
<30-minute port>_PPC_30min.autosave
```

Because the 30-minute interval record is resident in memory, the temporary file serves as a backup in case MDM is shut down or loses power. The presence of this file in the /opt/MagellanNMS/data/pmsp directory indicates that the 30-minute interval has not expired and that the data for the interval has not been sent to the SuperNode Data Manager (SDM).

If the TCP socket sessions from MDM are interrupted, and MDM and the PM Stream Processor are restarted and re-establish connectivity during the same 30-minute interval, the contents of the autosave file are repopulated into memory and the PM Stream Processor continues aggregating PM statistics for the remainder of the 30-minute interval. If MDM and the PM Stream Processor are restarted after the expiry of the 30-minute interval, the previously open 30 minute autosave file is closed and a new one is opened for the new 30-minute interval period.

The contents of the autosave file are written to an ASCII based historical file /opt/MagellanNMS/data/pmsp/
<30-minute port>_PP_30MIN_PM_YYYYMMDDThhmmss.csv. The timestamp in the filename of the historical file represents the end time of the 30-minute interval to which the file belongs. The contents of the historical file are identical to the CSV data stream which is sent to the SDM, with the exception of the timestamp and record lengths, which are not included in the historical file.

SDM processing for the 30-minute interval record

After expiry of the 30-minute interval and incorporation of the last 5-minute interval record, the PM Stream Processor transmits the contents of the 30-minute interval record to the SuperNode Data Manager (SDM) as a CSV data stream. Transmission occurs over the dedicated 30-minute TCP socket. The format of the 30-minute interval data stream is identical to the 5-minute interval CSV data stream. The TCP socket that is used for the 30-minute data is separate from the TCP socket used for the 5-minute data.

Upon successful transmission of the 30-minute interval record to the SDM, the SDM sends a successful transmission acknowledgement flag to the PM Stream Processor. The PM Stream Processor then deletes the 30-minute autosave file /opt/MagellanNMS/data/pmsp/ppc_30min.autosave.

The PM Stream Processor allocates a configurable interval (based on the value of the *-ppexpire* command line option) from the start of transmission of the 30-minute CSV data stream (that is, 30 seconds after the end of the 30-minute interval) for the receipt of the successful transmission acknowledgement flag from the SDM. If the PM Stream Processor does not receive the flag within this period, it assumes that the SDM has not received the 30-minute CSV data stream. The PMSP then creates an historical file /opt/MagellanNMS/data/pmsp/
<30-minute port>_PP_30MIN_PM_YYYYMMDDThhmmss.csv.

Connectivity

The node and the SuperNode Data Manager (SDM) are connected through Internet Protocol (IP) to the workstation. The node, the SDM, and workstation are time synchronized.

WARNING
Loss of connectivity
Timing between the node, the Preside Multiservice Data Manager workstation, and the SDM must be properly synchronized before connections can be established. For information and procedures to configuring timing, see 241-6001-303 *Preside MDM Administrator Guide*

This section includes

- “Connectivity to SDM” (page 440)
- “Connectivity to the node” (page 440)
- “Recovery from loss of connectivity” (page 441)

Connectivity to SDM

The SuperNode Data Manager (SDM) establishes two TCP socket connections with the PM Stream Processor. The connections support the transmission of the 5- and 30-minute data, one connection for each data type.

When a TCP socket connection is established, the PM Stream Processor sends a start flag (which represents the length of the header) and a header (which details the attribute field ordering of the 5-minute ATM interface record and the 30-minute interval record). The header is sent once to the SDM. The PM Stream Processor does not include the header in any of the subsequent CSV data streams sent for the subsequent intervals.

Connectivity to the node

When the PM Stream Processor is started, it sets up an NMIS/FMIP session on a best effort basis on the node. The NMIS/FMIP session is used for the transmission of statistics data between the node and the MDM workstation. The PM Stream Processor then enables the rtStats and disables all other streams. The objective of this second event is to prevent these streams from transmitting during the NMIS/FMIP session. In the event that an NMIS/FMIP session cannot be established, the PM Stream Processor generates an error message for that switch and the error message is displayed on the System log display tool.

MDM also establishes a separate NMIS/FMIP session for surveillance of the node. Surveillance is possible by enabling alarm and SCN data streams. MDM disables the *rtStats* data stream for the surveillance session so that there is no interaction with statistics transmission sessions. A node supports up to 35 concurrent NMIS/FMIP sessions.

Recovery from loss of connectivity

If the MDM workstation loses connectivity with the SuperNode Data Manager (SDM) during transmission of the 30-minute data stream, the SDM immediately attempts to re-establish the connection using the same TCP sockets.

When successfully connected, the SDM sends an ISO timestamp (YYYYMMDDThhmmss) on the 30-minute TCP socket. This timestamp indicates whether or not the SDM wants the interrupted 30-minute data CSV stream transmission. If requested, the PM Stream Processor re-transmits the 30-minute historical files (which are created later than the timestamp) in the /opt/MagellanNMS/data/pmsp directory in chronological order.

The SDM is responsible for the management of any duplicate records received in the case of loss of connectivity with the workstation. Upon successful retransmission to the SDM, the PM Stream Processor deletes the successfully transmitted historical files from the /opt/MagellanNMS/data/pmsp directory.

If the SDM does not want the interrupted historical data, the PM Stream Processor deletes the 30-minute historical files in the /opt/MagellanNMS/data/pmsp directory.

If the workstation loses connectivity to a node, the PM Stream Processor attempts to re-establish the connection and set up a new NMIS/FMIP session at 5-second intervals. If the workstation loses connectivity during transmission of statistics records, the single record in the NMIS buffer is lost. The interface between the node NMIS and DCS is flow-controlled so that only one record is handled by NMIS at any time. The remaining records are queued in DCS and are transmitted when the PM Stream Processor re-establishes the connection and sets up the new NMIS/FMIP session.

In the event that an NMIS/FMIP session cannot be established, the DCS continues to collect statistics records until the DCS queue for the rtStats data type is full. When the queue is full, the DCS discards any additional records until the queue begins to empty.

Compensating for network time changes

If the network time changes (for example, because of a change to Daylight Savings Time) the PM Stream Processor takes no special action, but the switch back from Daylight Savings Time does require additional management. When this seasonal time change occurs, for the hour of time that is typically repeated (in North America, this hour usually falls between 1:00 and 2:00 am) CSV files have the same names as similar files produced an hour earlier, prior to the time change.

This potential problem requires that the OSS application either move or rename the older files before the new ones with the same name are created. The application can also wipe out the older files if they were left in their original location. This potential problem applies to both SDM and files created by Preside Multiservice Data Manager (MDM). For details concerning seasonal time changes, refer to the customer-specific documentation.

However, if there is a requirement for network time to be changed, it is recommended that the change occur after the transmission of the 30-minute interval data has been successfully completed and before the receipt of the data from the first 5-minute interval (that is, prior to 5 minutes and 35 minutes past the hour).

> ***Note:*** If the time on the network is changed, the time on the MDM must also be changed to remain synchronized with the network.

Suggested name in Server Administration

The recommended name for the Performance Measurement Stream Processor is PMSP.

Configuring PMSP with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Start-up command

Use the Server Administration tool to automate PMSP startup and monitoring. Using the Server Administration tool ensures reactivation of the PMSP after process or platform failure. For more information about the Server Administration tool, see 241-6001-303 *Preside MDM Administrator Guide*.

> ***Note:*** Multiple instance of the PMSP can be running on a Preside Multiservice Data Manager workstation. They must be launched on separate 5-minute and 30-minute ports.

Command syntax

The syntax of the PMSP command is

```
/opt/MagellanNMS/bin/pmsp
-g <Passport group name>
-u <Passport userID>
-p <Passport userID[password|passwordFile.pwd]
[-5port <5 minute TCP port>]
[-30port <30 minute TCP port>]
[-ppexpire <seconds>]
[-savefile -hgds|-shelf]
[-nosync]
[-h|-help]
```

where:

`-g <Passport group name>`

is the name of the Passport group, as defined in file HGDS.cfg, to collect data from. Only one group can be specified.

`-u <Passport userID>`

is the user ID required to access the nodes in the group. All nodes in the group must use the same user ID.

```
-p <Passport userID password>
```

is the user ID required to access the nodes in the group. All nodes in the group must use the same user ID. If you provide a password in the command line, this password is treated as a plain text password.

```
passwordFile.pwd
```

is the file that contains an encrypted password. If you provide a file, the contents of the file must only be the encrypted password. For more information on passwords, see the section on setting server passwords in NN10600-605 *Passport - MDM Network Security: Operations*.

```
[-5port <5 minute TCP port>]
```

this optional parameter specifies the TCP port used for transmitting the 5-minute CSV data stream. The default port is 1646.

```
[-30port <30 minute TCP port>]
```

this optional parameter specifies the TCP port used for transmitting the 30-minute CSV data stream. The default port is 1647.

```
[-ppexpire <seconds>]
```

this optional parameter specifies the time available for data collection after the 5-minute interval has started. The minimum is 5 seconds, the maximum is 270 seconds, and the default is 15 seconds.

```
[-savefile -hgds|-shelf]
```

this optional parameter specifies that 5- and 30- minute streamed PM statistics are written to files in directory </opt/Magellan/data/pmsp/<Passport group name>/closedNotSent/> on the workstation.

> ***Note:*** If -savefile is specified, -hgds or -shelf must be specified.

There are two options for CSV file names. The file names of the CSV files saved on the workstation can be based on the group name from HGDS.cfg (-hgds). One file is saved for the group. Optionally, the file name can be based on the Passport Shelf Name from the PM statistic record (-shelf). In this case, a file is saved for each node in the group.

```
[-nosync]
```

this optional parameter specifies that PMSP will still process the record, even though its timestamp falls out of the 5-minute collection interval or the 30-minute aggregate interval.

```
[-h|-help]
```

this optional parameter provides a list of these options.

Error messages

The PMSP forwards its error and warning messages to the System Log Display tool in real time. The following tables provide a description of the errors generated by the PMSP.

- “Fatal error messages” (page 445)
- “Non-fatal error messages” (page 447)
- “Warning messages” (page 449)

See 241-6001-303 *Preside MDM Administrator Guide* for information on using the System Log Display tool.

Table 66
Fatal error messages

Error message	Description
`invalid command line argument: <arg>`	The command line passed to the PMSP is not recognized. The invalid argument is presented in place of <arg>. Verify the command line argument is applicable to the PMSP.
`missing mandatory argument: <what argument>`	Indicates that a required command line argument was missing. Verify that all mandatory arguments are used.
(Sheet 1 of 3)	

Table 66 (Continued)
Fatal error messages

Error message	Description
`error from FDTM: <error>`	Indicates that the PMSP received an error from Passport Communications Manager (FDTM) server. See 241-6001-303 *Preside MDM Administrator Guide* for information on using the <error> string to determine the problem.
`lost connection to: <service>`	The PMSP has lost a connection to the service indicated. Verify that the service is running. See 241-6001-303 *Preside MDM Administrator Guide* for details.
`could not connect to FDTR: (host <hostname>)`	The PMSP could not connect to FDTR. See 241-6001-303 *Preside MDM Administrator Guide* for details.
`could not create session with FDTR: <servicename>`	The PMSP cannot open a RTSTATS session with FDTR. See 241-6001-303 *Preside MDM Administrator Guide* for details.
`error response from: <service>`	The PMSP has encountered an error response from one of its required services. See 241-6001-303 *Preside MDM Administrator Guide* for details.
`unknown response from: <service>`	The PMSP has encountered an unknown response from one of its required services. See 241-6001-303 *Preside MDM Administrator Guide* for details.
`Failed to connect to HGDS`	The PMSP cannot connect to the Host Group Directory Service. Verify that the service is running. See 241-6001-303 *Preside MDM Administrator Guide* for details.
`out of memory: <for_what>`	The PMSP is unable to allocate critical memory. The memory is required for the component indicated in <for_what>. Verify that the machine has sufficient memory resources to run the PMSP.
`internal error: <where>`	An internal error has occurred. Verify that the executable is not corrupted.
`improper permissions on directory: <dir>`	Indicates that the PMSP cannot read and write to the directory specified. Change the permissions on the directory or run the PMSP under the proper user (for the directory).
(Sheet 2 of 3)	

Table 66 (Continued)
Fatal error messages

Error message	Description
`TCP error: <error>`	Indicates that the PMSP has failed to use a required TCP service. Verify that the service indicated in <error> is available to the PMSP.
`cannot create backup disk stream`	The PMSP cannot create the service to backup historical data. Check permissions on the pmsp directory, check for number of available file descriptors. Check for available memory.
`comm error with FDTM`	Communication error with FDTM.
`value is out of range <value>`	ppexpire value is out of range.
`cannot restart process with execve. Check -P command line option.`	Cannot re-execute program.
(Sheet 3 of 3)	

Table 67
Non-fatal error messages

Error message	Description
`connection to FDTM failed: retry in 2 minutes`	The PMSP cannot contact FDTM. FDTM is likely not running or not responding. It will retry in 2 minutes.
`TCP warning: <warning>`	Indicates that the PMSP cannot use a desired TCP service. The service is not essential to the operation of the PMSP. The details of the unavailable service is available in <warning>.
`cannot open aggregate autosave file: <filename>`	The PMSP cannot open the autosave file and therefore cannot read or write its contents. Verify the permissions on the file are correct.
`failure to write to aggregate autosave file: <reason>`	The PMSP cannot write to the autosave file. Check the available disk space.
(Sheet 1 of 2)	

Table 67 (Continued)
Non-fatal error messages

Error message	Description
`cannot open csv historical file: <filename>`	The PMSP cannot open the historical data file and therefore cannot read or write its contents. Verify the permissions on the file are correct.
`failure to write to csv historical file: <reason>`	The PMSP cannot write to the historical csv file. Check the available disk space.
`corrupt historical csv file: <reason>`	The PMSP cannot read from the historical csv file. Check the contents of the csv file.
`cannot access directory: <dir>`	The PMSP cannot read the directory specified in <dir>. Check the permissions on the directory.
`TCPOutputStream lost connection: <reason>`	The PMSP has lost connection or terminated connection with the client for the reason specified.
`failed to remove historical data file ... aborting historical data transfer`	This is related to the “Recovery from loss of connectivity.” After re-transmitting a historical file to SDM after reconnecting, it could not delete the file. Check permission.
`improper date/time on historical file: <filename>`	The date/time format of historical file is wrong. Check the file.
`cannot get file status: <filename>`	Cannot get file status of the file. Check the file.
`cannot read csv historical file: <filename>`	Cannot read csv historical file. Check file permission.
`Could not rename <filename1> to <filename2>`	Cannot rename file. Check the files.
(Sheet 2 of 2)	

Table 68
Warning messages

Error message	Description
`unrecognized record format (<what>)`	Unrecognized record format from the node.
`connection to FDTM failed .. retry in 1 minute (retry <number> of <number>).`	Re-try connection to FDTM.

Chapter 56
Real Time Alarm Collector (RTACCOL)

This section contains information on the Real Time Alarm Collector Server (RTACCOL). See the following topics for more information:

About the RTACCOL server

The RTACCOL server is the server responsible for collecting all alarms [DPN, Preside Multiservice Data Manager (MDM)], and other devices generating alarms) and storing them in files, one file per day. The RTACCOL server is started by the Server Administration tool.

The RTACCOL server reads the configuration file /opt/MagellanNMS/cfg/RTAC.cfg to determine the following:

- where to retrieve the Record Definition Files (RDFs)
- where to store the files
- which characters to use to separate fields and to replace new line characters

See “Configuration” (page 453) for a description of the RTACCOL server parameters.

The RTACCOL server collects and processes each alarm received from the GMDR server. It reads the configuration file /opt/MagellanNMS/cfg/RTAC.cfg to locate the RDF. Each alarm received is converted to the alarm syntax, as defined by the Record Definition File (RDF), ala.rdf in the /opt/MagellanNMS/data/rtac/rdf directory. See “RTAC_DATA_DIR” (page 453) for information on the directory path for the RDF.

Each alarm takes the following format:

```
<value of field 1><sep><value of field2><sep><value of
field n>
```

> ***Note:*** The order of the fields is defined by the RDF.

where:

<sep> is defined under RTAC_FIELD_DELIMITER.

See “RTAC_FIELD_DELIMITER” (page 453) for additional information.

The server reads the configuration file, /opt/MagellanNMS/cfg/RTAC.cfg, to determine where to store the alarm files. See “RTAC_DATA_DIR” (page 453) for information on the directory path where alarm fields are stored.

Each alarm received is saved in the file corresponding to the date of the alarm (alarms.<yyyy>-<mm>-<dd>) and stored in a directory as defined in the configuration file, RTAC.cfg.

You can retrieve the short-term historical alarms collected by the RTACCOL server by using the Query Historical Alarm tool available from the MDM toolset or by using the rtacsrch command line. For information about these methods, see 241-6001-011 *Preside MDM Fault Management User Guide*.

Configuration

The RTACCOL server configuration file, RTAC.cfg, contains the parameters for the RTACCOL server. The configuration file, RTAC.cfg, can be found in the /opt/MagellanNMS/cfg directory.

The following describes each RTACCOL server parameter contained in the configuration file /opt/MagellanNMS/cfg/RTAC.cfg.

RTAC_DATA_DIR

Directory containing the alarm files. This parameter defines the absolute directory path where the alarm files are stored. The default is /opt/MagellanNMS/data/rtac/data.

RTAC_RDF_DIR

Directory containing the Record Definition File (RDF) for the RTAC server. This parameter defines the absolute directory path where the RDF file for RTAC can be found. The default is /opt/MagellanNMS/data/rtac/rdf.

RTAC_FIELD_DELIMITER

This parameter defines the field delimiter used to separate each field of an alarm record. The default is the delete character.

RTAC_NEWLINE_DELIMITER

This parameter defines the character used to replace newline in multiple line fields. The default is | .

Managing the RTACCOL server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the RTACCOL server. Any changes that you make to the startup command or options take effect when you restart the RTACCOL server. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 454)
- “Startup command” (page 454)

Suggested name in Server Administration

The recommended name for the real time alarm collector server is Real Time Alarm Col.

Configuring the RTACCOL server with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start the RTACCOL server has the following syntax:

```
/opt/MagellanNMS/bin/rtaccol [-h]
[-filecleanup <number of days>]
```

Note: It is recommended that you specify the command option *-filecleanup <number of days>* to ensure that files with data that was modified <number of days> ago are deleted.

Variable definitions

Variable	Definition
`-h`	is the option to display information about the startup command.
`-filecleanup <number of days>`	files with data that was modified the specified number of days ago are deleted. The file system modification date is used to determine which files to remove. Obsolete files are removed once per day at midnight.

Deleting alarm records

In order to avoid disk space problems, it is recommended that the -filecleanup option be specified in the command line. Alarm files (alarms.<yyyy>-<mm>-<dd>) can also be deleted manually.

Interdependencies

The RTACCOL server relies on the GMDR server to send it all alarms.

Exit Codes

Exit codes for the RTACCOL server are shown in the following table.

Table 69
Exit codes for the RTACCOL server

Exit code	Description
0	Normal success (should never happen)
1	Major errors were found
2	Terminated because of signal received

Chapter 57
Restore Controller (NSCTLRST)

This section contains information on the Restore Controller (NSCTLRST). See the following topics for more information:

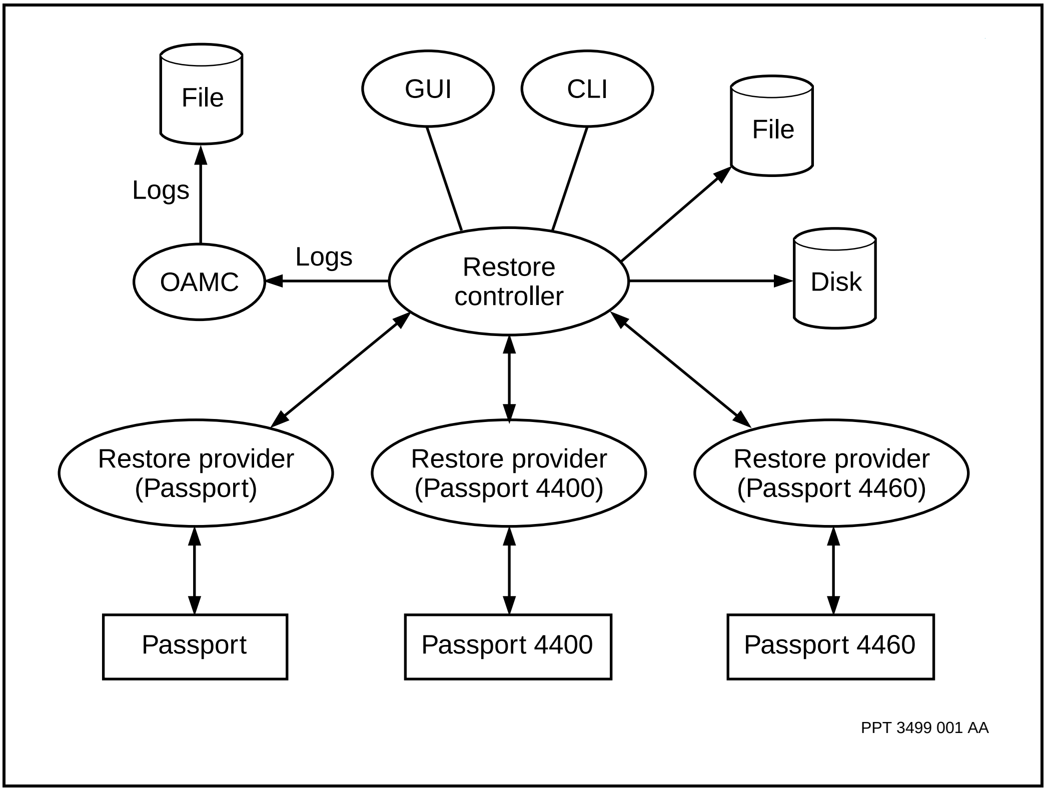
- “About the NSCTRLST” (page 457)
- “Managing the NSCTLRST” (page 459)

About the NSCTRLST

The Restore Controller and Restore Provider are responsible for restoring the backed up views to the node. See “Restore Controller data flow diagram” (page 458).

See the 241-6001-807 *Preside MDM Network Backup and Restore* for more information.

Figure 48
Restore Controller data flow diagram



Restore Controller logging

The OAMC server collects logs from the Restore Controller. These logs can also be sent directly to a log file. The Restore Controller logs are stored in /opt/MagellanNMS/data/log/bckRst/restore.dlog. Logging is automatically enabled as a default. The default log levels included are error, warning and notice.

Logging is automatically enabled as a default. The Restore Controller has the following logging levels: `error, warning, notice, info, debug, trace`. The default is `notice`. The initial logging level is set with the command line option for the Backup Controller: `-ll <level>`, for example, `nsctlbck -ll debug`. This starts the Backup Controller with the logging level of `debug` and also includes `trace`. Similarly, if the level `error` is set, it includes all the other logging levels. See “Backup Controller startup command” (page 57) for more information.

Managing the NSCTLRST

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the server. Any changes you make to the server startup command or options take effect when the server is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 459)
- “Startup command” (page 460)

Suggested name in Server Administration

The recommended name is Restore Controller.

Configuring the NSCTLRST with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The startup command for the NSCTLRST is as follows:

```
/opt/MagellanNMS/bin/nsctlrst [-p <port_no>]
  [-c <remote_mapping_file>]
```

Use the following table to substitute command parameters:

Parameter	Definition
port_no	is the port number the NSCTLRST uses. The port number is dynamically assigned but you can also specify it as port number 5001. To launch the Passport/SNMP Service Data Restore interface, you must specify port number 5001.
remote_mapping_file	is the name of the remote mapping file. The default remote mapping file is /opt/MagellanNMS/cfg/Controller.cfg.

Chapter 58
Security Audit Log Collector (SALC)

This section contains information on the Security Audit Log Collector (SALC). See the following topics for more information:

- “About the SALC server” (page 461)
- “Managing the SALC server” (page 468)
- “Configuring the Security Audit Log Collector (SALC) server” (page 470)
- “Configuring the SALC server to send real-time security logs to Syslog daemon” (page 474)

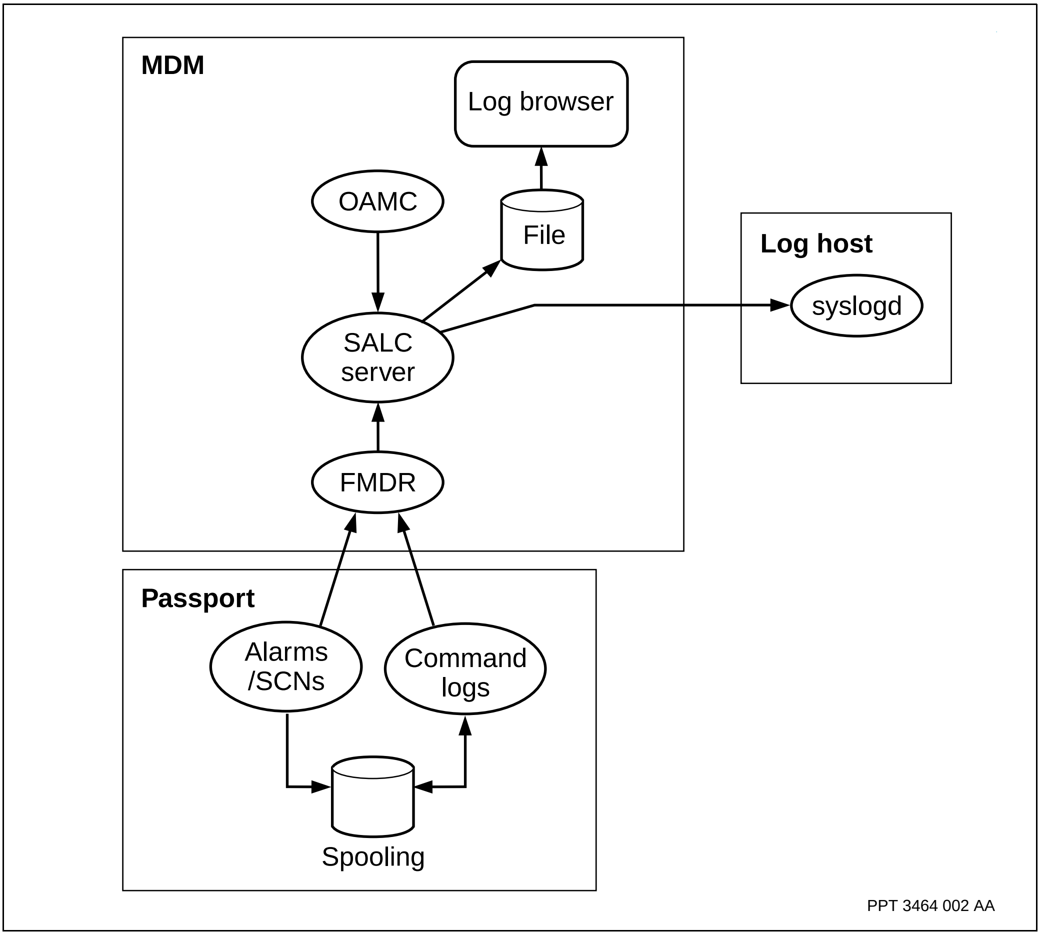
For detailed information on security audit logs, see the NN10600-605 *Passport - MDM Network Security: Operations*.

About the SALC server

The SALC server is the central collector of security logs and events (see “SALC data flow diagram” (page 462)). The SALC server has the following functions:

- collects alarms and command logs from the node
- receives security-related logs from the OAMC server
- sends logs to file
- directs output to either a local or remote log collection workstation running a Syslog service. The user is responsible for collecting, archiving, and managing security audit logs on the log collection workstation.

Figure 49
SALC data flow diagram

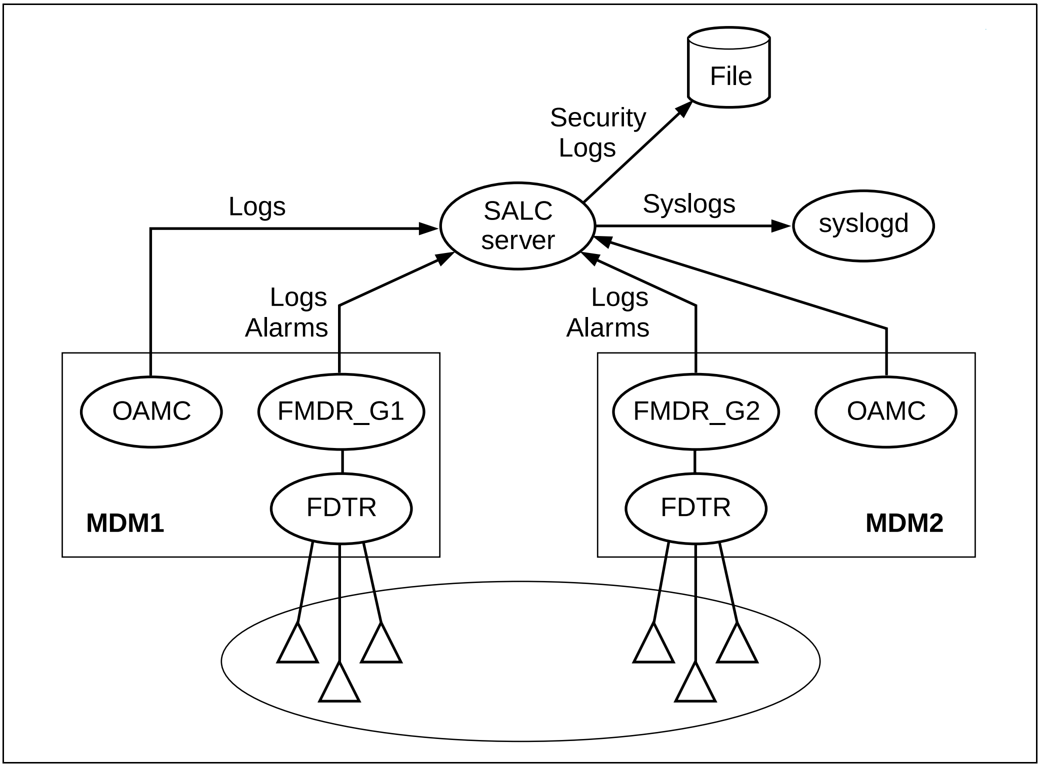


Real-time security logs

The SALC server process connects to FMDR to open command log and alarm streams from the node. It also connects with OAMC to collect security log information from Preside Multiservice Data Manager (MDM) servers (see “Real-time security log collection” (page 464)). Security logs are written to /opt/MagellanNMS/data/security/ security.nlog. Security logs are passed through the MDM logging system regardless of severity.

FDTR opens the command log stream on the node. FMDR allows for API sieves to be opened against the command logs and security alarms on the node and filters out security events. The log stream is opened the first time a client requests it, and remains open until FMDR is reset, regardless of whether the SALC server has been stopped.

Figure 50
Real-time security log collection

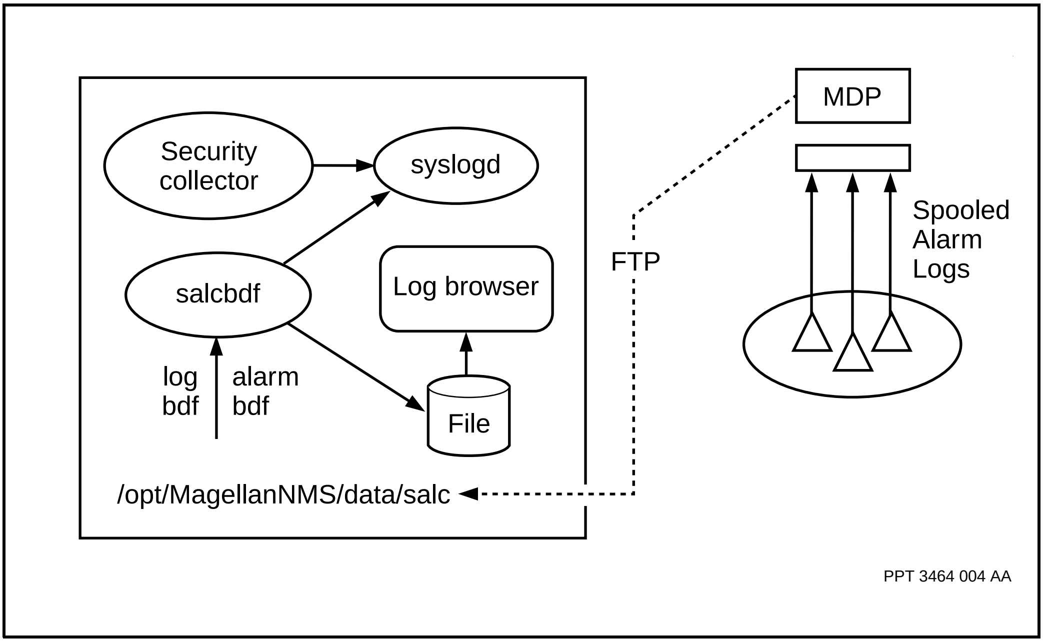


Non Real-time security logs

Security events for the network can be recovered by converting spooled log and alarm records using salcbdf. Salcbdf operates on bulk data format (BDF) records that are generated by MDP (see “Non-real time security audit log collection” (page 465)). For more information on BDF files, refer to 241-6001-309 *Preside MDM Management Data Provider User Guide*.

The salcbdf filters and transforms these records and sends them to either a local or remote syslog daemon in a similar form as that produced by the SALC server. See Security audit logs in the NN10600-605 *Passport - MDM Network Security: Operations* for configuration information.

Figure 51
Non-real time security audit log collection



The Security Audit Log Collector BDF converter is invoked on command line and has the following syntax:

```
/opt/MagellanMDP/bin/salcbdf
[-input <input directory>]
[-escape <escape character>]
[-start <start date>] (Format: YYYYMMMDDD [hhmmss])
[-end <end date> ](Format: YYYYMMMDDD [hhmmss])
[-facility local [0..7]]
[-outputSyslog [<hostname>]]
[-outputFile <filename>]
[-erase]
[-help]
```

Variable definitions

Variable	Value
- input <input directory>	is the directory where input alarm and log BDFs are stored. The default location is /opt/MagellanNMS/data/salc.
- escape <escape character>	is the escape character used in the BDF file. By default, the escape character is set to %.
-start <start date>	is a mask to delimit the start time. If the hhmmss are not provided, the default hhmmss is 000000. If this argument is not present, then no masking occurs based on start time.
-end <end date>	is a mask to delimit the end time. If the hhmmss are not provided, the default hhmmss is 235959. If this argument is not present, then no masking occurs based on end time.
facility local [0..7]	is used to set the facility value for bdf security logs directed to syslog. The default is local2.
-outputSyslog	is used to write bdf security logs to syslog daemon residing on <hostname>. If the host name is omitted, then the local host is the default location. By default, the logs are sent to the local syslog when neither -outputSyslog or -outputFile is used.
-outputFile <filename>	is the output file destination for the node security log records. If this option is used, the default salcbdf syslog stream is disabled unless the -outputSyslog option is also specified on the command line. If the <filename> is not specified, the output is standard out.
(Sheet 1 of 2)	

Variable	Value
-erase	is provided to allow BDF files that have been processed by salcbdf to be removed. By default, the BDF file is not removed.
-help	provides usage information.
(Sheet 2 of 2)	

Managing the SALC server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for SALC. Any changes you make to the startup command or options take effect when SALC is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 468)
- “Startup command” (page 468)

Suggested name in Server Administration

The recommended name for the SALC server is Security Audit Log Collector.

Adding the SALC server in the Server Administration tool requires you to select the server name in the **SVM New Server Selection** dialog. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted. See adding a new server in the 241-6001-303 *Preside MDM Administrator Guide* for more information.

Startup command

The startup command for the SALC server has the following syntax:

```
/opt/MagellanNMS/bin/salcserver
[-OAMCFacility local[0. .7]]
[-passportFacility local[0. .7]]
[-queue [0..50000]]
[-outputSyslog [<hostname>]]
[-outputFile]
[-logFile [<log level>]]
[-help]
```

Variable definitions

Variable	Value
-OAMCFacility local[0. .7]	is used to set the facility value for OAM security logs. The default is local0.
-passportFacility local [0. .7]	is used to set the facility value for node security logs. The default is local1. OAMC logs should already contain the facility value for the Preside Multiservice Data Manager audit logs.
-queue[0..50000]	is used to perform duplicate security log checking. Logs that match existing entries in the queue will not be processed again. The queue size for checking can be set between 0 and 50,000. Larger queue sizes may impact performance. The default queue size value is 5000.
-outputSyslog [<hostname>]	is used to write security logs to the syslog daemon residing on the workstation <hostname>. If <hostname> is not specified, then the localhost is the default location.
-outputFile	is used to write security logs to /opt/MagellanNMS/data/ security/security.nlog as a daily rolling log that is readable by all, but writable only by the owner of the SALC process, which is usually the root user.
-logFile [<log level set>]	is used to write logs belonging to the <log level set>. Logs are written to /opt/MagellanNMS/data/log/salcserver/ salcserver.alog as a daily rolling log that is readable by all, but writable only by the owner of the SALC process, which is usually the root user. The log level can be set to one or more of the available values [DEBUG, INFO, NOTICE, CLEARED, WARN, ERROR, CRIT, ALERT, FATAL]. If the log level set is not specified, then the default log level is set to [FATAL, ALERT, CRIT, CLEARED, NOTICE].
-help	provides usage information.

Configuring the Security Audit Log Collector (SALC) server

Configure the SALC server to collect security logs and events from Preside Multiservice Data Manager (MDM) and the network in order to monitor these logs for critical events that can lead to security breaches.

Prerequisites

- The Security Audit Log Collector (SALC) server has been added to the list of servers in Server Administration. See Adding a new server in the 241-6001-303 *Preside MDM Administrator Guide*.
- FMDR groups must be defined for the nodes in the network. See the 241-6001-303 *Preside MDM Administrator Guide*.
- The FMDR server needs to be started with a command line option specifying a userID with command impact of systemAdministration and a scope of device or higher and a customer ID of 0

Procedure steps

1 Create the SALC server configuration file:

```
vi /opt/MagellanNMS/cfg/SALCServer.cfg
```

2 Specify the list of MDM host names, OAMC, and FMDR servers to connect with in the file /opt/MagellanNMS/cfg/SALCserver.cfg using the following format:

```
Hostname: mdmhost

Servername: FMDR_GROUP1

UserID: mdmuser

Password: passwd1


Hostname: mdmhost

Servername: OAMC
```

Note: Use blank lines to separate records.

Variable definitions

Variable	Value
Hostname	is the MDM host name.
Servername	is the MDM server name. Possible values are OAMC and FMDR.
UserId	is the node user ID. At a minimum, the user ID must have systemAdministration impact and a scope of device or higher and a customer ID of 0. UserId and Password fields are not required for OAMC.
Password	is a clear-text password that corresponds to the user id.
EncryptedPassword	A Password line without an EncryptedPassword line is left unchanged by SALC. If the EncryptedPassword line is present, but empty, then SALC encrypts the value in the Password line and blanks it out. Then, it fills the EncryptedPassword value accordingly. If the customer wishes to change the Password value, they edit the configuration file by adding un-encrypted password in place of the blank Password field value. When the configuration file is refreshed by SALC, the Password is encrypted, blanked out, and the Encrypted Password field value is replaced accordingly.

Example configuration

The following is an example of a configuration file prior to running the SALC server:

```
Hostname: MDM1

Servername: FMDR_G1

UserID: userid

Password: password1


Hostname: MDM2

Servername: FMDR_G2

UserID: userid2

Password: password2

EncryptedPassword:


Hostname: MDM1

Servername: OAMC


Hostname: MDM2

Servername: OAMC
```

This is the same configuration file after running the SALC server: the FMDR_G1 password remains unchanged as no ‘EncryptedPassword:’ line was specified. The FMDR_G2 password is now encrypted on the “EncryptedPassword:’ line and cleared from the ‘Password:’ line.

```
Hostname: MDM1

Servername: FMDR_G1

UserID: userid

Password: password1


Hostname: MDM2

Servername: FMDR_G2

UserID: userid2

Password:

EncryptedPassword: ncrptdpwd2


Hostname: MDM1

Servername: OAMC


Hostname: MDM2

Servername: OAMC
```

Configuring the SALC server to send real-time security logs to Syslog daemon

Configure the SALC server to send logs to a Syslog daemon on a local or remote host for customers who use a centralized log collection system based on syslog.

Prerequisites

- Be familiar with the SUN documentation or manpages for syslogd and syslog.conf.

Procedure steps

1. Input the following command on the command line:

```
/opt/MagellanNMS/bin/salcserver -outputSyslog
<hostname>
```

Variable definitions

Variable	Value
hostname	is the host name of the workstation where the syslog daemon resides.

Chapter 59
SNMP Data Collection Daemon (GENDCD)

This section contains information on the SMDR-based data collection daemon (DCD). See the following topics for more information:

> ***Note:*** For more information on the generic DCD and the SNMP Surveillance Adapter process, see 241-6001-118 *Preside MDM SNMP Surveillance Adapter Guide*.

About the DCD server

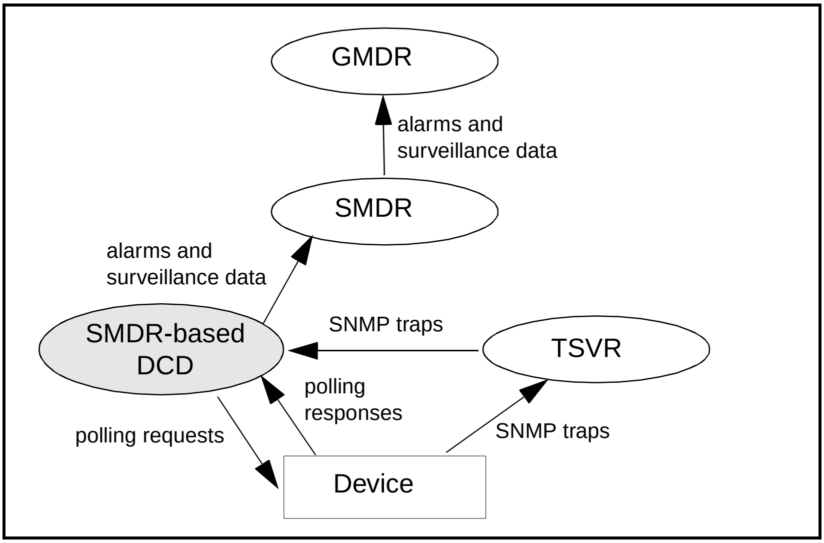
The DCD server is a process whose purpose is to collect surveillance data from selected devices.

The DCD performs the following functions:

- polls the devices using the SNMP protocol
- notifies SMDR when there is a new component, when a component is deleted, and when there is a change in state of a component

- converts traps received from TSVR to alarms and forwards the alarms to SMDR
- maintains a device seed file containing a list of devices, with their IP addresses and community strings, that is used to rediscover these devices upon restart

Figure 52
DCD data flow diagram



Managing the DCD server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the DCD. Any changes you make to the startup command or options take effect when the DCD is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested names in Server Administration” (page 477)
- “Startup command” (page 477)

Suggested names in Server Administration

Configuring the DCD with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

The three types of names for the DCD are as follows:

- The DCD server name refers to the name of the DCD server entered through the Server Administration tool.
- The DCD executable name is specified in the startup command and is device-specific. The executable name is represented by the parameter <dcd name>.
- The DCD process name is the DCD executable name with an optional name extension appended to it. This name extension (-n) differentiates two or more DCD servers, used by SMDR, of the same type that are running on the same workstation. The DCD process name is represented by the parameter <process name> which is <dcd name>_ [<name extension>].

Example

If the command is the following:

```
/opt/MagellanNMS/bin/gendcd -n reg1
```

Then the <process name> is:

```
gendcd_reg1
```

Startup command

The command to start the DCD has the following syntax:

```
/opt/MagellanNMS/bin/<dcd name> [-N <name prefix>]
[-n <name extension>] [-b <congestion buffer size>]
[-v] [-d <log levels list>]
```

where:

`<dcd name>` is the executable name of the DCD. This name is based on the device being supported, as follows:

- `gendcd` for the generic DCD

`-N <name prefix>` specifies a name modifier to replace the `gen` substring for the `gendcd` at the beginning of the executable name to form the process name. This modification also applies to the log file, seed file, and configuration file. For example, if you specify **-N devx**, the process name becomes `devxdcd`. The default names of the files become devxdcd.log, devxdcd.sed, and devxdcd.cfg respectively.

`-n <name extension>` specifies a name extension to append to the DCD executable name. This extension also applies to the log file, seed file, configuration file, and process name. Use this parameter when two or more DCD servers of the same type are running on the same workstation.

`-b <congestion buffer size>` specifies the maximum number of congested replies before the client is cut off. This option protects against slow, disconnected, or non-communicating clients, such as when a client has disconnected or cannot communicate with the DCD. If this option is not specified, the default value of 5,000 is used. The default value is adequate for most installations.

`-v` causes the DCD to run in file verification mode. In this mode, the DCD reads the files, issues associated error logs if there are any, and stops with a final log.

> ***Note:*** You can also use the -i parameter with the -v parameter. The -i parameter causes the system to display logs in the UNIX window instead of sending them to a log file.

`-d <log levels list>` defines DCD log levels.

Interdependencies

The DCD relies on the trap server daemon (TSVR) and the SNMP Management Data Router (SMDR).

Configuration

There are no mandatory configuration files to manually set up for device-specific DCDs. The configuration files for the device-specific DCDs are already set up. There are two configuration files that you can edit, the run-time options file and the seed file.

See also...

- “Run-time options file” (page 479)
- “Seed file” (page 482)

Run-time options file

You can use this file to override the default values of some configuration parameters. Its location is /opt/MagellanNMS/cfg/dcd/<process name>.cfg.

where:

`<process name>` is the DCD executable name with a name extension if there is one. In the following example, the process name is **gendcd_reg1**:

```
/opt/MagellanNMS/cfg/dcd/gendcd_reg1.cfg
```

You can override some configuration parameters by specifying them as options in either the run-time options file or in the DCD startup command. If you specify the same configuration parameter in both places, the value in the run-time options file is used.

There is one line in the run-time options file for each configuration parameter that you override. The format of each line is

```
<keyword>:<value>
```

where:

`keyword` is the configuration parameter

`value` is the value you assign to the configuration parameter

The following list shows the configuration parameters that you can edit in the run-time options file.

> ***Note:*** The defaults in the following list apply to the DCD library and may not be the same as the defaults selected for a device-specific DCD. For more run-time options that apply to the generic DCD, see 241-6001-118 *Preside MDM SNMP Surveillance Adapter Guide*.

- addrFilter, which is a character string that you use to filter the IP addresses of devices monitored by this DCD. The DCD rejects any trap or request that does not pass this filter. Address filters are used to split the network devices between several DCD processes. The size of your network can be a reason to update addrFilter. You can have multiple occurrences of addrFilter if you want to cover more than one subnetwork.

  The format of addrFilter is

  ```
  <IPelement>[.<IPelement>[.<IPelement>[.<IPelement>]]]
  ```

  where:

  `IPelement` represents one of the four positions of an IP address and can be in one of the following formats:

  - — * accepts all values between 0 and 255
  - — *integer* accepts only this value, which must be between 0 and 255
  - — *integer-integer* accepts only this range, which must be between 0 and 255

The default value of this parameter is to accept all IP addresses.

- autoObjDel, which is a flag that indicates whether devices are automatically deleted if they fail to respond to polls within the time interval defined by the parameter objDelInt. Its default value is TRUE.
- cfgPollInt, which represents the time, in seconds, between consecutive configuration polls of a device. A configuration poll is the polling process that discovers which subcomponents are part of the device and their configuration attributes. The default value of this parameter is 43200 (12 hours).
- congLevel, which specifies the maximum number of congested replies before the client (SMDR) is cut off. This parameter protects against slow, disconnected, or non-communicating clients, such as when a client has disconnected or cannot communicate with the DCD. Its default value is 5000.

- maxStPollInt, which represents the normal time interval, in seconds, between consecutive state polls of a device. A state poll is the polling process that verifies the state of already discovered subcomponents.The default value of this parameter is 300 (5 minutes).
- minStPollInt, which represents the minimum time interval, in seconds, between consecutive state polls of a device. Although several events, such as traps, can cause state polls to occur more frequently than the interval specified in the parameter maxStPollInt, state polls are not allowed to occur more frequently than the value of this parameter. Its default value is 30.
- objDelInt, which represents a time interval, in seconds. If the parameter autoObjDel is set to TRUE and a device does not respond to polling in the time interval defined by objDelInt, the device is deleted. The default value of this parameter is 259200 (3 days).
- reachPollInt, which represents the time, in seconds, between consecutive reachability polls. A reachability poll is the polling process that verifies whether the device is still reachable. The default value of this parameter is 30.
- seedFileInt, which represents the time, in seconds, between consecutive seed file refreshes. Its default value is 7200 (2 hours).
- snmpRespInt, which represents the waiting time, in seconds, for a response to an SNMP request. Its default value is 10.

  Note: The value of this parameter must be high enough to prevent SNMP response timeout in normal operating conditions. However, the value must not be excessively high because it also determines the length of time to detect device unreachability (this value times the number of request retries plus 1).

- snmpRetrCnt, which specifies the maximum number of retries for an SNMP request. Its default value is 2.
- trapDisabled is an option that is set by default to disable trap discovery when the DCD process receives a trap from a device. If a device can use different addresses in the traps it sends, trap discovery may attempt to poll for different devices that do not exist. To enable trap discovery, specify FALSE with this parameter. To disable trap discovery, specify TRUE with this parameter.

- useAgentAddr, which is a an option that indicates whether or not the DCD uses the IP address contained in the trap as the address of the device. By default, the IP address associated with the trap is the address associated with the sender device. However, the trap agent address can be set to a value other than the sender’s address. If you specify TRUE with this parameter, the DCD will use the IP address contained in the trap instead of the sender’s address as the address of the device.

Seed file

> ***Note:*** Editing the seed file is not recommended. Instead you can use the dcdAddNode script or IP Discovery to discover new devices.

This configuration file lists the connection information for each device that has been discovered. Its location is /opt/MagellanNMS/cfg/<process name>.sed.

`<process name>` is the DCD executable name with a name extension if there is one. In the following example, the process name is **gendcd_reg1**:

```
/opt/MagellanNMS/cfg/gendcd_reg1.sed
```

This file is optional and if it doesn’t exist, the DCD builds one. Using the configuration parameter seedFileInt, the DCD periodically refreshes this file with a list of the devices that are currently discovered. The DCD reads the seed file upon restart and tries to discover the devices listed in the file.

There is one line for each device in the seed file. The format of each line is

```
<device name>:<IP address>::<port number>:<device type>
<encrypted community string>
```

Example

MPA BRANCH1:45.236.4.120::161:5:ssYgjL7wj 71sj

You can discover a new device without having to wait for a trap from the new device by using the dcdAddNode script. See “dcdAddNode” (page 483) for more information.

Action scripts

Use these scripts when you need to issue action requests to the DCD:

- “dcdAddNode” (page 483)
- “dcdTriggerPoll” (page 484)

For other action scripts related to generic DCD and adding multiple addresses, see the following topics in the 241-6001-118 *Preside MDM SNMP Surveillance Adapter Guide*:

- dcdAddAddress
- dcdDeleteAddress

dcdAddNode

The dcdAddNode script issues a request to connect a device to an appropriate DCD and for the DCD to monitor the device. The syntax of this script is

```
/opt/MagellanNMS/bin/dcdAddNode <nodeType> <devName>
<addr> <community> <port> <devType>
```

where:

`nodeType` is the category name for the device.

`devName` is the name of the device.

`addr` is the IP address of the device.

`community` is the community string of the device.

`port` is the port number of the device SNMP agent.

`devType` is the value of the device type. This value must match the value in the corresponding agent profile configuration file for devices supported by the SNMP Surveillance Adapter. Device type values defined by customers must be within the range of 900 - 999. Values equal to or below 899 are reserved for Nortel Networks development. The following list shows values for specific devices:

Example

```
dcdAddNode MPA NEWDEV 48.222.5.111 public 161 5
```

dcdTriggerPoll

The dcdTriggerPoll script issues a request to the DCD to rediscover a device. The syntax of this script is

```
/opt/MagellanNMS/bin/dcdTriggerPoll <nodeType>
<devName> [<delay time> [<group number> [<component
type> [<distance number>]]]]
```

where:

`nodeType` is the category name for the device.

`devName` is the device name.

`delay time` is an integer expression that specifies the delay, in seconds, before the request is scheduled. If the value of this parameter is 0, the request is scheduled with the current time and is added to the ready request queue. If the value of this parameter is greater than 0, the current time is increased by the specified value and the request is added to the ready request queue. If this parameter is not specified, none of the following parameters can be specified.

`group number` is the identifier of the polling request group that must be scheduled on demand. The only group number that cannot be specified is the trap polling group.If this parameter is not specified, neither of the following parameters can be specified.

`component type` is an integer expression that identifies the request that must be scheduled on demand. When you specify this parameter, only the request group with the matching attribute is scheduled. If this parameter is not specified, the following parameter cannot be specified.

`instance number` is a string expression that specifies the instance to be added to each polled variable table OID to poll a single row. This value must be entered in OID format unless the table has only one integer in the index. If this parameter is not specified, the entire table is polled.

Exit codes

Exit codes for when the DCD process terminates under its own control are shown in the following table. The DCD can terminate with other exit codes if it crashes or is stopped by the Server Administration tool.

Table 70
Exit codes for the DCD

Exit code	Description
4	The -v option was used and the process was terminated after verification of configuration files completed.
5	One of the mandatory configuration files cannot be processed, or some mandatory configuration parameters are undefined.
6	Registration with MNSD failed.
7	Software error
8	Memory allocation failed.

Error messages

This section contains error messages in the DCD library. All of these error messages apply to the generic DCD.

Error messages for the DCD fall into the following general categories:

- FATAL, for errors that are fatal to the operation of the DCD. This category of message always causes the DCD to terminate. It is generally caused by a faulty workstation setup or a bad process configuration. See the table “FATAL messages for the DCD” (page 486).
- SNO (should not occur), when the DCD experiences an error situation that should not occur (probably a software error). The error is logged and the DCD tries to continue executing. If an error of this type occurs repeatedly, report it to your Nortel Networks representative. See the table “SNO (should not occur) messages for the DCD” (page 487).

- ERROR, when the DCD experiences an error situation that is probably caused by an invalid configuration or switch input. The cause needs to be identified and corrected. See the table “ERROR messages for the DCD” (page 489).
- MAJOR, when the DCD signals the occurrence of an important event (usually not an error), such as the establishment or the loss of the connection with a server. See the table “MAJOR messages for the DCD” (page 503).
- MINOR, when the DCD signals the occurrence of a minor event which is usually not an error. The DCD does not issue these logs by default.
- INFO, when the DCD issues logs that contain processing information usually used for debugging. The DCD does not issue these logs by default.
- TRACE, when the DCD issues logs that contain information used for tracing the sequence of procedure entries and exits. The DCD does not issue these logs by default.

Table 71
FATAL messages for the DCD

Message	Meaning and action
Cannot read API dictionary file: <filename>	The DCD.dict file cannot be found/opened; it was not installed properly. Report the problem to your Nortel Networks representative.
Invalid address filter	The address filter specified in the run-time options file is invalid. Fix the addrFilter specification in the file.
IPC initialization failed	MNSD is not running or else it rejected the DCD registration; Preside Multiservice Data Manager (MDM) is not properly installed. Fix MDM installation.
(Sheet 1 of 2)	

Table 71 (Continued)
FATAL messages for the DCD

Message	Meaning and action
No DcdBuffer available	No DCD buffer was available from the DCD free buffer list. This should only happen for a trap that has an unusually high number of variables. Disable translation of this trap as a temporary work-around. Report the problem to your Nortel Networks representative.
No valid agent profile	No valid device profile was found. Verify validity of configuration files.
out_of_memory	DCD is not able to allocate memory for some data structure. Review the workstation engineering.
Undefined trap filter	The device-specific trap filter is not defined. This is a software error. Verify the oid filter definitions in the configuration files.
(Sheet 2 of 2)	

Table 72
SNO (should not occur) messages for the DCD

Message	Meaning and action
Add address failed. unknown address type	A software error occurred in the process of adding a DCD address. If this message occurs repeatedly, report it to your Nortel Networks representative.
Bad argument count	There is a software error in the DCD software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Bad command type: <command type>	There is a software error in the SNMP response handling software. If this message occurs repeatedly, report it to your Nortel Networks representative.
(Sheet 1 of 3)	

Table 72 (Continued)
SNO (should not occur) messages for the DCD

Message	Meaning and action
Cannot bind trap variable <variable index>	There is a software error in the proxy trap creation software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Cannot compute instance	A polled variable is missing in an SNMP response. Report the problem to your Nortel Networks representative.
Cannot create proxy trap for agent <device name>	There is a software error in the proxy trap creation software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Missing response variable for object <object name>	A polled variable is missing in an SNMP response. Report the problem to your Nortel Networks representative.
NULL trap pointer received	There is a software error in the proxy trap reception software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Packet decoding failed for agent <device name>	There is a software error in the process communication software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Packet decoding failed for trap, err=<error code>	There is a software error in the trap reception software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Recvfrom failed for agent <device name>	There is a software error in the response reception software. If this message occurs repeatedly, report it to your Nortel Networks representative.
Remove address failed. Unknown address type	A software error occurred in the process of removing a DCD address. If this message occurs repeatedly, report it to your Nortel Networks representative.
(Sheet 2 of 3)	

Table 72 (Continued)
SNO (should not occur) messages for the DCD

Message	Meaning and action
Undefined SNMP Probe Method	The polling configuration file contains an invalid line. Fix the file.
Undefined translation function used	The trap translation configuration file contains an invalid line. Fix the file.
Unknown address type	A software error occurred in the DCD address process. If this message occurs repeatedly, report it to your Nortel Networks representative.
Wrong command line parser	There is an internal software problem. Report the problem to your Nortel Networks representative.
The creation of the internal trap for agent failed.	The internal trap for the agent cannot be created when using the Accumulate rule. No action required. You cannot recreate the internal trap.
The binding of the SNMPv2 <nth> variable failed.	The Accumulate rule cannot bind the specified SNMPV2 variable. No action required.
The binding of the SNMPv1 <nth> variable failed.	The Accumulate rule cannot bind the specified SNMPV1 variable. No action required.
(Sheet 3 of 3)	

Table 73
ERROR messages for the DCD

Message	Meaning and action
Add address failed. Multiple address is disallowed	Multiple addresses are not allowed for the device type.
Add address failed. The address is in use	The address is currently an active polling address for a device while an attempt is being made to reassign it to another device.
Address <addr> rejected by address filter, line <lineNo>	The given address <addr> failed the address filter at line number <lineNo>. Fix the configuration file.
(Sheet 1 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Argument of VAROID is not a variable name	VAROID operator argument is not a variable name. Fix the configuration file.
Argument of VARTYPE is not a variable name.	VARTYPE operator argument is not a variable name. Fix the configuration file.
Bad address type: <addrtype>, line <lineNo>	The address type is illegal at line number <lineNo>. Fix the configuration file.
Bad command label (<label name>), line <line number>	This command type is not defined. Fix the configuration file.
Bad component id: <id name>	The argument of a compId command is not a valid component identifier.Fix the configuration file.
Bad component state: <state name>	The argument of a state command is not a valid state. Fix the configuration file.
Bad declaration (<decl label>) line <line number>	Invalid record type. Fix the configuration file.
Bad log level: <level name>	The first argument of a log command is not a valid log level. Default log level INFO was used. Fix the configuration file.
Bad numerical value	Invalid integer string found. Fix the configuration file.
Bad numerical value <integer string>	Invalid integer string found. Fix the configuration file.
Bad pattern: error: <error number>	Operator MATCH given an invalid pattern. Fix the configuration file.
Bad selector, line <line number>	Invalid selector value in conditional trap translation rule. Fix the configuration file.
Bad timestamp: <timestamp string>	Operator TIMESTAMP given a timestamp argument that does not match the format specified. Fix the configuration file.
(Sheet 2 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Bad timestamp format: <format string>	Operator TIMESTAMP given an invalid format selector. Fix the configuration file.
Bad translation command, line <line number>	Invalid trap translation command. Fix the configuration file.
Bad translation expression type, line <line number>	Trap translation command contains invalid expression. Fix the configuration file.
Bad variable name (<name string>) on line <line number>	Variable name contains invalid character. Fix the configuration file.
Bad variable number or OID in Name command	NAME command variable selector is not a number or an OID. Fix the configuration file.
Brace index out-of-range	MATCH operator given a brace index higher than the number of braces in the pattern. Fix the configuration file.
Cannot bind socket address for agent <device name>	A socket cannot be allocated for polling a newly discovered device. Review the workstation engineering.
Cannot evaluate code in Alarm command, line:<line number>	Evaluation of expression producing trap code in Alarm command failed. Fix the configuration file.
Cannot evaluate condition in Alarm command, line:<line number>	Evaluation of condition expression in Alarm command failed. Fix the configuration file.
Cannot evaluate condition in Discover command, line:<line number>	Evaluation of condition expression in Discover command failed. Fix the configuration file.
(Sheet 3 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Cannot evaluate condition in Next command, line:<line number>	Evaluation of condition expression in Next command failed. Fix the configuration file.
Cannot evaluate condition in Poll command, line:<line number>	Evaluation of condition expression in Poll command failed. Fix the configuration file.
Cannot evaluate condition in Quit command, line:<line number>	Evaluation of condition expression in Quit command failed. Fix the configuration file.
Cannot evaluate condition in UseBlock command, line:<line number>	Evaluation of condition expression in UseBlock command failed. Fix the configuration file.
Cannot evaluate content in Log command, line:<line number>	Evaluation of content expression in Log command failed. Fix the configuration file.
Cannot evaluate expression in CompId command, line:<line number>	Evaluation of identifier expression in CompId failed. Fix the configuration file.
Cannot evaluate expression in State command, line:<line number>	Evaluation of state expression in State command failed. Fix the configuration file.
Cannot evaluate level in Log command, line:<line number>	Evaluation of level expression in Log command failed. Fix the configuration file.
Cannot evaluate name in Cache command, line:<line number>	Evaluation of name expression in Cache command failed. Fix the configuration file.
(Sheet 4 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Cannot evaluate name in Property command, line:<line number>	Evaluation of name expression in Property command failed. Fix the configuration file.
Cannot evaluate value in Assign command, line:<line number>	Evaluation of value expression in Assign command failed. Fix the configuration file.
Cannot evaluate value in Cache command, line:<line number>	Evaluation of value expression in Cache command failed. Fix the configuration file.
Cannot evaluate value in Property command, line:<line number>	Evaluation of value expression in Property command failed. Fix the configuration file.
Cannot evaluate value in trap translation command, line:<line number>	Evaluation of value expression in trap translation rule command failed. Fix the configuration file.
Cannot evaluate variable in Alarm command, line:<line number>	Evaluation of one of the value expressions in Alarm command failed. Fix the configuration file.
Cannot find variable in Name command, line: <line number>	Variable corresponding to Name command variable selector does not exist. Fix the configuration file.
Cannot get socket for agent <device name>	A socket cannot be allocated for polling a newly discovered device. Review the workstation engineering.
cannot open <file name>	The file cannot be found or opened because it was not installed correctly. Report the problem to your Nortel Networks representative.
Cannot open seed file <filename> for refreshing	The DCD cannot open the seed file to rewrite its contents. Verify the file and subdirectories permissions.
(Sheet 5 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Cannot open trap translation cfg file <file name>	The specified file cannot be found or opened because it was not installed correctly. Report the problem to your Nortel Networks representative.
Component id not defined	The component identifier must have been defined by a CompId command before a State or Property command can be used. Fix the configuration file.
Delimiters string is empty	The TOKEN operator is given an empty delimiters argument. Fix the configuration file.
Duplicate class name <class name>, line <line number>	The polling configuration file contains a duplicate CLASS declaration. Fix the configuration file.
Duplicate object name <object name>, line <line number>	The polling configuration file contains a duplicate polling variable declaration. Fix the configuration file.
Duplicate type name <type name>, line <line number>	The polling configuration file contains a duplicate TYPE declaration. Fix the configuration file.
Error reading <file name>, line <line number>	Error when reading <filename> at line <line number>. The file may be corrupted. Report the problem to your Nortel Networks representative.
Error reading option file: <filename>	The contents of the run-time options file are corrupted. Restore the file contents.
Error reading seed file: <filename>	The contents of the seed file are corrupted. Restore the file contents.
Error response from sendRequestPDU for <device name>	An SNMP request cannot be sent to the device; probably due to transient workstation communication problems. If this message occurs repeatedly, review the workstation engineering.
Error response from SNMP_Bind_Null	This is an internal software error. Report it to your Nortel Networkss representative.
(Sheet 6 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Error return from SNMP_Create_Request	This is an internal software error. Report it to your Nortel Networks representative.
Error return from SNMP_Encode_Packet	This is an internal software error. Report it to your Nortel Networks representative.
Error writing to seed file <filename>	The DCD cannot write to the seed file. Verify the disk space and file permissions.
Failed to evaluate operator argument	Evaluation of an argument expression in an operator call failed. Fix the configuration file.
Failure to send SNMP request, Agent <device>, Error Number <error code>	An SNMP request cannot be sent to the device; probably due to transient workstation communication problems. If this message occurs repeatedly, review the workstation engineering.
Handling command failed	Evaluation of an SNMP response handling command failed. Fix the configuration file.
Ignored input on line <line number>	A command line contains additional text after the required parameters. This additional text is ignored.
Incomplete conditional expression, line <line number>	A conditional expression is not terminated by the required closing square bracket. Fix the configuration file.
Incomplete decimal constant, line <line number>	A - (minus sign) is not followed by decimal digits. Fix the configuration file.
Incomplete string constant, line <line number>	A quoted string is not terminated by the required closing quote. Fix the configuration file.
Invalid argument delimiter, line <line number>	Operator arguments are not separated by a comma (,). Fix the configuration file.
(Sheet 7 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Invalid argument number in call to <operation name>, line <line number>	The required number of arguments are not supplied in an operator call. Fix the configuration file.
Invalid command line parameter: <char>	There is an invalid option in the command line. Fix the command line.
Invalid conditional expression, line <line number>	There is a syntax error at line <line number>. Report the problem to your Nortel Networks representative.
Invalid IP address: <addr>	An invalid IP address was given. Check the address and correct it.
Invalid IP address: <addr>, line <lineNo>	An invalid IP address was given at line number <lineNo>. Fix the configuration file.
Invalid line (<line number>) in <file name>	There is a syntax error at line <line number> in file <file name>. Fix the configuration file.
Invalid line (<line number>) in run-time cfg	There is an invalid line in the run-time configuration file. Fix the file.
Invalid line type in trap translation data file: line <line number>	The trap translation file contains an invalid line. Fix the configuration file.
Invalid line (<line number>) in TYPE declaration	The polling configuration file contains an invalid line. Fix the configuration file.
Invalid METHOD (<method label>), line <line number>	The polling configuration file contains a CLASS declaration with an invalid METHOD specification. Fix the configuration file.
Invalid OID in trap translation data file, line <line number>	The trap translation configuration file contains a translation record with an invalid OID. Fix the configuration file.
Invalid OID <oid string>, line <line number>	The polling configuration file contains a CLASS declaration with an invalid variable OID. Fix the configuration file.
(Sheet 8 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Invalid option <option label> on line <line number>	There is an invalid option in the .cfg or .agp configuration file. Fix the configuration file.
Invalid symbol in expression <a symbol>, line <line number>	Response handling expression syntax error at line <line number>. Fix the configuration file.
Invalid syntax in trap translation data file: line <line number>	There is a missing label or colon (:) in front of a trap translation command. Fix the configuration file.
Left parenthesis missing in operator call, line <line number>	There is a syntax error at line <line number>. Fix the configuration file.
Logical line starting at line <line number> too long.	The line at <line number>, including the continuation lines, exceeds the maximum length. Fix the configuration file.
Missing class name, line <line number>	A CLASSNAME declaration is not followed by a polling class name. Fix the configuration file.
Missing community string in seed file line: <line number>	The seed file entry is missing the community string. Fix the configuration file.
Missing device type in seed file line: <line number>	The seed file entry is missing the device type. Fix the configuration file.
Missing elements in CLASS declaration <line number>	A mandatory declaration is missing from a polling class declaration. Fix the configuration file.
Missing elements in TYPE declaration <line number>	A mandatory declaration is missing in a polling type declaration. FIx the configuration file.
Missing left brace, line <line number>	An opening left brace is missing from a polling configuration file declaration. Fix the configuration file.
(Sheet 9 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Missing mandatory option in file <file name>	An agent profile configuration file is missing from some mandatory options. Fix the configuration file.
Missing node address in seed file line: <line number>	The seed file contains an invalid line. Fix the file.
Missing node name in seed file line: <line number>	The seed file contains an invalid line. Fix the file.
Missing remote port in seed file line: <line number>	The seed file contains an invalid line. Fix the file.
Missing right brace, line <line number>	The last record in a polling configuration file is not terminated by a right brace. Fix the configuration file.
Missing type name, line <line number>	The polling configuration file contains a TYPE declaration that is not followed by a type name. Fix the configuration file.
Multiple polling address is disallowed in seed file line: <lineNo>	The seed file contains multiple entries with the same node name but different addresses, while multiple polling address is not allowed for this type of device. Fix the seed file.
Negative PDU string variable length	This is an internal software error. Report it to your Nortel Networks representative.
Node address rejected by filter in seed file line: <line number>	A line in the seed file contains a device address that is rejected by the configured DCD address filter. Either the line has been manually added to the wrong seed file, or the address filter needs to be modified to accept this address.
No Such Variable, object type = <type>	The polling request configuration for this object type uses a polling variable that does not exist. Fix the configuration file.
(Sheet 10 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Pattern string is empty	The pattern string passed to the MATCH response handling operator is empty. Fix the problem.
Profile <file name> required for <device name> not defined	A profile name given to a Discover response handling command is not defined. Fix the problem.
Read error on polling configuration file, line <line number>	The polling configuration file is corrupt. Fix the file.
Read error on trap translation data file, line <line number>	The trap translation configuration file is corrupt. Fix the file.
Referenced variable does not exist	The polling variable name given to a VARTYPE or VAROID response handling operator does not exist. Fix the problem.
Referenced variable has no OID	The polling variable name given to an VAROID response handling operator is a local variable and has no OID. Fix the problem.
Registration with trap server failed: <error msg>	The DCD cannot register with the Preside Multiservice Data Manager trap server. Verify the workstation configuration.
Removed address failed. No such address in agent address list	The address to be removed is not found.
Reply received, no request associated with agent <device name>	A reply to an SNMP request has been received from the device after the request data structure has been destroyed. If this message occurs frequently, consider increasing the value of the Dcd_SnmpRespInt configuration parameter.
(Sheet 11 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
Response and snmpRequest Id mismatch	A reply to an SNMP request has been received from the device after the request data structure has been destroyed. If this message occurs frequently, consider increasing the value of the Dcd_SnmpRespInt configuration parameter.
String too long, truncated, line <line number>	The string produced by a response handling/trap translation command exceeds the maximum length. Report the problem to your Nortel Networks representative.
Substring to replace is empty	The REPLACE response handling operator is called with an empty second argument. Fix the problem.
Timestamp string is empty	The TIMESTAMP response handling operator is called with an empty timestamp argument. Fix the problem.
Trap discarded, no translation available	A trap has been received for which no translation is defined, and there is also no default translation. If this message occurs repeatedly, report it to your Nortel Networks representative.
Unexpected left brace, line <line number>	The polling configuration file contains a record that has more than one left brace. Fix the problem.
The "Accumulate" rule is not written correctly in <line>.	There is an unexpected command label on a line. Ensure that the Accumulate rule is correctly written.
The mode is not correct in <line>.	There is an unknown command mode for the Accumulate rule. Correct the command mode.
The "setClear" is already defined.	setClear is defined. Remove this setClear.
The "Incremental" is already defined.	Incremental is defined. Remove this Incremental.
(Sheet 12 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
The mode is not correct in the Accumulate rule in <line>.	There is an unknown command mode for the Accumulate rule. Correct the command mode.
The <trap code> is not the correct expression in <line>.	The trap code in the Accumulate rule cannot be evaluated. Ensure that the trap code is correct.
The <trap code> is not a numerical character in <line>.	The trap code in the Accumulate rule should be numerical. Change the trap code to a number.
The <interval> is not the correct expression in <line>.	The interval in the Accumulate rule cannot be evaluated. Ensure that the interval is correct.
The <interval> is not the numerical character in <line>.	The interval for the Accumulate rule should be numerical. Change the interval to a number.
The <threshold1> is not the correct expression in <line>.	The <threshold1> in the Accumulate rule cannot be evaluated. Ensure that the <threshold1> is correct.
The <threshold1> is not a numerical character in <line>.	The <threshold1> in the Accumulate rule should be numerical. Change <threshold1> to a number.
The <threshold2> is not a numerical character in <line>.	The <threshold2> in the Accumulate rule should be numerical. Change <threshold2> to a number.
The <threshold3> is not a numerical character in <line>.	The <threshold3> in the Accumulate rule should be numerical. Change <threshold3> to a number.
The <trap code> in the Accumulate rule is a negative number in <line>.	The trap code in the Accumulate rule should be greater than zero. Ensure that the trap code is greater than zero.
(Sheet 13 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
The <interval> is 0 or a negative number in <line>.	The interval in the Accumulate rule should be greater than zero. Ensure that the interval is greater than zero.
The <threshold1> in the Accumulate rule is 0 or a negative number in <line>.	The <threshold1> in the Accumulate rule should be greater than zero. Ensure that <threshold1> is greater than zero.
The <threshold2> in the Accumulate rule is 0 or a negative number in <line>.	The <threshold2> in the Accumulate rule should be greater than zero. Ensure that <threshold2> is greater than zero.
The <threshold3> in the Accumulate rule is a negative number in the <line>.	The <threshold3> in the Accumulate rule should be greater than zero. Ensure that <threshold3> is greater than zero.
The <threshold3> in the Accumulate rule is equal or greater than the <threshold1> in <line>.	The <threshold3> in the Accumulate rule cannot be greater than or equal to the rule’s <threshold1>. Ensure that <threshold3> is less than and not equal to <threshold1>.
The <compid> is not defined when the Accumulate is used. Define the <compid> before using the Accumulate rule.	Ensure that the CompId has been defined before using the Accumulate rule. Define the CompId before using the Accumulate rule.
The <node> is not the top component in <line>.	The <node> is not the top component in <line>. The Accumulate object can only be the attribute of the top component. This <node> is the wrong node.
The <agent> does not exist. The internal trap cannot be created.	The agent must exist when using the Accumulate rule. In this case, the agent does not exist which results in the creation of a failed internal trap.
(Sheet 14 of 15)	

Table 73 (Continued)
ERROR messages for the DCD

Message	Meaning and action
The top component <node> cannot be found.	The Accumulate rule cannot find the top component likely because the top component is the wrong top node.
This component <component name> doesn’t belong to this agent.	The DCD has received a trap for the wrong device. Check the configuration files to ensure that it contains the correct devices.
(Sheet 15 of 15)	

Table 74
MAJOR messages for the DCD

Message	Meaning and action
<anAddr> is added to <anAgent> polling address list	An IP address is added to a device polling address list.
<anAddress>is added to <anAgent> trap address list.	An IP address is added to a device trap address list.
<anAgent>is deleted	A device of this name is deleted.
cannot open option file: <filename>	The DCD cannot open the run-time options file. This is not necessarily a problem since, if default values or command line options are used for all configuration parameters, this file is not needed.
Cannot open seed file <filename> for reading	The DCD cannot open the seed file in read mode. This is not necessarily a problem since, if no devices have been discovered yet, this file is not needed.
Configuration files verification completed	The -v option was used and verification of configuration files completed.
<dcd name> is starting	The DCD process is starting.
(Sheet 1 of 2)	

Table 74 (Continued)
MAJOR messages for the DCD

Message	Meaning and action
Device <devName> at <ipaddr> is replaced by Device <devName>	The old device at the IP address is replaced by the new device.
Lost Trap Server connection	The connection with the trap server daemon (TSVR) has been lost. DCD periodically tries to restore this connection; until it is successful, no traps are received from the devices.
reading <file name>	The DCD is starting to read the configuration file.
reading option file: <filename>	The DCD is starting to read the identified run-time options file.
reading polling cfg file: <filename>	The DCD is starting to read the identified polling configuration file.
Reading trap translation cfg file <file name>	The DCD is starting to read the identified trap translation configuration file.
Registration with trap server completed	The DCD has registered, or re-registered, with the trap server daemon (TSVR).
(Sheet 2 of 2)	

Chapter 60
SNMP IP Discovery server (IPDSVR)

This section contains information on the SNMP IP Discovery server (IPDSVR). See the following topics for more information.

About the SNMP IP Discovery server

The SNMP IP Discovery (IPDSVR) server monitors the changes to the properties of SNMP devices. SNMP-managed devices can be viewed and monitored through the IP Discovery graphical-user interface (GUI). The following device properties are displayed:

- the IP address of the device
- the Read Community string used to communicate with the device, and
- the SNMP version used to communicate with the device

The devcache is located in the directory /opt/MagellanNMS/cfg/ipm. The refresh interval specified in the command line option for this server determines how often the device property changes are updated in the devcache and in the IP Discovery GUI. For more information, see "Startup command" (page 506).

The svmadm tool starts the IPDSVR server. Once the server is started, it establishes a connection with the GMDR server and synchronizes with any changes to the existing SNMP-managed devices. The IPDSVR server periodically queries the GMDR server for information about discovered devices. It populates and updates a seed file called the devcache with this device information

Managing the IPDSVR server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for the IPDSVR server. Any changes you make to the startup command or options become active when the IPDSVR server is restarted. See the 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

See the following:

- “Suggested name in Server Administration” (page 506)
- “Startup command” (page 506)

Suggested name in Server Administration

The recommended name for the IPDSVER server is SNMP IP Discovery Svr.

Startup command

The startup command for the IPDSVR is:

```
/opt/MagellanNMS/bin/ipdsvr [-h] [-i <refresh_int>]
```

where:

[-h] displays the command usage.

[-i <refresh_int>] specifies the refresh interval in seconds. The default value is 600 seconds or 10 minutes.

Setting up the refresh interval for IPDSVR

The refresh interval for IPDSVR determines how often the server queries the GMDR for device properties, and updates the devcache file when device properties have changed for SNMP devices.

Criteria to consider before specifying the refresh interval

Querying GMDR for device properties consumes CPU and GMDR resources. Before deciding on a refresh interval, consider the following questions:

- How soon should a device property change be propagated to the GUI?
- How frequently do these changes occur in the user network?
- How critical is CPU and GMDR resource utilization in the user environment?

The duration of the refresh interval value affects the following:

- GMDR server performance
- CPU utilization
- the amount of time for device property changes to appear in GUI

The default value for the refresh interval is 10 minutes. If the user does not require notification of device property changes, the user can set the refresh interval to 0 to prevent the querying of devices. With this configuration, property changes are not updated in the devcache and IP Discovery GUI. This is not a recommended configuration.

Interdependencies

The IPDSVR server depends on the GMDR server.

Exit codes

Exit codes for the IPDSVR server are shown in the following table:

Table 75
Exit codes for the IP Discovery server

Exit code	Description
1	Error values returned from GMDR EPI
4	Error connecting to GMDR or GMDR interface not connected
50	Another instance already running
55	Invalid command line argument

Chapter 61
SNMP Management Data Router (SMDR)

This section contains information on the SNMP Management Data Router (SMDR). See the following topics for more information:

About the SMDR server

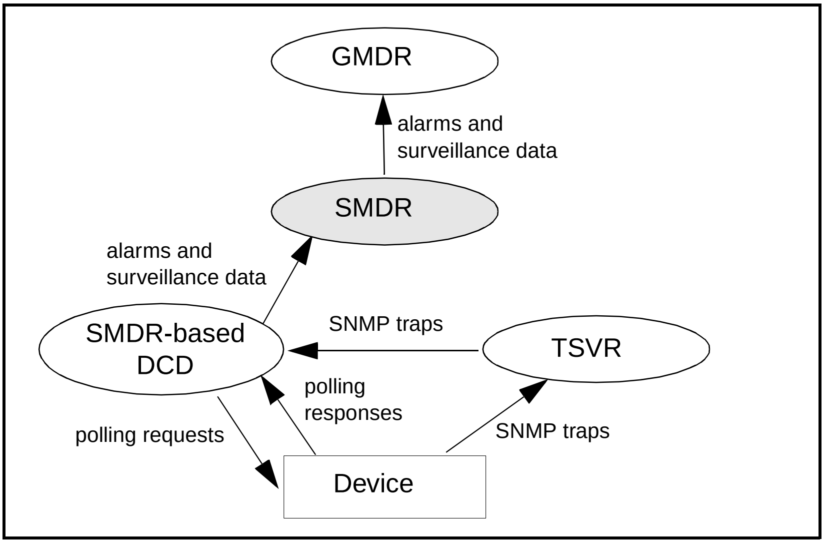
The SMDR is responsible for merging the SNMP surveillance data obtained from SMDR-based DCDs (data collection daemons) and making it available to the General Management Data Router (GMDR).

SMDR performs the following functions:

- collects surveillance data from SMDR-based DCDs
- supports REGISTER, GET, CREATE, and ACTION API requests from GMDR

- forwards alarms and raw state change notifications to GMDR
- issues proxy alarms when a polling state indicates that a component is down while the active alarm list is empty, or when a device becomes unreachable or reachable again
- clears outstanding active alarms when the polling state indicates the component is up
- automatically deletes components when a disconnected SMDR is reconnected with a DCD - this is referred to as resynchronization-based automatic component deletion. For SMDR disconnects that are the result of administrator action, the components are deleted on the SMDR and the GMDR and are also deleted from the network model and fault applications using the fault API or EPI. For SMDR disconnects that are the result of communication failure, the components stay in the unknown state until the component indicates no alarms; the component is then eligible for deletion. Exceptions to resynchronization-based automatic component deletion include:
    - partial view sub-servers
      Components not deleted on a server can be deleted by a client of this server if the client has multiple data providers for a node and one, or more, of these data providers provides only a partial view of the node.
    - component discovered by event, only
      Components that exist in the network can be deleted throughout the fault stack. Most components discovered by event, only, have already been removed from the network model because the network model recognizes these components as dynamic.
    - components with historical alarms
      Components with historical alarms are tagged as eligible for deletion but are not deleted. After a resynchronization of servers, if the component no longer has any alarms, the component is deleted. This situation only occurs if the disconnect is due to network congestion; the active alarms become historical alarms.
    - a server using pre-release R14.1 Preside Multiservice Data Manager software and any server above this server in the fault stack; unless one of the servers above this server in the fault stack is a redundant server using software release R14.1, or greater.

Figure 53
SMDR data flow diagram



Managing the SMDR server

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for SMDR. Any changes you make to the startup command or options take effect when SMDR is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for the instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 512)
- “Startup command” (page 512)

Suggested name in Server Administration

The recommended name for the SNMP Management Data Router is SMDR.

Configuring SMDR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start SMDR has the following syntax:

```
/opt/MagellanNMS/bin/smdr \
[-b <congestion buffer size>] \
[-f]
```

where:

`[-b <congestion buffer size>]` specifies the maximum number of congested replies before the client is cut off. This option protects against slow, disconnected, or non-communicating clients, such as when a client has disconnected or cannot communicate with SMDR. If this option is not specified, the default value of 5,000 is used. The default value is adequate for most installations.

`[-f]` disables the resynchronization-based deletion of components.

Interdependencies

SMDR relies on the SMDR-based DCD (data collection daemon) for the device that is being supported.

Configuration

There are no mandatory configuration files to manually set up for SMDR. There are two configuration files that you can edit. See the following sections:

- “Run-time options file” (page 513)
- “Server list configuration file” (page 513)

Run-time options file

You can use this file to override the default values of configuration parameters. Its location is /opt/MagellanNMS/cfg/smdr.cfg.

You can override some configuration parameters by specifying them as options in either the run-time options file or in the SMDR startup command. If you specify the same configuration parameter in both places, the value in the run-time options file is used.

The following list shows the configuration parameters that you can edit in the run-time options file.

- congLevel, which specifies the maximum number of congested replies before the client is cut off. This option protects against slow, disconnected, or non-communicating clients, such as when a client has disconnected or cannot communicate with SMDR. Its default value is 5,000.
- maxServers, which specifies the maximum number of server processes. Its default value is 60.
- maxSieves, which specifies the maximum number of sieves that SMDR is allowed to create. Its default value is 360.

Server list configuration file

This configuration file identifies the DCD from which SMDR obtains its surveillance data. Its location is /opt/MagellanNMS/cfg/smdr.svr.

You can use an editor such as vi to add entries to this file. Then restart SMDR to connect to the new servers. You can connect to a new server without stopping and starting SMDR by using the smdrCreateServer script; see “Action script smdrCreateServer” (page 514) for more information. You can disconnect from a server without stopping and starting SMDR by using the smdrDeleteServer script; see “Action script smdrDeleteServer” (page 514) for more information.

There is one line for each server in the server list configuration file. The format of each line is

```
:<host name>:<server name>:<user id>:<password>:
```

where user ID and password are optional and can be left empty.

Example

:host24d:gendcd:::

Action script smdrCreateServer

This script sends a request to SMDR to connect to the named DCD. The syntax of this script is

```
/opt/MagellanNMS/bin/smdrCreateServer <name> \
<host>
```

where:

`name` is the DCD process name, for example gendcd_ott

`host` is the workstation name or IP address of the workstation where the DCD resides

Example

```
smdrCreateServer gendcd host24aa
```

Action script smdrDeleteServer

This script sends a request to SMDR to disconnect from the named DCD and to remove the DCD from the SMDRs server list configuration file. The syntax of this script is

```
/opt/MagellanNMS/bin/smdrDeleteServer <name> \
<host>
```

where:

`name` is the DCD process name, for example `idi_genericdcd`; this name is case sensitive and must match the name recorded in the server list configuration file

`host` is the workstation name or IP address of the workstation where the DCD is (or was) running, for example `host33xx`; this name is case sensitive and must match the name recorded in the server list configuration file

Example

```
smdrDeleteServer idi_genericdcd host33xx
```

Note: This script can also be used to remove a DCD process that is not running from the SMDRs server list.

If the DCD is running, the DCD process is not stopped. This script only causes the SMDR to disconnect from the DCD.

Proxy alarms

Most alarms are translations of traps. In addition to this type of alarm there are proxy alarms, which occur as a result of polling. Proxy alarms occur under the following circumstances:

- When reachability polling fails, SMDR issues the proxy alarm 09990001.

  A proxy SET alarm is issued when connectivity to the device is lost. This alarm carries a state of UNKNOWN and deletes the current active alarms.

  A proxy CLR alarm is issued when connectivity to the device is regained and puts the device back in service (INSV).

- State or configuration polling indicates that the state of a component is different from what the active alarms indicate. If SMDR receives a state change notification (SCN) that is not compatible with the component’s active alarm list, it issues the proxy alarm 09990013.

  A proxy SET alarm is issued when a state change notification indicates that the node is in a worse state than indicated by the active alarm list.

  A proxy CLR alarms is issued when a state change notification indicates that the node is in a better state than indicated by the active alarm list. This alarm clears the active alarms that are not compatible with the new state.

- A proxy CLR alarm can be issued when a proxy SET alarm is issued as a result of a polling reply and a trap is subsequently received that identifies the problem. In this case, the proxy CLR alarm clears the proxy SET alarm before a normal SET alarm is issued.

Exit codes

Exit codes for when SMDR terminates under its own control are shown in the following table. SMDR can terminate with other exit codes if it crashes or is stopped by the Server Administration tool.

Table 76
Exit codes for SMDR

Exit code	Description
5	One of the mandatory configuration files cannot be processed, or some mandatory configuration parameters are undefined.
6	Registration with MNSD failed.
7	Software error
8	Memory allocation failed.

Error messages

Error messages for SMDR fall into the following general categories:

- FATAL, for errors that are fatal to the operation of SMDR. This category of message always causes SMDR to terminate. It is generally caused by a faulty workstation setup or a bad process configuration. See the table “FATAL messages for SMDR” (page 517).
- SNO (should not occur), when SMDR experiences an error situation that should not occur (probably a software error). The error is logged and SMDR tries to continue executing. If an error of this type occurs repeatedly, report it to your Nortel Networks representative. See the table “SNO (should not occur) messages for SMDR” (page 517).

- ERROR, when SMDR experiences an error situation that is probably caused by an invalid configuration or switch input. The cause needs to be identified and corrected. See the table “ERROR messages for SMDR” (page 518).
- MAJOR, when SMDR signals the occurrence of an important event (usually not an error), such as the establishment or the loss of the connection with a server. See the table “MAJOR messages for SMDR” (page 520).

Table 77
FATAL messages for SMDR

Message	Meaning and action
IPC initialization failed	MNSD is not running or else it rejected the SMDR registration; Preside Multiservice Data Manager is not properly installed. Fix the installation.

Table 78
SNO (should not occur) messages for SMDR

Message	Meaning and action
Could not find/create comp: <component name>	SMDR cannot create a new component in its component tree. This is an internal software error. Report it to your Nortel Networks representative.
Reachability event for non-device	An incoming alarm that is reporting device unreachability has a component identifier that does not represent a device. This is an internal software error. Report it to your Nortel Networks representative.
Reachability SET alarm received from server	An incoming alarm that is reporting device unreachability has been received before the corresponding server notification. This is an internal software error. Report it to your Nortel Networks representative.

Table 79
ERROR messages for SMDR

Message	Meaning and action
attribute name not a token: <attribute value>	SMDR does not support non-tokenized attributes in GET replies. This is an internal software error. Report it to your Nortel Networks representative.
ERROR ALARM SIEVE CREATE for server: <server name>	The server rejected an SMDR CREATE request for an alarm sieve. Either SMDR is trying to connect to a server of an unsupported type (not a DCD) or this server has already created the maximum number of sieves it can support.
ERROR GET ALARM request for server: <server name>	The server rejected an SMDR GET request for alarms. SMDR should only send this request to hierarchical SMDR subservers (unsupported). If this request is sent to a DCD, it is rejected but the server connection process continues.
ERROR GET RAW STATE request for server: <server name>	The server rejected an SMDR GET request for components and states. SMDR is probably trying to connect to a server of an unsupported type, or a communication problem has occurred. Delete this server from the server list configuration file, restart SMDR, and use the smdrCreateServer script to add the server again.
Error reading option file: <filename>	The contents of the run-time options file are corrupted. Restore the file contents.
ERROR SERVER RESET SIEVE CREATE for server: <server name>	The server rejected an SMDR CREATE request for a raw state sieve. Either SMDR is trying to connect to a server of an unsupported type (not a DCD) or this server has already created the maximum number of sieves it can support.
Invalid address filter <filter> from server <server name>	The server has specified an invalid address filter in its reply to an SMDR registration request. This is an internal software error. Report it to your Nortel Networks representative.
Invalid command line parameter: <char>	There is an invalid option in the command line. Fix the command line.
(Sheet 1 of 2)	

Table 79 (Continued)
ERROR messages for SMDR

Message	Meaning and action
Invalid line (<line number>) in run-time cfg	The run-time options file contains an invalid line. Fix the file.
Invalid option <option label> on line <line number>	The run-time options file contains an invalid option. Fix the file.
SMDR - Circular server specification	SMDR is trying to connect to itself. There is a bad entry in server list configuration file or the smdrCreateServer script has been used incorrectly.
SMDR - Communications error (<error code>): passthrough request	A server has rejected a passthrough request. This is an internal communication problem. Report it to your Nortel Networks representative.
SMDR - Configuration file read error	The contents of the server list configuration file are corrupted. Restore the file contents.
SMDR - Configuration file write error	SMDR cannot write to the server list configuration file. Verify the file permissions and disk capacity.
SMDR - Could not open configuration file	SMDR cannot open the server list configuration file. Verify the file permissions.
SMDR - Duplicate server specification	SMDR is being told to connect to a server twice. This can be caused by a duplicate entry in the server list configuration file (delete the duplicate entry in this case) or else by using the smdrCreateServer script for a server that is already listed in the server list configuration file.
SMDR - Invalid server specification	Either the host name or the server name specification is missing from an entry in the server list configuration file.
(Sheet 2 of 2)	

Table 80
MAJOR messages for SMDR

Message	Meaning and action
cannot open option file: <filename>	SMDR cannot open the run-time options file. This is not necessarily a problem since, if default values or command line options are used for all configuration parameters, this file is not needed.
reading option file: <filename>	SMDR is starting to read the identified run-time options file.
SMDR is starting	The process is starting.
SMDR - <client name>: killed client connection due to congestion	The specified client connection has been dropped due to congestion.
SMDR - <client name>: unexpectedly lost client connection.	The specified client connection has been lost.
SMDR - <server name>: lost connection	The specified server connection has been lost.

Chapter 62
SNMP Proxy Agent (SPA) on MPE 9500

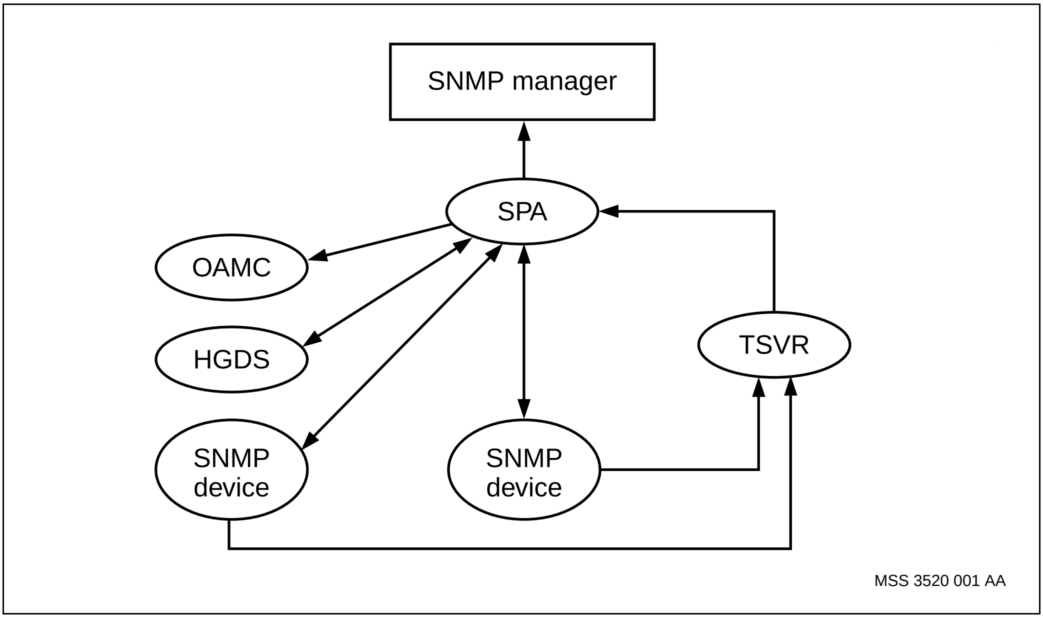
This section contains information on the SNMP Trap Proxy Agent (SPA). See the following topics for more information:

About SPA

The SNMP Proxy Agent server process receives SNMP requests from SNMP management processes, forwards the requests to an MPE 9500 device and forwards the replies back to the SNMP manager. SPA also forwards SNMP traps received from MPE 9500 devices to registered SNMP managers.

Each SPA instance is created and monitored by the MDM server manager. It obtains the list of MPE 9500 devices it provides access to from the host group directory server (HGDS) and receives traps issued by the MPE 9500 devices through the MDM trap server (TSVR). SPA issues logs to notify operators of important server events and problems. These logs are collected and displayed by the MDM OAM log collector.

Figure 54
SPA data flow diagram



Managing SPA

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for SPA. Any changes you make to the startup command or options take effect when SPA is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following sections:

- “Suggested name in server administration” (page 522)
- “Startup command” (page 523)

Suggested name in server administration

The recommended name for the proxy agent is SNMP Proxy agent.

Configuring SPA with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start SPA has the following syntax:

```
/opt/MagellanNMS/bin/spa \
[-m <max requests] \
[-p <port number>]
```

where:

`-m <max requests>` specifies the maximum number of concurrent requests that this SPA can handle. This value is limited to 1000 requests which is close to the maximum number of UDP sockets per process (1024 on Solaris 8). The default value is 200 if not specified on the command line.

When SPA is currently handling the maximum number of requests defined by this parameter and it receives a new SNMP request, it issues an ALERT log and discards the request. Subsequently, if the number of concurrent requests falls below this maximum number by 5 requests, a CLEARED log is issued. To avoid issuing too many ALERT logs, if other new requests must be discarded before the CLEARED log is issued, the ALERT log is replaced by an ERROR log.

`-p <port number>` specifies the UDP port number where SPA receives requests from SNMP managers. The default value is 361 if not specified on the command line.

Interdependencies

SPA relies on the trap server daemon (TSVR) and the host group directory server (HGDS).

Configuration

Each SNMP proxy agent instance needs two configuration files:

- “SNMP managers configuration file” (page 524)

- “Runtime options configuration file” (page 524)

Note: Both files can contain blank lines or comment lines beginning with “#” or “!”.

SNMP managers configuration file

The SNMP managers configuration file lists the SNMP managers to which traps must be forwarded. For each manager, the file specifies:

- Manager IP address
- Required SNMP version
- Destination UDP port (This parameter is optional. If none is specified, 162 is used.)

This SNMP managers file is named: /opt/MagellanNMS/cfg/spa_<port number>.mgr. SNMP managers not present in this list are not prevented from sending SNMP requests to SPA. However, SPA does not forward traps to them. This file has the following format:

```
SNMP v1

Manager:<manager IP address> [<manager port>]

SNMP v2c

Manager:<manager IP address> [<manager port>]
```

This file must be created by the customer for each SPA instance. The file /opt/Magellan-NMS/lib/cfg/spa.mgr can be used as a template to be copied to the /opt/MagellanNMS/cfg directory, then renamed and completed.

Runtime options configuration file

Note 1: Any other modification of a run-time parameter (including command line options) requires the server to be stopped and restarted.

Note 2: If one of the new files contains invalid options or syntax errors, SPA terminates after issuing the proper FATAL log.

The runtime options configuration file lists the following SPA instance runtime parameters: /opt/Magellan-NMS/cfg/spa_<port number>.cfg. The parameters that can be specified in this file are:

- hgdsGroup represents the MPE 9500 HGDS group names. By default, all the MPE 9500 groups configured in HGDS are used. The user can specify a comma separated group list. For example: hgdsGroup: G1,G2,G5.

  Addition of new groups as well as deletion of existing groups is supported. The introduction of a new hgdsGroup option as well as the suppression of the current option are also supported. If modification of this option results in the suppression of a managed device against which there are outstanding requests, those requests are discarded.

- snmpRespInt represents the waiting interval (in seconds) for a response to an SNMP request before re-sending the request. The default value is 10. Each request is re-sent 2 times before being discarded for lack of response. This later value is not configurable. New values are allowed. However, it will only be applied to new requests.

  Note: SNMP manager's SNMP response time-out value is recommended to be larger than this value to make sure SPA has enough time to wait for an SNMP response in normal operational conditions.

- statisticInt represents the interval time (in minutes) between two sets of statistical logs issued. The default value is 1440 (24 hours). All modifications are allowed.

- addressFilter represents the character string passed to TSVR to filter the traps forwarded to this SPA. The default value is "*" (For example: accept all IP addresses). The address filter supported formats are described in the next section.

  TSVR filters received traps according to the address filter supplied by its client SPAs and forwards them to the proper clients. However, if the user wants to use this parameter more efficiently, MPE 9500 devices in the same logical region should be allocated IP addresses in the same subnetwork.

  This parameter can be specified several times in the runtime parameters configuration file if this SPA instance is intended to manage the MPE 9500 devices in many subnetworks.

This parameter is only useful when there are more than one SPA instance on a workstation each managing MPE 9500 devices from different logical network regions. This parameter is optional even in this case. Any modification is ignored since this would require disconnecting and reconnecting with TSVR

- logLevels represent a comma separated list of the log levels which this SPA instance writes to the process log file. The values that can be included in this list are: FATAL, ALERT, CRITICAL, ERROR, WARNING, CLEARED, NOTICE, INFO, DEBUG, TRACE. The levels selected by default are FATAL, ALERT, CRITICAL, ERROR, CLEARED, NOTICE, INFO. For example: logLevels: ERROR,DEBUG,TRACE. All modifications are allowed.
- defaultComm This parameter specifies a default community string used for incoming SNMP requests only supplying the NE_id in their community string. This default community string is then used in the request sent to the device. The default value is “public”. All modifications are allowed.

Exit codes

The Exit codes for SPA are displayed in the following table.

Table 81
Exit codes for SPA

Exit code	Description
0	Normal exit
1	Failed to perform configuration
2	Failed to perform initialization of IPC
4	Memory error
5	Failed to open session
(Sheet 1 of 2)	

Table 81 (Continued)
Exit codes for SPA

Exit code	Description
6	Failed to allocate session
7	No device
(Sheet 2 of 2)	

Chapter 63
SNMP Trap Server Daemon (TSVR)

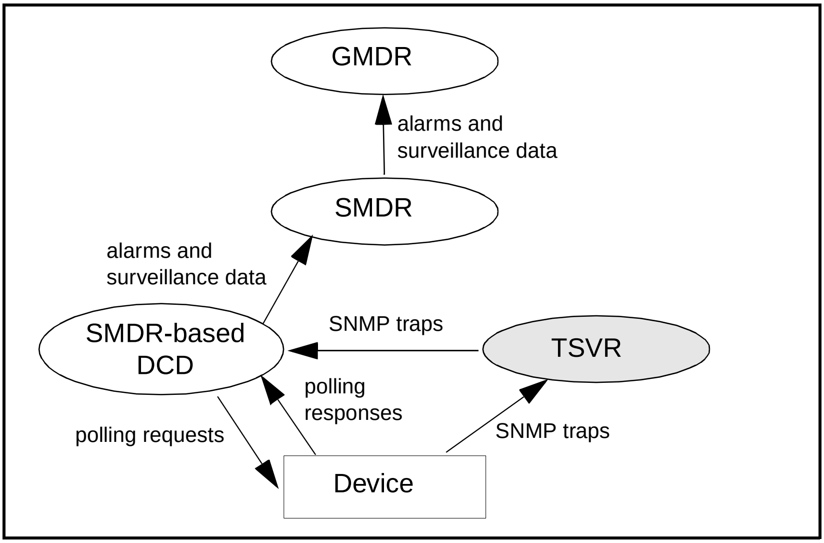
This section contains information on the SNMP Trap Server Daemon (TSVR). See the following topics for more information:

- “About the TSVR” (page 529)
- “Managing the TSVR” (page 530)
- “Interdependencies” (page 531)
- “Exit codes” (page 531)
- “Error messages” (page 532)

About the TSVR

The trap server daemon (TSVR) receives trap information from network elements that are managed by Preside Multiservice Data Manager using an SMDR-based DCD (data collection daemon) and forwards the traps to a registered DCD. Before forwarding the traps to the DCD, TSVR filters them according to the filter rules of the DCD.

Figure 55
TSVR data flow diagram



Managing the TSVR

Use the Server Administration tool to enter or edit the startup command and to start, stop, and set options for TSVR. Any changes you make to the startup command or options take effect when TSVR is restarted. See 241-6001-303 *Preside MDM Administrator Guide* for instructions to use the Server Administration tool.

For more information, see the following:

- “Suggested name in Server Administration” (page 530)
- “Startup command” (page 531)

Suggested name in Server Administration

The recommended name for the trap server daemon is SNMP Trap Server.

Configuring TSVR with the Server Administration tool requires you to enter the server name in the Descriptive name field of the Server Administration dialog for new servers. This name also identifies the server in file /opt/MagellanNMS/cfg/SVMList.cfg. This file lists the servers that are to be started automatically when the workstation is rebooted.

Startup command

The command to start TSVR has the following syntax:

```
/opt/MagellanNMS/bin/tsvr \
[-a <ip address>] \
[-b <congestion buffer size>] \
[-p <port number>]
```

where:

`-a <ip address>` specifies the address that the TSVR binds to. When you use this command, the TSVR binds to the specified address only, not all of the addresses on the port.

`-b <congestion buffer size>` specifies the maximum number of congested replies before the client is cut off. This option protects against slow, disconnected, or non-communicating clients, such as when a client has disconnected or cannot communicate with TSVR. If this option is not specified, the default value of 5,000 is used. The default value is adequate for most installations.

`-p <port number>` specifies the port number that receives the SNMP traps. The default value is 162.

Interdependencies

TSVR relies on the SMDR-based DCD (data collection daemon) for the device that is being supported.

Exit codes

Exit codes for TSVR are shown in the following table.

Table 82
Exit codes for TSVR

Exit code	Description
51	Out of memory.
55	Bad arguments on the command line.
(Sheet 1 of 2)	

Table 82 (Continued)
Exit codes for TSVR

Exit code	Description
57	Unable to fork TSVRREP child process.
59	Cannot initialize IPC system.
60	Cannot register service (probably already running)
(Sheet 2 of 2)	

Error messages

Error messages for TSVR are shown in the following table.

Table 83
Error messages for TSVR

Error message	Meaning and action
TSVR -- Invalid command line argument	The command line has an invalid option.
TSVR -- Killed client connection due to congestion	The client connection was terminated due to either a client that exceeded the congestion buffer size or a client that has no further communication.
TSVR -- Unexpectedly lost client connection.	The client has disconnected without issuing a disconnect request, or the communication path with the client has been terminated.

Chapter 64
Secure FTP Daemon (launchSecureFTPD)

This section contains information about the Secure FTP Daemon. See the following sections for information about the launchSecureFTPD daemon:

- About the Secure FTP Daemon (page 533)
- Managing the Secure FTP Daemon (page 533)
- Interdependencies (page 534)

About the Secure FTP Daemon

The Secure FTP Daemon is responsible for encrypting and decrypting passwords transferred by means of FTP Protocol between a node in the network and the Preside Multiservice Data Manager workstation.

The Secure FTP daemon is used primarily to ensure that the password a node uses to log in to a workstation for downloading software is secure.

Managing the Secure FTP Daemon

This section contains information about the following topics:

- Configuration (page 534)
- Suggested name in Server Administration (page 534)
- Start-up command (page 534)

Configuration

Configuring and starting the Secure FTP Daemon is part of a much larger task: configuring IP security. Configuring IP security involves not just Preside Multiservice Data Manager, but also other devices in the network, and is beyond the scope of this section. For the instructions to plan, configure, and install IP Security, see NN10600-607 *Passport - MDM Network Security: Secure Communications Configuration*.

Suggested name in Server Administration

The recommended name for the Secure FTP Daemon to enter in the Server Manager Administration tool is Secure FTP Daemon.

Configuring the Secure FTP Daemon with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the Secure FTP Daemon has the following syntax:

```
/opt/MagellanNMS/bin/launchSecureFTPD
```

Interdependencies

None

Chapter 65
VPN Monitor Extractor (VPNMonitorExtractor)

This section contains information about the VPN Monitor Extractor (VPNMonitorExtractor). See the following topics for information about this server:

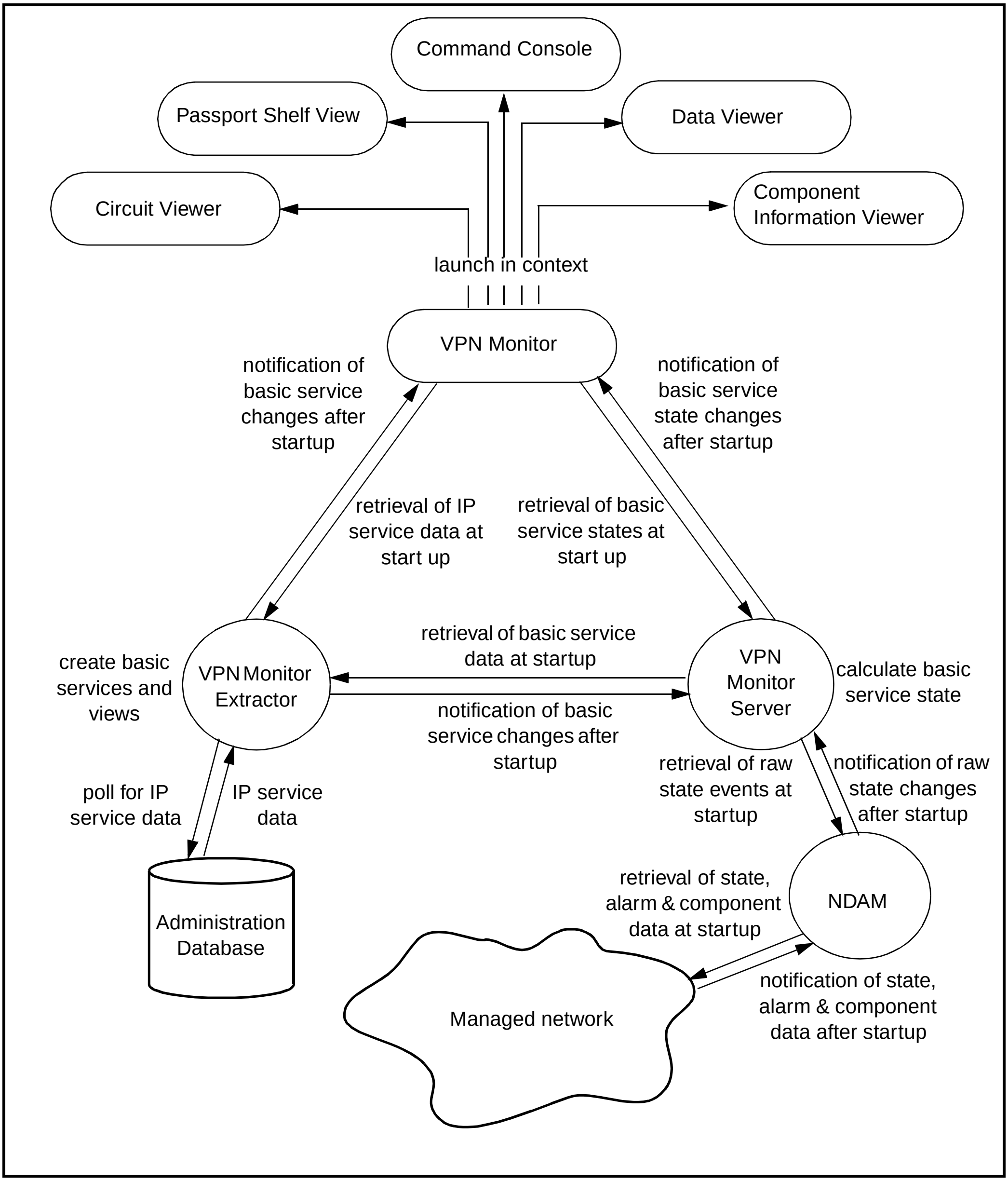
- About the VPN Monitor Extractor (page 535)
- Managing the VPN Monitor Extractor (page 537)
- Interdependencies (page 539)
- Exit codes (page 539)

About the VPN Monitor Extractor

The VPN Monitor Extractor server works in conjunction with the VPN Monitor Server as part of the VPN Monitor tool for monitoring virtual private networks (VPN). For details about the VPN Monitor Server, see VPN Monitor Server (VPNMonitorServer) (page 541).

The VPN Monitor Extractor collects VPN service data from the Administration Database and delivers the data to the VPN Monitor server and client. The figure VPN Monitor data flow diagram (page 536) provides a graphical representation of the data flow from the Administration database source to the VPN Monitor Client.

Figure 56
VPN Monitor data flow diagram


Command Console
Passport Shelf View
Data Viewer
Circuit Viewer
Component Information Viewer
launch in context
VPN Monitor
notification of basic service changes after startup
notification of basic service state changes after startup
retrieval of IP service data at start up
retrieval of basic service states at start up
create basic services and views
VPN Monitor Extractor
retrieval of basic service data at startup
notification of basic service changes after startup
VPN Monitor Server
calculate basic service state
poll for IP service data
IP service data
retrieval of raw state events at startup
notification of raw state changes after startup
Administration Database
retrieval of state, alarm & component data at startup
NDAM
Managed network
notification of state, alarm & component data after startup

Database polling

At startup, the extractor retrieves all of the VPN service-related data in the Administration database.

After the initial VPN data collection, the extractor regularly polls the database to check for any changes in the VPN data. If the extractor determines that the IP services have been modified since the last poll, the extractor notifies the VPN Monitor servers and clients of these changes.

Automatic polling

The default polling interval is 30 minutes. However, the timer for a new poll begins only when the current poll is complete. If you need, you can change the polling interval by changing the -sleep parameter on the startup command for the extractor server.

Manual polling

The network administrator can manually trigger a poll using the “kick” script. The kick script causes polling to begin immediately. If a poll is in progress when you execute the kick script, the manual poll begins after the poll in progress completes. The script has the following syntax:

```
/opt/MagellanNMS/bin/VPNMonitorExtractor.kick
```

Redundancy

The VPN Monitor extractor can deliver VPN related data to more than one instance of a VPN Monitor server. Because the extractor supports communication with multiple servers on different workstations, if one server fails, then you can connect to the redundant VPN Monitor server.

If required, a second extractor can be set up on a different workstation to deliver the VPN data to a different set of servers for full redundancy.

Log file

The */opt/MagellanNMS/data/log/VPNMonitorExtractor/VPNMonitorExtractor.alog* file captures the Extractor’s events.

Managing the VPN Monitor Extractor

This section contains information about the following topics:

- Configuration (page 538)

- Suggested name in Server Administration (page 538)
- Start-up command (page 538)

Configuration

The VPN Monitor Extractor uses the */opt/MagellanNMS/cfg/dba/dbaccess.cfg* to locate and access the Administration database.

To change the default settings of this configuration file, use the Database Administration tool. For details, see the procedure for setting configuration options in 241-6001-400 *Preside MDM Administration Database User Guide*.

Suggested name in Server Administration

The recommended name for the VPN Monitor Extractor to enter in the Server Manager Administration tool is VPN Monitor Extractor.

Configuring the VPN Monitor Extractor server with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

The start-up command to start the VPN Monitor Extractor server has the following syntax:

```
/opt/MagellanNMS/bin/VPNMonitorExtractor \
-dbuser <string>|-dbu <string> \
-dbpassword <string>|-dbp <string> \
[-port <number>] \
[-sleep <period>] \
[-help|-h]
```

where:

`-dbuser <string>|-dbu <string>` specifies the Administration database user ID. If the user ID is not encrypted, specify the user ID itself. If the user ID is encrypted, specify the file name that contains the encrypted user ID.

`-dbpassword <string>|-dbp <string>` specifies the Administration database password. If the password is not encrypted, specify the password itself. If the password is encrypted, specify the file name that contains the encrypted password. For details about password encryption, see NN10600-605 *Passport - MDM Network Security: Operations*.

`[-port <number>]` specifies the TCP port number that the VPN Monitor Extractor uses to listen for communication with the client. Do not use this port number for any other process. The VPN Monitor Extractor registers this port with the Preside Multiservice Data Manager context server. If you do not specify a port number, the server dynamically assigns an available port number.

`[-sleep <period>]` specifies the time interval (in minutes) between database polls. The allowable range is from 1 to 10000 minutes. The default interval is 30 minutes.

`[-help|-h]` displays the parameters on the Extractor start-up command.

Interdependencies

The Preside Multiservice Data Manager Administration Database must be installed and running before you start the VPN Monitor Extractor.

Exit codes

For exit codes produced by the VPN Monitor Extractor server, their meanings, and corrective action, see Table 84.

Table 84
Exit codes for the VPN Monitor Extractor server

Exit Code	Meaning and corrective action
50	Do not restart server.
53	Communication failure.
55	Invalid command line argument.
56	Configuration directory does not exist.
58	Normal exit.
(Sheet 1 of 2)	

Table 84 (Continued)
Exit codes for the VPN Monitor Extractor server

Exit Code	Meaning and corrective action
59	Missing virtual machine.
60	Port number is already in use.
(Sheet 2 of 2)	

Chapter 66
VPN Monitor Server (VPNMonitorServer)

This section contains information about the VPN Monitor server (VPNMonitorServer). See the following topics for information about the VPN Monitor server:

About the VPN Monitor Server

The VPN Monitor Server works in conjunction with the VPN Monitor Extractor as part of the VPN Monitor tool for monitoring virtual private networks (VPN). For information about the VPN Monitor Extractor, see VPN Monitor Extractor (VPNMonitorExtractor) (page 535).

The VPN Monitor Server receives VPN basic service information and associated components from the VPN Monitor Extractor, as well as component raw state information from NDAM. Based on the basic service definitions and the raw state of the basic service components, the VPN Monitor Server calculates the status of basic services and forwards this information to the VPN Monitor Client. For details about VPN basic services and their component makeup, see the VPN Monitor section in 241-6001-011 *Preside MDM Fault Management User Guide*.

The figure VPN Monitor data flow diagram (page 536) provides a graphical representation of the flow of data from the Administration database source to the VPN Monitor Client.

Log files

The */opt/MagellanNMS/data/log/VPNMonitorServer/ VPNMonitorServer.alog* file captures the server's events.

Managing the VPN Monitor Server

This section contains information about the following topics:

- Suggested name in Server Administration (page 542)
- Start-up command (page 542)

Suggested name in Server Administration

The recommended name for the VPN Monitor Server to enter in the Server Manager Administration tool is VPN Monitor Server.

Configuring the VPN Monitor Server with the Server Administration tool requires that you enter the server name in the Descriptive name field of the Server Administration dialog. The Server Administration tool writes this information into the /opt/MagellanNMS/cfg/SVMList.cfg file. This file lists the servers that are to be started automatically when the workstation reboots.

Start-up command

You can start the VPN Monitor Server from the Server Administration tool (svmadm) or from the command line. From the command line, the start-up command for the VPN Monitor Server has the following syntax:

```
/opt/MagellanNMS/bin/VPNMonitorServer \
[-extractorHost <hostName>] \
[-port <portNum>] \
[-ndam <servicename@host>]
[-h]
```

where:

`[-extractorHost <hostName>]` specifies the host name or IP address of the machine running the extractor service. If you do not specify this parameter, localhost is used as the default.

`[-port <portNum>]` specifies the TCP port number that the VPN Monitor Server uses to listen for communication with the client. Do not use this port number for any other process. The VPN Monitor Server registers this port with the Preside Multiservice Data Manager context server. If you do not specify a port number, the server dynamically assigns an available port number.

`[-ndam <servicename@host>]` specifies the NDAM service name and host. If you do not specify a service name, the server tries to find an NDAM process on the local machine with the service name “NDAM”.

`[-h]` displays the parameters on the VPN Monitor Server start-up command.

Interdependencies

The VPN Monitor Server requires the following processes to be up and running:

- VPN Monitor Extractor
- NDAM
  The NDAM server must be started with the -F option to ensure component type filtering using fully qualified type specifications.

Exit codes

For exit codes produced by the VPN Monitor server, their meanings, and corrective action, see Table 85.

Table 85
Exit codes for VPN Monitor Server

Exit Code	Meaning and corrective action
10	Incompatible database schema version.
11	Extractor cannot connect to database.
12	No database connection established.
13	Lost database connection.
14	Cannot open a file.
(Sheet 1 of 2)	

Table 85 (Continued)
Exit codes for VPN Monitor Server

Exit Code	Meaning and corrective action
23	Communication problems connecting to a server.
50	Do not restart server.
51	Memory failure. Out of memory.
52	DIsk failure. Cannot find file.
53	Communication failure.
55	Invalid command argument or option
56	Configuration directory/file does not exist or is configured incorrectly.
58	Normal exit.
60	Port Number already in use.
62	Invalid license.
63	Invalid user ID.
(Sheet 2 of 2)	

Chapter 67
Session Servers

This section contains information on the session servers. See the following topics for more information:

- “About the session servers” (page 545)
- “Command Functional Process (CMCFUN)” (page 546)
- “Connection Manager (CM)” (page 547)
- “Generic Prober (GP)” (page 551)

About the session servers

The following servers are session servers:

- “Command Functional Process (CMCFUN)” (page 546)
- “Connection Manager (CM)” (page 547)
- “Generic Prober (GP)” (page 551)

Session servers are started automatically when the user logs on to a user account that is set up to run the default Preside Multiservice Data Manager user environment, or when a command macro uses the cmcwrap command. Session servers do not have an entry in the file /opt/MagellanNMS/cfg/SVMList.cfg. Session servers cannot be started by the Server Administration tool.

See the figure “Session Servers data flow diagram” (page 549) for a data flow diagram of the Session servers.

Command Functional Process (CMCFUN)

The CMCFUN server forwards operator commands to network elements for execution.

Error messages do not appear on the System Log Display; they are sent to the Command Console, RNCS or CMCCMD (CMC command). CMCCMD enables the operator commands to network elements to be sent using a macro.

For more information, see the following:

- "Interdependencies" (page 546)
- "Exit codes" (page 546)
- "Error messages" (page 547)

CMCFUN startup command

The CMCFUN server is usually started automatically as part of a Preside Multiservice Data Manager (MDM) user session (upon login for enabled users, when starting the MDM window), or when starting a remote network communication system (RNCS) session.

CMCFUN needs to be started manually to enable command macros and other non-user driven tasks. In this context, CMCFUN supports the following command line option:

```
    -t <minutes>
```

If the specified number of minutes expires with no active commands being processed, CMCFUN automatically terminates to free up the network resources and connections reserved for it.

Interdependencies

The CMCFUN server relies on the CM server.

Exit codes

Exit codes for the CMCFUN server are shown in the following table.

Table 86
Exit codes for the CMCFUN server

Exit code	Description
0	help invoked or explicitly terminated
51	not enough memory
53	could not register service (is another CMCFUN for this DISPLAY already running?)
59	could not initialize IPC environment (is MNSD running?)

Error messages

None; error messages are sent to the command console, RNCS, or the cmccmd program.

Connection Manager (CM)

The Connection Manager (CM) server manages all network connections originating from user sessions. The CM server establishes connections to network elements, and stores all connection and authentication data. The authentication information is shared within a user session.

The CM server establishes DPN connectivity through the NCSMGR server, and establishes Passport connectivity through the FDTM server.

Initial network information is obtained from the NCSMGR server for DPN and from the FDTM server for Passport.

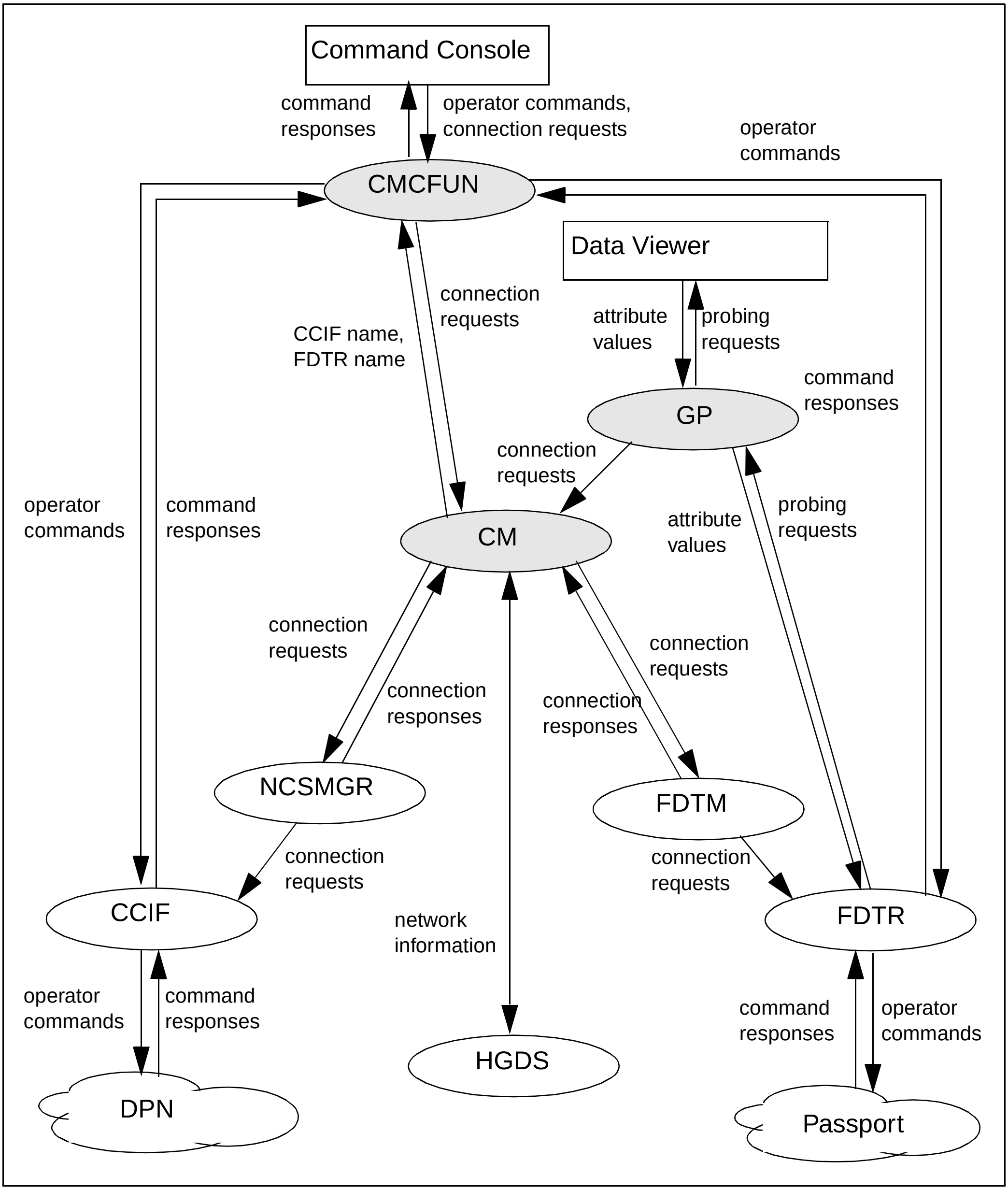
The FDTM server creates FDTR processes to communicate with a Passport node. The NCSMGR server creates CCIF processes to communicate with the DPN NCS.

When a network connection is established, the CM server returns the FDTR or CCIF service name to its CM client. The CM client can then use the services of the CCIF or FDTR to communicate with network elements.   An example of a CM client is the CMCFUN server, as shown in the figure “Session Servers data flow diagram” (page 549).

For more information, see the following:

- “CM startup command” (page 550)
- “Interdependencies” (page 550)
- “Connection Manager Server configuration” (page 550)
- “Exit codes” (page 550)
- “Error messages” (page 551)

Figure 57
Session Servers data flow diagram



CM startup command

The CM server is started automatically with the /opt/MagellanNMS/bin/nmssession file, when the user logs on.

For information on setting up UNIX accounts for Preside Multiservice Data Manager, see 241-6001-303 *Preside MDM Administrator Guide*.

The CM server is restarted if it exits during a user session. The CM server is terminated when the user logs out. The administrator may optionally configure the CM server so that authentications are not saved by the CM server.

Interdependencies

The CM server relies on the NCSMGR, FDTM and HGDS servers. All Preside Multiservice Data Manager provisioning tools, and the CMCFUN and Generic Prober servers rely on the CM server.

Connection Manager Server configuration

To configure the CM server so that authentications are not stored, do the following:

1 Log in as root.

2 Run the following command:

```
touch /opt/MagellanNMS/cfg/CMNoAuth.cfg
```

3 Set the file permissions to 644:

```
chmod 644 /opt/MagellanNMS/cfg/CMNoAuth.cfg
```

This only needs to be performed once per workstation, and will configure all future CM servers running on the workstation.

To revert to the default configuration so that authentications are stored, you enter the following command in a UNIX window:

```
rm /opt/MagellanNMS/cfg/CMNoAuth.cfg
```

Exit codes

Exit codes for the CM server are shown in the following table.

Table 87
Exit codes for the CM server

Exit code	Description
51	Not enough memory
59	Could not initialize IPC environment (is MNSD running?)
60	Could not register service (is another CMCFUN for this DISPLAY already running?)

Error messages

Error messages for the CM server are shown in the following table.

Table 88
Error messages for the CM server

Error message	Meaning and action
CM: Could not register service.	Is there another CM server running with the service name CM_DISPLAY (where display is the value of the DISPLAY environment variable)?”

Generic Prober (GP)

The GP server provides application specific real time probing of Passport component statistics. The client may specify the component name and attributes to probe, as well as the probing frequency. The GP server will set up a connection to the specified node and retrieve the specified attributes in the required frequency and returns the data to the client.

For more information, see the following:

- “Interdependencies” (page 551)
- “Exit codes” (page 552)
- “Error messages” (page 552)

Interdependencies

The GP server relies on the CM server, the HGDS server, and the FDTM/FDTR servers.

Exit codes

Exit does for the GP server are as follows:

Table 89
Exit codes for the GP server

Exit code	Description
0	not enough memory
51	help invoked
59	could not initialize the IPC environment
60	could not register service (already running?)

Error messages

Error messages for the GP server are shown in the following table.

Table 90
Error messages for the GP server

Error message	Meaning and action
Generic Prober 9132 : prober %s error.	Recreate the prober.
Generic Prober 9139 : disconnect to network node %s failed.	Leave an orphan FMIP connection.
Generic Prober 9152 : server cannot access anymore memory.	Restart Generic Prober.
Generic Prober 9154 : server lost an FDTR.	The corresponding probers need to be recreated.
Generic Prober 9161 : server lost connection to HGDS.	No prober is created before auto reconnect.
Generic Prober 9171 : server lost connection to CM.	No prober is created before auto reconnect.

Chapter 68
Servers controlled by HP OpenView

This section contains information on the HP OpenView servers: Open View Alarm Translator (OVAT) and OpenView Data Access Mediator (OVDAM). See the following topics for more information:

- "Open View Alarm Translator (OVAT)" (page 554)
- "Open View Data Access Mediator (OVDAM)" (page 564)

Open View Alarm Translator (OVAT)

The OVAT provides trap information from nodes to client applications.

The OVAT server performs the following functions:

- sends requests for alarm information to a Network Access Data Mediator (NDAM) server on a Preside Multiservice Data Manager workstation
- translates the received alarms into trap format
- sends the traps to the PostMaster Daemon (PMD) for broadcasting to registered HP OpenView applications

See the following sections for more information on the OVAT server.

- “Data flow” (page 554)
- “Interdependencies” (page 554)
- “Configuration” (page 554)
- “Exit codes” (page 560)
- “OVAT logs” (page 560)

Data flow

OVAT receives alarm information from NDAM, translates it to traps, and sends the trap information to PMD. See the figure “OVDAM server data flow diagram” (page 566) for an illustration on the flow of data to and from OVAT.

Interdependencies

To provide HP OpenView applications with trap information, the OVAT server requires access to the following servers:

- NDAM
- PMD

Configuration

See the following sections for information on the OVAT configuration files:

- “Local registration file” (page 555)
- “Runtime configuration file” (page 555)

Local registration file

The installation process creates and installs the local registration file using the path */etc/opt/OV/share/lrf/nortelOvat.lrf*. Do not modify this file.

Each daemon process running under HP OpenView and controlled by the OpenView Supervisor process Management Daemon (OVSPMD) requires this local registration file.

Runtime configuration file

OVAT does not support command line parameters, however, you can specify runtime options in the runtime configuration file. To modify the runtime configuration, edit the file */etc/opt/OV/share/conf/C/nortel/nms/odn/ovat.cfg*.

You can make the following modifications to the runtime configuration file by specifying

- runtime options
- alternate NDAM servers
- component types filters for Preside Multiservice Data Manager (MDM) data
- device set filters for MDM data
- exclusions to device set and component types filters
- filters to include in the CREATE sieve request for alarms

Although the order of changes in the runtime configuration file does not matter, it is recommended that all lines pertaining to one attribute be grouped together for ease of use.

Runtime options

You can modify the following runtime options.

- logLevel
  LogLevel specifies the type of information to save in a log file. By default, FATAL, SNO, ERRORS, and MAJOR levels are logged in the file */var/opt/OV/share/log/ovat.trace*. To modify the logLevel, use the following command:

  ```
  logLevel: <list of levels>
  ```

  where:

  `<list of levels>` is at least one of the following:

  `FATAL`
  logs errors fatal to OVAT operation

  `SNO`
  (should not occur) logs errors caused by unexpected conditions

  `ERRORS`
  logs all other errors

  `MAJOR`
  logs major events

  `MINOR`
  logs minor events

  `INFO`
  logs information messages

  `TRACE`
  logs functional call tracing

  `ALL`
  logs all of the above message levels

- interactiveLog
  InteractiveLog directs the log information to standard output rather than a log file. By default, the interactiveLog option is off. To modify the interactiveLog option use the following command:

  ```
  interactiveLog: <on|off>
  ```

  where:

  `<on>` turns on the interactiveLog option.

  `<off>` turns off the interactiveLog option.

Alternate NDAM servers

You can specify alternate NDAM servers. If the primary NDAM server incurs a connection failure, OVDAT cycles through the list of alternate NDAM servers to find one that accepts the connection request. To specify an alternate NDAM server, use the following command:

```
ndamServer: <name>@<host>
```

where:

<`name`> is the name of the server to act as an alternate to NDAM.

<`host`> is the location of the server specified by a host name or an IP address.

Component type filters

Component type filters specify which typesets to use when filtering Preside Multiservice Data Manager (MDM) data from NDAM to OVDAM. The typeset name is included during OVDAM registration with NDAM. Typesets are defined in the NDAM typeset configuration file (for details, see “Typeset configuration files” (page 269)). To specify a typeset filter, use the following command:

```
compTypeSet: <typeset name>
```

where:

<`typeset name`> is the name of a set of device types.

You can exclude some components from the set defined by the compTypeSet line by using the following command:

```
compExcluded: <NMS component type>
```

where:

<MDM component type> is at least one of the following:

- `<device type>`
  excludes from the typeset all devices of the specified type. For example, if you specify a device type of EM, all Passport module data are excluded.

- `<device type>-<last subcomponent type>`
  excludes from the typeset all subcomponents of the specified device and subcomponent type. For example, if you specify EM-DS1, all DS1 subcomponents data for all EM devices are excluded.

- `<device type>-*`
  excludes from the typeset all subcomponents of the specified device type. For example, if you specify EM-*, data for all subcomponents of EMs are excluded.

- `<link type>:`
  excludes from the typeset the specified link type. For example, if you specify PTK:, data for all links of type PTK are excluded

- *:
  excludes from the typeset all link types.

Device set filters

Device set filters specify which device sets to use when filtering Preside Multiservice Data Manager data from NDAM to OVDAM. The device set name is included during OVDAM registration with NDAM. Device sets are defined in the NDAM device set configuration file (for details, see “Deviceset configuration files” (page 273). To specify a device set filter, use the following command:

```
deviceSet: <device set name>
```

where:

`<device set name>` is the name of a set of devices.

You can exclude some devices from the set defined by the deviceSet line by using the following command:

```
deviceExcluded: <excluded device>
```

where:

`<excluded device>` is at least one of the following:

- `<device type> *`
  excludes from the device set all devices of the specified type
- `<device type> <name prefix>*`
  excludes from the device set all devices of the specified type whose name starts with the specified name prefix
- `<device type <device name>`
  excludes from the device set the specified device type and device name

Filters

You can specify filters to include in the CREATE serve request for alarms. To do so, use the following command:

```
filter: <attribute> <operator> <type> <value>
```

where:

`<attribute>` is the name of an attribute.

`<operator>` is a valid operator for filtering.

`<type>` is the attribute value type.

`<value>` is the attribute value.

The filter line causes OVAT to add the following line to the CREATE request sent to NDAM:

```
_attr: evenFilter SS <attribute> <operator> <type> <value>
```

See 241-6001-203 *Preside MDM Alarm and Status API Reference Guide* for expressions that you can use as filters.

Exit codes

Exit codes for the OVAT server are shown in the following table.

Table 91
Exit codes for the OVAT server

Exit code	Description
1	Registration with OVSPMD failed
2	Registration with MNSD failed
3	Software error
4	Termination required by OVSPMD
5	Memory allocation error
6	Configuration file access error

OVAT logs

Certain errors and events cause OVAT to issues logs. Log messages are written to */var/opt/OV/share/log/ovat.trace*. Each log message is prefaced by one of the following log levels:

- FATAL
  An error situation prevents OVAT from functioning. A FATAL log is always followed by OVAT termination. For details about OVAT FATAL logs, see “FATAL logs issued by the OVAT server” (page 561).
- SNO
  An error that should not occur (possibly a software error). The error is logged and, if possible, OVAT continues. Report these errors if the recur.
- ERRORS
  An error situation that is likely due to invalid configuration or input. For details about OVAT ERRORS logs, see “ERRORS logs issued by the OVAT server” (page 561).
- MAJOR
  OVAT is signaling the occurrence of an important event (not usually an error). For details about OVAT MAJOR logs, see “MAJOR logs issued by the OVAT server” (page 563).

The following table details FATAL logs.

Table 92
FATAL logs issued by the OVAT server

FATAL Log message	Description
Failed to register with OVSPMD	OVAT registration with OVSPMD failed. The ovstop and ovstart utilities should be used to restart all the workstation daemons.
Invalid API dictionary config file	OVAT failed to process the configuration file /opt/OV/nortel//nms/odn/ovcfg/conf/ovatDict.cfg. Verify that the file exists and that it and the directories on its path have the required access permission.
Invalid runtime config file	The file ovat.cfg contains invalid entries. Additional details may be available in ERRORS logs. Correct the runtime configuration file.
Memory allocation failed	OVAT cannot allocate memory for some data structure. Review the workstation engineering.

The following table details ERRORS logs.

Table 93
ERRORS logs issued by the OVAT server

ERRORS Log message	Description
Alarm sieve creation failed on <NDAM name>; error code: <code>	NDAM rejected OVAT attempt to create a sieve for alarms notifications. OVAT disconnects from this server and tries the next server in the list in 15 seconds.
Cannot add trap variable, memory allocation failure	A request to add a variable to an event notification has been rejected due to memory allocation failure. If possible, terminate some applications to increase available memory.
(Sheet 1 of 3)	

Table 93 (Continued)
ERRORS logs issued by the OVAT server

ERRORS Log message	Description
Cannot create notification, interface not connected	OVAT is not currently connected to PMD and cannot create an event notification. OVAT continues to process NDAM input and tries to establish the connection.
Cannot create notification, memory allocation failure	A request to create an event notification has been rejected due to memory allocation failure. If possible, terminate some applications to increase available memory.
Cannot open the log file: <log file name>	The logging interface cannot open the log file. Some directory on the path to the log file probably does not have the correct permissions set. Correct and restart OVAT.
Cannot open session for SNMP receive interface	Connection with PMD cannot be established. OVAT cannot receive messages. OVAT continues to process NDAM input and tries to establish the connection.
Cannot open session for SNMP send interface	Connection with PMD cannot be established. OVAT cannot send event notifications. OVAT continues to process NDAM input and tries to establish the connection.
Connection attempt rejected to <NDAM name>	NDAM rejected OVAT attempt to connect. OVAT tries the next server in the list in 15 seconds.
Failure to send: <reason>	An event notification cannot be sent.
Invalid line (<line number>) in runtime config	The contents of the identified line in the ovat.cfg runtime configuration file is invalid. Correct and restart OVAT.
Invalid NDAM server specification: <specification>	An ndamServer specification in ovat.cfg has an invalid value. Correct and restart OVAT.
Lost connection with NDAM server <NDAM name>	OVAT lost connection with the identified server. OVAT tries to connect to the next server in the list in 15 minutes.
No NDAM server defined	There is no valid ndamServer specification in ovat.cfg. Correct and restart OVAT.
(Sheet 2 of 3)	

Table 93 (Continued)
ERRORS logs issued by the OVAT server

ERRORS Log message	Description
Registration with <NDAM name> failed; error code: <code>	OVAT attempt to register with the identified server failed. OVAT tries the next server in the list in 15 seconds.
SNMP receive interface connection lost	Connection with PMD has been lost; OVAT cannot receive messages. OVAT continues to process NDAM input and tries to reestablish the connection.
SNMP send interface connection lost	Connection with PMD has been lost; OVAT cannot send event notifications. OVAT continues to process NDAM input and tries to reestablish the connection.
(Sheet 3 of 3)	

The following table details the MAJOR logs.

Table 94
MAJOR logs issued by the OVAT server

MaJOR Log message	Description
Registration with <NDAM name> completed	OVAT has successfully registered with the identified NDAM server.
No device set defined	There is no device set defined in ovat.cfg (this is not necessarily an error). NDAM forwards information about all the devices in the network model.
No component type est defined	There is no component set defined in ovat.cfg (this is not necessarily an error). For each device, NDAM forwards information about all the component in the network model.

Open View Data Access Mediator (OVDAM)

The OVDAM server provides Preside Multiservice Data Manager (MDM) management data to applications running on a Hewlett-Packard (HP) OpenView platform.

The OVDAM server performs the following functions:

- sends requests for node and link information to a Network Data Access Mediator (NDAM) server on an MDM workstation
- translates the node and link data received from NDAM to a format suitable for the HP OpenView database (OVDB). For OVDAM to distinguish the different types of nodes, it retrieves network model data from NDAM. Two Network Model attributes, PP_TYPE and PP_SHELF_TYPE are used by the OVDAM server to identify the type of node. This enables the display of unique icons on the OpenView submap to differentiate between nodes belonging to different series of Nortel Networks network elements.
- sends the translated data to OVDB
- sends notifications about object creation, deletion, and modification to the OpenView PostMaster Daemon (PMD) which, in turn, broadcasts this information to registered applications
- creates the OpenView Simple Network Management Protocol (SNMP) configuration database entries which are used to enable SNMP access to monitored devices

See the following sections for more information on the OVDAM server.

- “Data flow” (page 565)
- “Interdependencies” (page 567)
- “Configuration” (page 567)
- “State propagation” (page 578)
- “Exit codes” (page 578)
- “OVDAM logs” (page 579)

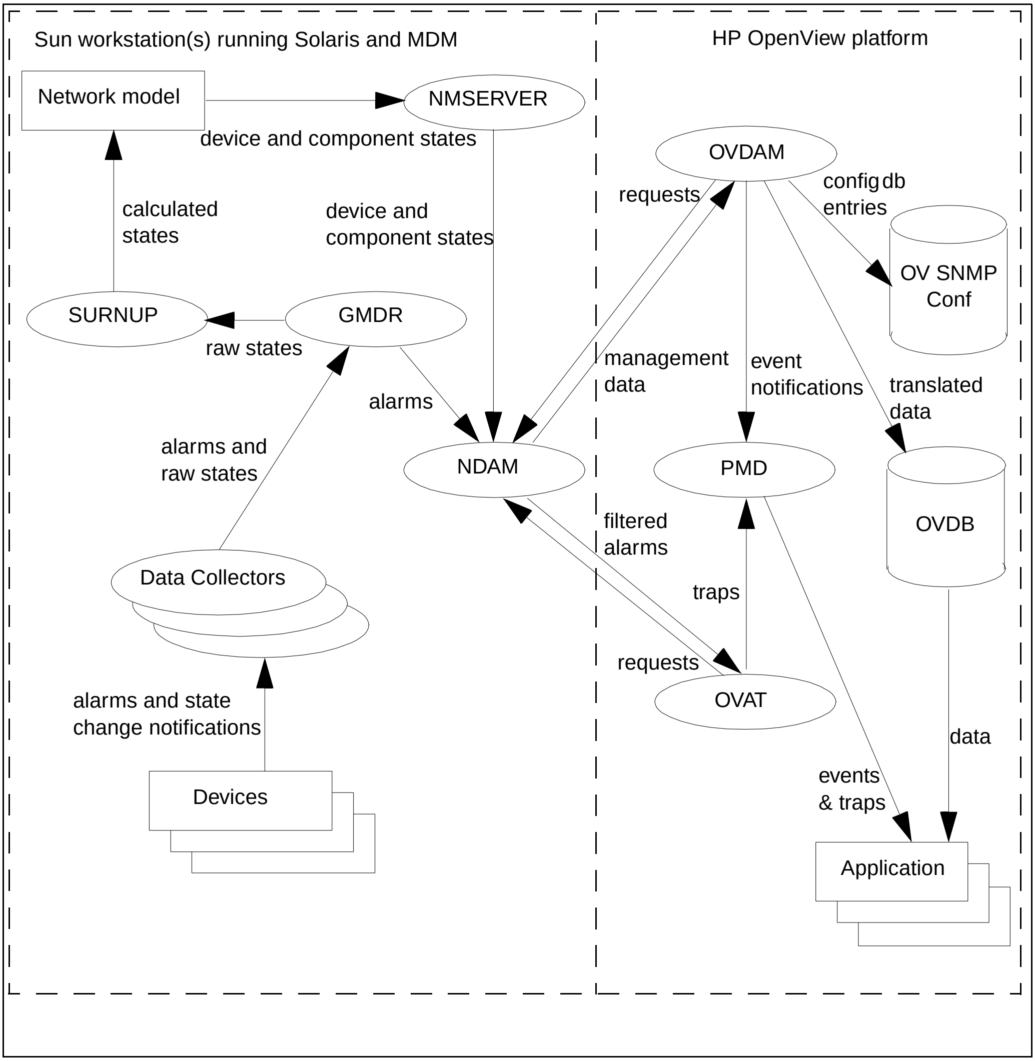
Data flow

OVDAM obtains node management data from an NDAM server that runs on a Preside Multiservice Data Manager (MDM) workstation. The figure “OVDAM server data flow diagram” (page 566) illustrates the data flow through the MDM and HP OpenView systems. For details about the data flow, see “Network Data Access Mediator (NDAM)” (page 263).

OVDAM requests node and link information from NDAM, translates the information, and sends the translated data to OVDB for storage. In addition, OVDAM sends event notifications to PMD which, in turn, delivers these notifications to registered applications.

HP OpenView applications do not interact directly with OVDAM. For applications to receive information from OVDAM, you need to configure OVDAM to create the fields required by the HP OpenView applications. Once OVDAM creates and stores these fields in OVDB, the information becomes available to HP OpenView applications.

Figure 58
OVDAM server data flow diagram



Interdependencies

To provide HP OpenView applications with management data, the OVDAM server requires access to the following servers:

- NDAM
- PMD

OVDAM also requires access to the following databases:

- OVDB
- OV SNMP configuration database

Configuration

See the following sections for information on the OVDAM configuration files:

- “Local registration file” (page 567)
- “Fields registration file” (page 568)
- “Runtime configuration file” (page 568)
- “Data translation configuration file” (page 573)

You cannot modify all of the OVDAM configuration files. You can modify runtime configuration and data translation files, but you must not modify the registration files.

Although command line parameters are not supported for OVDAM, you can specify runtime options in the runtime configuration file.

Local registration file

The installation process creates and installs the local registration file using the path /etc/opt/OV/share/lrf/nortelOvdam.lrf. Do not modify this file.

Each daemon process running under HP OpenView and controlled by the OpenView Supervisor Process Management Daemon (OVSPMD) requires this local registration file.

Fields registration file

The installation process creates and installs the fields registration file in the file */etc/opt/OV/share/fields/C/nortelOvdam.fields*. This registration file contains instructions for registering fields. Do not modify this file.

The fields registration file registers with OVDB those fields created by OVDAM for OVDAM's own use. Registration of fields for use with other applications must be done by those applications. If other applications do not register the fields, then OVDAM is not able to create them.

Runtime configuration file

OVDAM does not support command line parameters, however, you can specify runtime options in the runtime configuration file. To modify the runtime configuration, edit the file */etc/opt/OV/share/conf/C/nortel/nms/odn/ovdam.cfg*.

You can make the following modifications to the runtime configuration file by specifying

- runtime options
- alternate NDAM servers
- component types filters for Preside Multiservice Data Manager (MDM) data
- device set filters for MDM data
- exclusions to device set and component types filters

Although the order of changes in the runtime configuration file does not matter, it is recommended that all lines pertaining to one attribute be grouped together for ease of use.

Runtime options

You can modify the following runtime options.

- logLevel
  LogLevel specifies the type of information to save in a log file. By default, FATAL, SNO, ERRORS, and MAJOR levels are logged in the file */var/opt/OV/share/log/ovdam.trace*. To modify the logLevel, use the following command:

```
logLevel: <list of levels>
```

where:

`<list of levels>` is at least one of the following:

`FATAL`
logs errors fatal to OVDAM operation

`SNO`
(should not occur) logs errors caused by unexpected conditions

`ERRORS`
logs all other errors

`MAJOR`
logs major events

`MINOR`
logs minor events

`INFO`
logs information messages

`TRACE`
logs functional call tracing

`ALL`
logs all of the above message levels

- interactiveLog
  InteractiveLog directs the log information to standard output rather than a log file. By default, the interactiveLog option is off. To modify the interactiveLog option use the following command:

```
interactiveLog: <on|off>
```

where:

`<on>` turns on the interactiveLog option

`<off>` turns off the interactiveLog option

- stateSelection
  StateSelection specifies the MDM state to be used as a default when generating the corresponding OVDB object state. If needed, you can override the default for a specified MDM node type in the data translation configuration file (see “Data translation configuration file” (page 573)). By default, the propagated state is used for top level objects such as nodes and links and the raw state is used for all other objects. To modify the stateSelection option, use the following command:

  ```
  stateSelection <default|raw|propagated>
  ```

  where:

  `<default>   uses the propagated state for top level objects and the raw state for all other objects`

  `<raw>` uses the raw state for all objects

  `<propagated>` uses the propagated state for all objects

- restartDelay
  RestartDelay specifies the length of time OVDAM waits before deleting nodes that were created by an OVDAM process and that have a state IGNORED_STATE after an OVDAM restart, a network model change, or a connection to a new NDAM. By default, OVDAM waits 90 minutes. To modify the restartDelay option, use the following command:

  ```
  restartDelay: <duration of delay>
  ```

  where:

  <`duration of delay`> is the length of time in minutes that OVDAM waits before deleting IGNORED_STATE nodes

Alternate NDAM servers

You can specify alternate NDAM servers. If the primary NDAM server incurs a connection failure, OVDAM cycles through the list of alternate NDAM servers to find one that accepts the connection request. To specify an alternate NDAM server, use the following command:

```
ndamServer: <name>@<host>
```

where:

<name> is the name of the server to act as an alternate to NDAM

<host> is the location of the server specified by a host name or an IP address

Component type filters

Component type filters specify which typesets to use when filtering Preside Multiservice Data Manager (MDM) data from NDAM to OVDAM. The typeset name is included during OVDAM registration with NDAM. Typesets are defined in the NDAM typeset configuration file (for details, see “Typeset configuration files” (page 269)). By default, the compTypeSet specified in the ovdam.cfg file is PP_Mod_and_Link. This setting restricts management data to the node and link level to preserve resources. If needed, you can modify this setting, but first evaluate how much data you require. Additional requirements may cause performace to degrade. To specify a typeset, use the following command:

```
compTypeSet: <typeset name>
```

where:

<typeset name> is the name of a set of device types.

You can exclude some components from the set defined by the compTypeSet command by using the following command:

```
compExcluded: <NMS component type>
```

where:

<MDM component type> is at least one of the following:

- `<device type>`
  excludes from the typeset all devices of the specified type. For example, if you specify a device type of EM, all Passport module data are excluded.

- `<device type>-<last subcomponent type>`
  excludes from the typeset all subcomponents of the specified device and subcomponent type. For example, if you specify EM-DS1, all DS1 subcomponents data for all EM devices are excluded. This option is not applicable if only rnodes and links are being modelled.

- `<device type>-*`
  excludes from the typeset all subcomponents of the specified device type. For example, if you specify EM-*, data for all subcomponents of EMs are excluded. This option is not applicable if only nodes and links are being modelled.

- `<link type>:`
  excludes from the typeset the specified link type. For example, if you specify PTK:, data for all links of type PTK are excluded.

- *:
  excludes from the typeset all link types

Device set filters

Device set filters specify which device sets to use when filtering Preside Multiservice Data Manager data from NDAM to OVDAM. The device set name is included during OVDAM registration with NDAM. Device sets are defined in the NDAM device set configuration file (for details, see “Deviceset configuration files” (page 273)). To specify a device set filter, use the following command:

```
deviceSet: <device set name>
```

where:

`<device set name>` is the name of a set of devices

You can exclude some devices from the set defined by the deviceSet command by using the following exclude command:

```
deviceExcluded: <excluded device>
```

where:

`<excluded device>` is at least one of the following:

- `<device type> *`
  excludes from the device set all devices of the specified type
- `<device type> <name prefix>*`
  excludes from the device set all devices of the specified type whose name starts with the specified name prefix
- `<device type <device name>`
  excludes from the device set the specified device type and device name

Data translation configuration file

The data translation configuration files contain two types of records. One type of record is the object correspondence record which specifies the OVDB object class that corresponds to an Preside Multiservice Data Manager (MDM) network model node or link type. Another type of record is the object translation record which specifies the fields contained in a created OVDB object and how they are obtained. You only need to modify this configuration file if, for some object types, you want to generate the OVDB state using a different MDM state than that selected by default.

Data translation configuration files have a .tcf extension and are found under the following paths:

- /opt/OV/nortel/nms/odn/ovcfg/conf
  The .tcf files in this path are defined by the system. Do not modify.
- /etc/opt/OV/share/conf/C/nortel/nms/odn
  To modify data translation configuration files, use the .tcf files in this path.

  You define each record by a set of consecutive lines. Separate each record with a blank line. Each record line defines an attribute of the record and has the following format:

```
<attribute type>: <attribute definition>
```

The following example shows two records. The first record is an object correspondence record and assigns the MDM node type EM-DS1 to OVDB object class DS1. The second record is an object translation record and specifies the two fields in the OVDB class Passport DS1 and how they are obtained.

Example:

nodeType: EM-DS1
objectClass: Passport:DS1

objectClass: Passport DS1
fieldDef: isNortelMomsElement B F TRUE
fieldDef: isNortelMomsObjectManaged B F TRUE

Object correspondence record

The object correspondence record creates a correspondence between a Preside Multiservice Data Manager (MDM) Network Model node or link type and an OVDB object class. This record consists of the following lines:

- nodeType (mandatory)

  You must include the MDM object type specification line to define the MDM network node or link type. This specification line has the following format:

  `nodeType: <node_type>` or `linkType: <link_type>`

  where:

  <`node_type`> for a device is the first token of its component identifier. For a subcomponent, the node type is created from the first and last component category tokens from the component identifier separated by a dash (-). For example, the node type of the subcomponent EM NODEX LP 2 DS1 0 is EM-DS1.

  <`link_type`> for a link is the first token of its component identifier (that is, the token before the first colon (:)).

- objectClass (mandatory)

  You must include an OVDB object class specification line to define the class of the OVDB object. This specification line has the following format:

  ```
  objectClass: <OVDB class>
  ```

  where:

  `<OVDB class>` is any character string not beginning with a blank or tab character. OVDB classes are defined by the types of devices in HP OpenView.

- stateSelection (optional)

  The state selection line is optional. Use this line to specify which MDM state is to be used to determine the OVDB object state. If you do not specify this option, then the OVDB state is derived from the MDM default state selection or from the state selection specified in the runtime configuration file. The state selection line has the following format:

  ```
  stateSelection: [raw | propagated]
  ```

  If more than one object correspondence record is found with the same MDM object type specification (for example, if there is more than one data translation file), only the last specification applies.

  If OVDAM receives information for an MDM object that has no object correspondence records, OVDAM discards the information.

Object translation record

The object translation record specifies the fields in a created OVDB object and how these fields are created. There are two parts to this record. The first part specifies the OVDB object class. The second part defines the fields to be created and how the fields’ values are obtained.

The object class line establishes a link between one object translation record and one or more object correspondence records. This line has the same format as the object correspondence record’s objectClass line:

```
objectClass: <OVDB class>
```

where:

<`OVDB class`> is any character string not beginning with a blank or tab character. OVDB classes are defined by the types of devices in HP OpenView.

A field definition line defines one field to be created in the OVDB for an object of the specified class. Specify a field definition line for each field that needs to be created in OVDB. The field definition line has the following format:

```
fieldDef: <field_name> <value_expr>
```

where:

<`field_name`> specifies the name of the OVDB field. If the field name contains spaces, enclose the name in double quotes (“)

<`field_type`> specifies the type of field as B (Boolean), E (enumeration), I (integer), or S (string). If you use the enumeration field type, the value_expr must produce an integer. The integer is then matched with the corresponding constant in the enumeration.

<`value_expr`> is an expression that specifies how to obtain the field value. This expression has a token that indicates a retrieval mode followed by the information needed to obtain the value. You can abbreviate the retrieval mode up to a single letter. The retrieval modes and their required information are as follows:

- P (property) followed by the property name
  The value of the property is obtained by sending NDAM a GetCompInfo action request for the specified property. If the value of the property cannot be obtained from NDAM, the OVDB field is not created.
- F (fixed) followed by the fixed value
  Specify the fixed value.

- I (identifier) followed by a substring of the Preside Multiservice Data Manager (MDM) component identifier
  Use the following format:

  ```
  identifier (CID|EP1|EP2|[<category>]
  ```

  where:
  `CID` is the value extracted from the full component identifier.

  `EP1` and `EP2` are the values of the first and second link endpoints.

  <`category`> is the MDM component category. If category is specified, the value includes all the category/value pairs up to and including the specified category. If the specified category does not exist in the selected identifier substring, the complete substring is used. For example, if the component identifier is NL:EM NODE21 DPNGATE 8:PM A1234 PE 4 PI 4 PO 1 and the identifier retrieval expression is I EP2 PE, then the returned value is PM A1234 PE 4.

Some fields are created automatically so you do not need to define them. For example, when OVDAM creates an object in OVDB, the following fields are automatically created:

- OVDB selection name (same as the component identifier)
- object class (defined by the first line of the record)
- parent object
- device name
- MDM translated raw state
- MDM translated propagated state
- OVDB object state

Data translation records are used to create object fields in the following circumstances:

- object creation
  When a new OVDB object is created, queries are sent for the fields with a retrieval mode of property. The other fields are immediately created.

- object recreation
  If a state change notification is received for an OVDB object currently in the state IGNORED_STATE, then its fields are recreated.
- OVDB refresh
  When OVDAM receives an HUP signal, queries are sent to refresh the contents of all fields with a retrieval mode of property.

State propagation

Although both Preside Multiservice Data Manager (MDM) and HP OpenView compute object states based on the raw state of their components or endpoints, they use different algorithms. MDM propagates the state of subcomponents to parent components until the state of the device is established. As a result, every component has a raw state based on active alarms and a propagated state based on the component's raw state and the propagated state of the subcomponents. HP OpenView computes the state of an object from the states of those objects that are included in the submap and that are owned by that object.

Since methodologies differ, MDM and HP OpenView propagated states can differ. It is recommended that you can make configuration changes that allow all devices, links and components to show a state compatible with the MDM propagated state. To do so, make the following modifications:

- turn off state propagation in HP OpenView applications (submap symbols)
- use the OVDAM stateSelection option with MDM propagated states to define OVDB object states

If you cannot turn off HP OpenView state propagation for reasons of compatibility with data collected from other sources, it is best to use OVDAM default MDM state selection (raw state for components; propagated state for devices and links). In this case, only devices can have submaps and state propagation must be turned off for links.

Exit codes

Exit codes for the OVDAM server are shown in the following table.

Table 95
Exit codes for the OVDAM server

Exit code	Description
1	Registration with OVSPMD failed
2	Registration with MNSD failed
3	Software error
4	Termination required by OVSPMD
5	Memory allocation error
6	Configuration file access error
7	OVDB connection failed

OVDAM logs

Certain errors and events cause OVDAM to issues logs. Log messages are written to the file */var/opt/OV/share/log/ovdam.trace*. Each log message is prefaced by one of the following log levels:

- FATAL
  An error prevents OVDAM from functioning. A FATAL log is always followed by OVDAM termination. For details about OVDAM FATAL logs, see “FATAL logs issued by the OVDAM server” (page 580).

- SNO
  An error that should not occur (possibly a software error). The error is logged and, if possible, OVDAM continues. Report these errors if they recur.

- ERRORS
  An error that is likely due to invalid configuration or input. For details about OVDAM ERRORS logs, see “ERRORS logs issued by the OVDAM server” (page 581).

- MAJOR
  OVDAM is signaling the occurrence of an important event (not usually an error). For details about OVDAM MAJOR logs, see “MAJOR logs issued by the OVDAM server” (page 584).

The following table details OVDAM FATAL logs.

Table 96
FATAL logs issued by the OVDAM server

FATAL Log message	Description
Failed to find DB <field name> field	OVDAM cannot retrieve a OVDB basic field definition. Reload the field definitions in OVDB. If the problem persists, report it since part of the installation has not been done correctly.
Failed to register with OVSPMD	OVDAM registration with OVSPMD failed. The ovstop and ovstart utilities should be used to restart all the workstation daemons.
Invalid runtime config file	The file ovdam.cfg contains invalid entries. Additional details may be available in ERRORS logs. Correct the runtime configuration file.
Invalid translation config data	OVDAM did not find any valid translation configuration file (.tcf). Verify the existence and the accessibility of those fields and restart OVDAM.
IPC initialization failed	OVDAM registration with MNSD failed. Either NMSD is not running or another OVDAM process (started from the command line) is already running. Correct the problem and restart OVDAM.
Memory allocation failed	OVDAM cannot allocate memory for some data structure. Review the workstation engineering.

The following table details ERRORS logs.

Table 97
ERRORS logs issued by the OVDAM server

ERRORS Log message	Description
Cannot add trap variable, memory allocation failure	A request to add a variable to an event notification has been rejected due to memory allocation failure. If possible, terminate some applications to increase available memory.
Cannot create notification, interface not connected	OVDAM is not currently connected to PMD and cannot create an event notification. OVDAM continues to process NDAM input and tries to establish the connection.
Cannot create notification, memory allocation failure	A request to create an event notification has been rejected due to memory allocation failure. If possible, terminate some applications to increase available memory.
Cannot extract endpoint for <component name>	The identified component is not a link, therefore the endpoint name required by the configured translation does not exist.
Cannot open data translation file <file name>	OVDAM cannot open the specified data translation file. Verify the access permissions associated with this file and the directories in its path.
Cannot open the log file: <log file name>	The logging interface cannot open the log file. Some directory on the path to the log file probably does not have the correct permissions set. Correct and restart OVDAM.
Cannot open session for SNMP receive interface	Connection with PMD cannot be established. OVDAM cannot receive messages. OVDAM continues to process NDAM input and tries to establish the connection.
Cannot open session for SNMP send interface	Connection with PMD cannot be established. OVDAM cannot send event notifications. OVDAM continues to process NDAM input and tries to establish the connection.
Connection attempt rejected to <NDAM name>	NDAM rejected OVDAM attempt to connect. OVDAM tries the next server in the list in 15 seconds.
(Sheet 1 of 3)	

Table 97 (Continued)
ERRORS logs issued by the OVDAM server

ERRORS Log message	Description
Failure to send: <reason>	An event notification cannot be sent.
Field type does not match line <line number> in mediation config file	The OVDB field name and type specified in the identified line of the .tcf configuration file currently processed do not match those registered with OVDB.
Invalid default state selection: <selection>	The stateSelection option in ovdam.cfg has an invalid value. Correct and restart OVDAM.
Invalid line <line number> in mediation configuration file	The contents of the identified line in the .tcf configuration file currently processed is invalid. Correct and restart OVDAM.
Invalid line (<line number>) in runtime config	The contents of the identified line in the ovdam.cfg runtime configuration file is invalid. Correct and restart OVDAM.
Invalid NDAM server specification: <specification>	An ndam Server specification in ovdam.cfg has an invalid value. Correct and restart OVDAM.
Lost connection with NDAM server <NDAM name>	OVDAM lost connection with the identified server. OVDAM tries to connect to the next server in the list in 15 minutes.
Network change sieve creation failed on <NDAM name>: error code: <code>	NDAM rejected OVDAM attempt to create a sieve for network change notifications. OVDAM will disconnect from this NDAM server and try the next NDAM server in the list in 15 seconds.
No database object translation record found	OVDAM has found no valid object translation record (listing the fields to include in a given OVDB object) in the .tcf configuration files read; a FATAL log is also issued and OVDAM terminates.
No NDAM server defined	There is no valid ndamServer specification in ovdam.cfg. Correct and restart OVDAM.
No object correspondence record found	OVDAM has found no valid object correspondence record (associating an Preside Multiservice Data Manager object type with an OVDB object type) in the .tcf configuration files read; a FATAL log is also issued and OVDAM terminates.
(Sheet 2 of 3)	

Table 97 (Continued)
ERRORS logs issued by the OVDAM server

ERRORS Log message	Description
OpiDbInterf::<procedure name>: <reason>	Invalid call to a procedure of the OVDB interface. This is usually due to a configuration error for some field of the database object (for example, the use of an unregistered field name). A separate log was probably issued when the configuration file was read.
Registration with <NDAM name> failed; error code: <code>	OVDAM attempt to register with the identified server failed. OVDAM tries the next server in the list in 15 seconds.
Read error on mediation configuration file, line <line number>	A read error has occurred trying to read the specified line in a .tcf configuration file.
SNMP receive interface connection lost	Connection with PMD has been lost; OVDAM cannot receive messages. OVDAM continues to process NDAM input and tries to reestablish the connection.
SNMP send interface connection lost	Connection with PMD has been lost; OVDAM cannot send event notifications. OVDAM continues to process NDAM input and tries to reestablish the connection.
State change sieve creation failed on <NDAM name>; error code: <code>	NDAM rejected OVDAM attempt to create a sieve for state change notifications. OVDAM tries the next server in the list in 15 seconds.
Synchronize action failed on <NDAM name>; error code: <code>	NDAM rejected OVDAM synchronization request. OVDAM disconnects from this NDAM server and tries the next server in the list in 15 seconds.
(Sheet 3 of 3)	

The following table details the MAJOR logs.

Table 98
MAJOR logs issued by the OVDAM server

MaJOR Log message	Description
Reading config file <file name>	OVDAM is starting to process the contents of the identified file. Therefore, all subsequent ERRORS logs about configuration file errors apply to this file.
Registration with <NDAM name> completed	OVDAM has successfully registered with the identified NDAM server.
No device set defined	There is no device set defined in ovdam.cfg (this is not necessarily an error). NDAM forwards information about all the devices in the network model.
No component typeset defined	There is no component set defined in ovdam.cfg (this is not necessarily an error). For each device, NDAM forwards information about all the component in the network model.

Appendix A
Server ports

Refer to the following for server and port information:

- “MDM server and bind ports” (page 585)

MDM server and bind ports

The following table lists Preside Multiservice Data Manager (MDM) servers, the ports that they bind to, and whether TCP or UDP is used as the transport protocol. The table also indicates other servers that the server communicates with and the port number used to make that connection. If a bind port is not listed for a server, that means that server binds to a dynamic port.

For information on Operator Client ports, refer to NN10600-607 *Passport - MDM Network Security: Secure Communications Configuration*.

Table 99
MDM servers and bind ports

Server	Bind port	Transport protocol (TCP or UDP	Connections to other servers
backup controller (nsctlbck)	5000	TCP	
backup provider (pbckpp)	5020	TCP	FTP on port 20 (TCP) FTP on port 21 (TCP)
backup server (nsctlbck-notification)	5000 5050	TCP TCP	

Table 99
MDM servers and bind ports

Server	Bind port	Transport protocol (TCP or UDP	Connections to other servers
configuration manager (configman)	6767	TCP	pcserver on 6760 (TCP)
MDM context server (ctxsvr)			
dbnl auto-disabling daemon (dbnlwatch)			
data collection daemon (DCD)	161	UDP	
data management agent (dma)			
DPN management data router (dmdr)			
network model coordinator (dnmnmc)			
end-to-end server (eteserver)	6600	TCP	
network model editing server (edserver)			
Passport communication manager (FDTM)			Passport FMIP on 5928 (TCP)
Passport management data router (FMDR)			
general management data router (GMDR)			
host group directory server (HGDS)			
injected managment data router (IMDR)			
IP VPN global server (ipvpnserver)	6800	TCP	
multinodal name service (mnsd)	5502 5503 random	UDP UDP UDP	
multinodal name service agent (mdsdagent)	5934	TCP	
network configuration database server (ncd server)			

Table 99
MDM servers and bind ports

Server	Bind port	Transport protocol (TCP or UDP	Connections to other servers
NCS communication manager (ncsmgr)			
network data access mediator (ndam)			
network model server (nmserver)			
MDM log collector (OAMC)			
Openview alarm translator (ovat)			
Openview data access mediator (ovdam)			
Passport configuration model server (pcms)			
Passport configuration server (pcserver)	6760	TCP	
PM file access server (pfas)			
data viewer data collection daemon (pmdcd)			
performance measurement stream processor (pmsp)	1646 1647	TCP TCP	
Passport command access server (ppaccessserver)	6601	TCP	
Fault device access server (psvagent)			mnsdagent on 5934 (TCP)
restore controller (nsctlrst)	5001	TCP	
restore provider	5021	TCP	
real time alarm collection (rtaccol)			
session servers (CMCFUN, CM, GP)			
SNMP management data router			
surveillance network model updater (surnup)			
server daemon (svmdmn)			

Table 99
MDM servers and bind ports

Server	Bind port	Transport protocol (TCP or UDP	Connections to other servers
trap server daemon (tsvr)	162	UDP	
workstation surveillance server (sfm)			
network time sync (xntp)	123 123	TCP UDP	
Apache	8080 8081	TCP TCP	
DBSyncController (dbsynccontr)	5757	TCP	backup server on 5050 (TCP) pcserver on 6760 (TCP) Interbase on 3060 (TCP) or Oracle 1521 (TCP)
interBase	3060	TCP	
Oracle	1521	TCP	
workstation telnet (telnetd)	23	TCP	
workstation ftpd (ftpd)	20 21	TCP TCP	
workstation nfsd (nfsd)	2049 2049	UDP TCP	
workstation rpc (rpcbind)	111 111	UDP TCP	
workstation print service (lpsched)	515	TCP	
secure FTP (mdmftpd)	2374 2373	TCP TCP	
RADIUS server	1812 1645	UDP UDP	ns-slapd on 389 (TCP)
secure shell daemon (sshd)	22	TCP	

Table 99
MDM servers and bind ports

Server	Bind port	Transport protocol (TCP or UDP	Connections to other servers
Passport Managed Object Agent	1570 1591-1640	TCP TCP	Orbix trader on 15001 (TCP) or next available
MDP administrator (mdpconfigd)	1099	TCP	
MDP file prober (mdpropber)			FTPD on 20 (TCP)and 21
Passport 4400 configurator	80	TCP	
workstation trivial file transfer (tftpd)	69	UDP	
SDM succession pserver	3197	TCP	

Appendix B
Server names, executables, and suggested names

This table lists all server names, their executables, and suggested name in Server Administration.

Table 100
Server name, executables, and suggested name

Server name	Executable	Suggested name in Server Administration
Backup Controller	nsctlbck	Backup Controller
Context Server	ctxsvr	Context Server
Customer Database Server	cdbserver	Cust Data Server
DPN DBNL Auto-disabling Daemon	dbnlwatch	DBNL Watch
DPN Management Data Router	dmdr	DMDR_<OA Name>
DPN NCS Communications Manager	ncsmgr	DPN NCS Comms Mgr
DPN PM File Access Server	pfas	DPN PFAS
DPN PM File Access Software Download	pfas	DPN PFAS SW Download
Data Manager Agent	dma	DMA
Data Synchronization Server	dataSyncServer	Data Sync Server
Data Viewer Agent	pmagent	Data Viewer Agent
(Sheet 1 of 4)		

Table 100
Server name, executables, and suggested name (Continued)

Server name	Executable	Suggested name in Server Administration
Data Viewer Data Collection Daemon	pmdcd	Data Viewer DCD
End-to-End Server	eteserver	End-to-End Server
Fault Device Access Agent	psvagent	Fault Device Access Agent
General Management Data Router	gmdr	GMDR
GMDR Agent	gmdragent	GMDR Agent
Host Group Directory Server	hgds	Host Group Directory
Injected Management Data Router	imdr	IMDR
Log Collector	oamc	Log Collector
MDP DPN File Collector	mdpcol	MDP DPN Collector
MDP DPN File Manager	mdpdpnmgr	MDP DPN File Manager
MDP Disk Manager	mdpdiskmgr	MDP Disk Manager
MDP File Mover Manager	mdpfmmgr	MDP File Mover Manager
MDP Passport Data Model Manager	mdpdmm	MDP Data Model Manager
MDP Passport File Manager	mdppmgr	MDP PP File Manager
MDP Passport File Prober Manager	mdpfpmgr	MDP PP File Prober Manager
MDP Statistics Retrieval System	mdpsrs	MDP SRS
MPE Nodal Provisioning Configuration Server	ncserver	MPE NP Config Server
Multi-nodal Name Server Agent	mnsdagent	MNSD Agent
Multi-nodal Name Server Level 2	mnsd 2	MNSD Level 2
NM Agent	nmagent	NM Agent
Network Configuration Database Server	ncdsvr	NCD Server
Network Data Access Mediator	ndam	NDAM
(Sheet 2 of 4)		

Table 100
Server name, executables, and suggested name (Continued)

Server name	Executable	Suggested name in Server Administration
Network Model Coordinator	dnmnmc	NM Coordinator
Network Model Edit Server	edserver	NM Edit Server
Network Model Server	nmserver	NM Server
Network Model Surveillance Updater	surnup	NM Surv Updater
Nodal Provisioning Configuration Manager	configman	NP Config Manager
Passport 4400 Backup Provider	pbckpp4400	PP4400 Backup Provider
Passport 4400 Restore Provider	prstpp4400	PP4400 Restore Provider
Passport 4460 Backup Provider	pbckpp4460	PP4460 Backup Provider
Passport 4460 Restore Provider	prstpp4460	PP4460 Restore Provider
Passport Backup Provider	pbckpp	PP Backup Provider
Passport Command Access Server	eteserver	PP Command Access Svr
Passport Communications Manager	fdtm	PP Comms Manager
Passport Configuration Model Server	pcms	PP Config Model Server
Passport Management Data Router	fmdr	FMDR_<Group Name>
Passport Nodal Provisioning Configuration Server	pcserver	PP NP Config Server
Passport Restore Provider	prstpp	PP Restore Provider
Performance Measurement Stream Processor	pmsp	PMSP
RTAC Agent	rtacagent	RTAC Agent
Real Time Alarm Collection	rtaccol	Real Time Alarm Col
Restore Controller	nsctlrst	Restore Controller
SNMP Data Collection Daemon	gendcd	gendcd_<group>
SNMP IP Discovery Server	ipdsvr	SNMP IP Discovery Svr
(Sheet 3 of 4)		

Table 100
Server name, executables, and suggested name (Continued)

Server name	Executable	Suggested name in Server Administration
SNMP Management Data Router	smdr	SMDR
SNMP Trap Server	tsvr	SNMP Trap Server
Server Daemon	svmdmn	N/A
Secure FTP Daemon	launchSecureFTPD	Secure FTP Daemon
VPN Monitor Extractor	VPNMonitorExtractor	VPN Monitor Extractor
VPN Monitor Server	VPNMonitorServer	VPN Monitor Server
Workstation Surveillance Server	sfm	Workstation Surv
(Sheet 4 of 4)		

Index

Preside Multiservice Data Manager

Server Reference

Guide

R15.1

Publication: 241-6001-310
Document status: Standard
Document version: 15.1RSUP
Document date: August 2004
Printed in Canada