NORTEL NETWORKS

Preside Multiservice Data Manager

Network Reporting System

User Guide

241-6001-022

Preside Multiservice Data Manager

Network Reporting System

User Guide

Publication: 241-6001-022
Document status: Standard
Document version: 14.3RSUP
Document date: December 2003

Copyright © 2003 Nortel Networks.
All Rights Reserved.

Printed in Canada

NORTEL, NORTEL NETWORKS, the globemark design, the NORTEL NETWORKS corporate logo, PRESIDE, and PASSPORT are trademarks of Nortel Networks. OPENVIEW is a trademark of Hewlett-Packard Company. UNIX is a trademark licensed exclusively through X/Open Company Ltd.

Publication history

December 2003

14.3RSUP Standard
Commercial availability

Contents

About this document

This document is a guide to installing, configuring, customizing and administering the Network Reporting System (NRS). It also describes how to run reports, and create new reports and reporting tools.

The following topics are discussed in this section:

- “Who should read this document and why” (page 17)
- “What you need to know” (page 17)
- “How this document is organized” (page 19)
- “What’s new in this document” (page 20)
- “Text conventions” (page 20)
- “Related documents” (page 22)

Who should read this document and why

This guide is intended for network operators who run network service data reports, administrators who will administer the Network Reporting System (NRS), and programmers who will be creating new reports, tools, and applications which use the NRS database. It can also be used by those who install, engineer, and monitor DPN and Passport networks.

What you need to know

Most NRS tools are command line applications. The exceptions are xnrsrpt, xnrdatah and xnrsdiff. These three tools provide a graphical user interface equivalents to nrsrpt, nrsdatah and nrsdiff.

To use the NRS reporting toolset some basic knowledge of Sun workstations, the Unix operating system and the *vi* editor is required.

The NRS administrator needs to be familiar with:

- NRS setup
- NRS customization options
- using the NRS Automatic Populator and the NRS Populator tools
- compiling NRS reports
- maintaining the NRS
- NCS authentication
- Passport authentication
- service data time stream management naming conventions

The network operators need to be familiar with:

- where the NRS executable reports are located
- how to run a report
- how to select data for input into a report
- how to print reports

The application developers need to be familiar with:

- NRS directory structure
- NRS service data structure
- NRS hierarchical report generator
- *nawk* - pattern matching and processing language
- NRS *nawk* language extensions
- methods for designing reports
- NRS record definition file (RDF) syntax
- NRS toolset

How this document is organized

This guide describes the NRS setup, customization, administration, running reports, and creating new reports and reporting tools. It consists of activity oriented and reference chapters each with a different intended audience. The information is divided into the following chapters:

- "Introducing the Network Reporting System" (page 23) gives a high level description of NRS and its architecture.
- "Installation and configuration" (page 29) is intended for Preside Multiservice Data Manager (MDM) workstation installers and provides details on the hardware and software requirements and configuration for NRS.
- "NRS database population" (page 39) describes the tools used to populate the NRS database with DPN or Passport data.
- "NRS database maintenance" (page 45) describes NRS database maintenance.
- "Automatic NRS/NCD population after download" (page 47) describes how you can automatically populate NRS and NCD after a Component Provisioning download.
- "Service Integrity Audit tool" (page 51) describes the Service Integrity Audit tool. It details how to configure and run NSIC and populate the NRS database.
- "NRS/NCD Population Manager" (page 83) describes the NRS/NCD Population Manager tool, which is used to simplify the population of the NRS and NCD databases.
- "NRS-based Service Integrity Check" (page 99) explains how to use the NRS-Based Service Integrity Checks (NSIC) to detect errors in service data before they are activated in the network.
- "Record definition and module data file formats" (page 129) is intended for application developers and for programmers. It describes the RDF file syntax and the NRS data format for DPN-100 and Passport service data.
- "Creating hierarchical reports" (page 151) is intended for users wanting to create hierarchical reports without having to program.

- “Programming with NRS” (page 157) gives a brief introduction to the *nawk* pattern matching and processing language, and describes the language extensions added to facilitate NRS application development. This chapter also covers, how to create reports and provides detailed descriptions of two sample reports.
- “Sample reports” (page 173) contains sample DPN and Passport reports provided as part of NRS.
- “Running NRS reports” (page 177) is intended for network operators and explains how to run NRS reports and select data for input.
- “NRS toolset graphical user interface” (page 185) describes the graphical user interfaces for the NRS report tools.
- “NRS toolset reference” (page 235) is intended for all NRS users and briefly describes each of the NRS tools and its command line syntax.
- “Configuring NRS” (page 313) provides a typical scenario on how to configure NRS when the NRS database has to be accessible by multiple workstations.
- “Configuring the Service Integrity Audit” (page 315) provides a typical scenario on how to configure the SIS.
- “Service data time management” (page 323) explains the importance of managing time streams in the Network Reporting System.
- “Regular expressions” (page 325) lists the metacharacters that can be used when using regular expressions in NRS.

What’s new in this document

There are no changes to this document for this release.

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

  In Preside Multiservice Data Manager (MDM), 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 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:

- 241-2001-340 *DPN-100 Envelope Definitions*
- 241-6001-100 *Preside MDM Installer Guide*
- 241-6001-101 *Preside MDM Engineering Guide*
- 241-6001-304 *Preside MDM Configuration Management Administrator Guide*
- 241-6001-310 *Preside MDM Server Reference Guide*
- 241-6001-303 *Preside MDM Administrator Guide*
- 241-6001-209 *Preside MDM Provisioning Command Filter API Reference Guide*
- 241-6001-308 *Preside MDM Network Configuration Database Administrator Guide*
- 241-6001-804 *Preside MDM Workstation Utilities User Guide*
- The AWK Programming Language, A. V. Aho, B. W. Kerninghan, P. J. Weinberger, Addison-Wesley, 1988.
- SunOS Reference Manual, Sun Microsystem Inc, 1990.

Chapter 1
Introducing the Network Reporting System

This chapter provides an introduction to the Network Reporting System. In this chapter, you can find the following information:

- “About the Network Reporting System” (page 23)
- “Date Convention” (page 27)

About the Network Reporting System

The Network Reporting System (NRS) provides the capability to extract service data from DPN-100 and Passport modules on the network and store all data in one central data repository. This data can then be used for reporting purposes and is also accessible to other custom-developed applications. Service data is uploaded and stored in a hierarchical data structure based on Configuration components. All data is stored in ASCII files and is therefore readable by other software packages such as a database system, spreadsheet, or report generator. Once the data has been uploaded it can be restructured to suit specific application requirements. Additional customer-specific data could also be stored with the uploaded network service data.

The Network Reporting System toolset consists of a number of simple UNIX-based tools that can be used individually or together. Command sequences that are executed periodically can be incorporated into script files and run with a single command. For example, the Network Reporting System tools along with standard UNIX tools allow the Network Reporting System administrator to setup the system such that any changes to the network service data will automatically be uploaded into the Network Reporting System database.

The components of the Network Reporting System can be viewed, for the purpose of describing the overall architecture, as the following functional areas.

- non-Network Reporting System components
- Network Reporting System administration
- Network Reporting System reporting
- Network Reporting System application development

The Network Reporting System retrieves data from the DPN-100 or Passport network through the system and stores the data in a database. The term database is used in this context to mean a data repository and not a database management system. For an overview of the architecture, see the figure “Network Reporting System” (page 26).

Responsibility for uploading service data and maintaining the database falls in the area of Network Reporting System administration. The tools used for uploading data are the DPN NRS Populator and the Passport NRS Populator. The Network Reporting System administrator, specifies which modules are to be uploaded. The NRS Populator then requests the modules to be uploaded. The system takes care of uploading the module service data from the network and stores it in the Network Reporting System database.

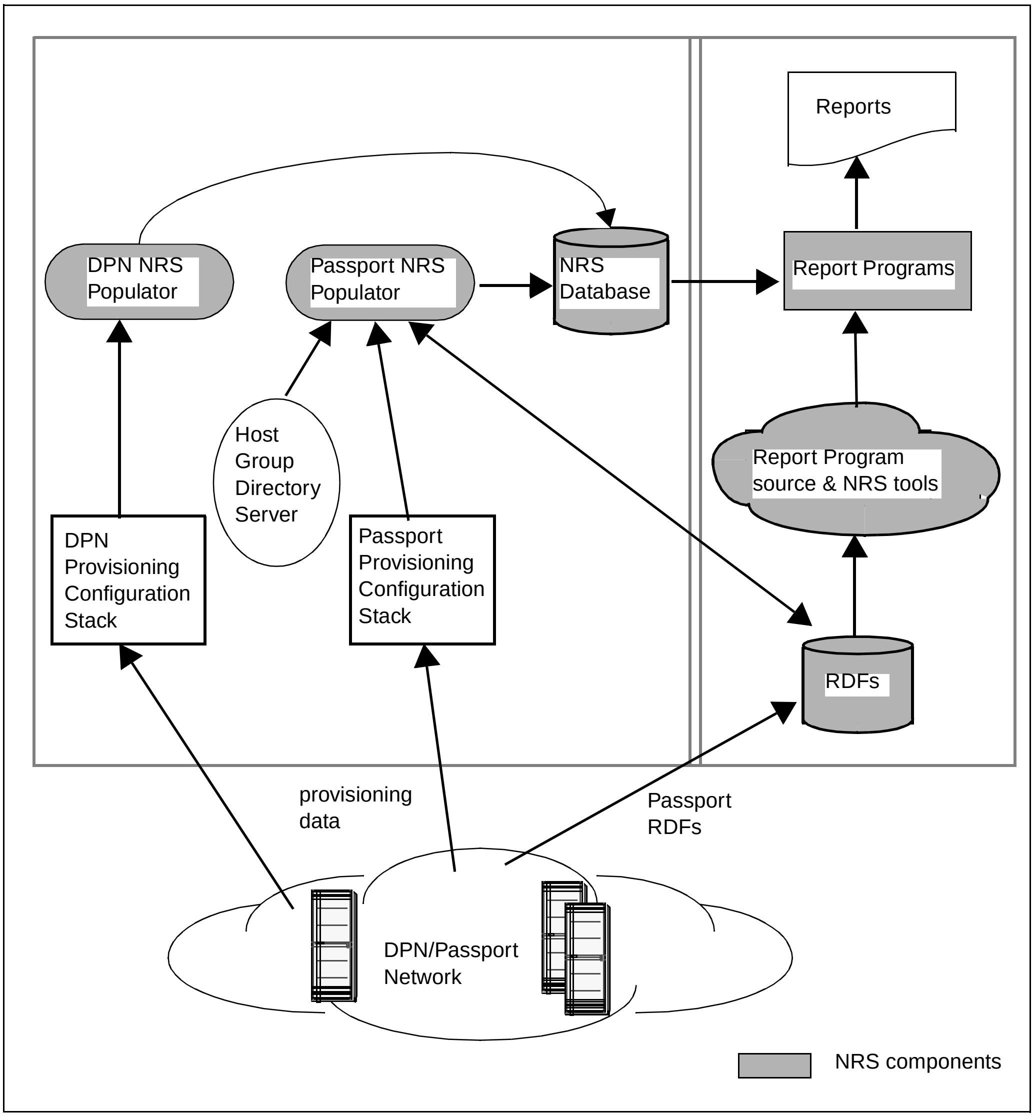
Automatic Network Reporting System population can be set up. For DPN modules, an application is provided that populates the database right after a download has successfully completed. The Service Integrity Audit tool can also be used to ensure that Network Reporting System population occurs regularly and automatically, for DPN and Passport modules.

The Network Reporting System application development area, represented in the figure “Network Reporting System” (page 26) by the “Report program source and Network Reporting System tools” cloud, contains tools to facilitate report creation. This area is the domain of the programmer who creates new reports, modifies existing reports, and develops new tools or applications. Sample reports are provided to show different types of reports that can be created.

The Network Reporting System reporting area, allows an operator to run report programs. The data for the report comes from the Network Reporting System database. A mechanism for selecting modules to report on is provided. Reports are normally output to a disk file and then printed.

Each service data component has a corresponding record definition file (RDF) which defines its record structure in the Network Reporting System database. The RDFs are used by the NRS Populator to determine which format to store the service data. The RDFs are also used by the NRS Preprocessor tool to allow the programmer to reference service data parameters by name rather than offset within the record as well as having access to the parameter attributes (for example, title, width, and so on).

Figure 1
Network Reporting System



Date Convention

In this guide you will often see dates in the format *yymmdd*. Such dates are used as parameters in certain line commands or appear as part of a file name.

To account for the year 2000 and beyond, Network Reporting System interprets 000101 as later than 991231. The base year has been chosen as 1980. This means 800101 precedes 900101 which precedes 000101 which precedes 790101. In other words:

```
if (yy < 80)
   year = 2000 + yy
else
   year = 1900 + yy
```

This means:

80 => 1980
90 => 1990
99 => 1999
00 => 2000
10 => 2010
79 => 2079.

Chapter 2 Installation and configuration

This chapter describes the Network Reporting System hardware and software requirements and how to install and configure the Network Reporting System software. In this chapter, you can find the following information:

- “Hardware requirements” (page 29)
- “Software requirements” (page 30)
- “Network Reporting System directory structure” (page 31)
- “Network Reporting System customization” (page 32)
- “Network Reporting System configuration file” (page 33)
- “Account setup” (page 37)

Hardware requirements

The Network Reporting System (NRS) runs on a standard Preside Multiservice Data Manager (MDM) workstation hardware platform. See 241-6001-100 *Preside MDM Installer Guide* and 241-6001-101 *Preside MDM Engineering Guide* for hardware requirements.

Disk storage requirements for service data in the Network Reporting System database are dependent on the following factors:

- number of modules in the network
- number of bundles stored for every module
- running report programs concurrently
- amount of data processed for every report

- keeping report files (output) on disk

Some Network Reporting System reports create temporary data files in the /tmp directory while they are executing. The size of these files depends on the module selection criteria for the report input, that is, the number of modules involved. For example, if a single module is selected for reporting the resulting temporary files and the report output will be small in contrast to selecting all the modules in the network as input to the report. If several reports are running at the same time the maximum amount of temporary disk space required will be the sum of the temporary disk space required for each report. Therefore, if disk space is limited, to minimize disk space requirements reports may be run in series rather than parallel.

For disk space requirements, see 241-6001-101 *Preside MDM Engineering Guide*.

Software requirements

The Network Reporting System system runs on standard Preside Multiservice Data Manager (MDM) workstation software. See 241-6001-100 *Preside MDM Installer Guide* and 241-6001-101 *Preside MDM Engineering Guide* for software requirements.

The following servers are required to be running in order for the DPN NRS Automatic Populator (nrsauto), Service Integrity Audit tool, and the DPN NRS Populator (nrspop) tools to work.

- Connection Manager (CM)
- file access server (PFAS) must be available on the LAN. See 241-6001-304 *Preside MDM Configuration Management Administrator Guide*.
- MDM multi-nodal name server (mnsd) must be available on the LAN
- MDM NCS communications manager (ncsmgr) must be running on the workstation(s) that are running the NRS Populator and the Service Integrity Audit tools.

The following servers are required to be running in order for the Passport NRS Populator to work.

- Connection Manager (CM)

- Host Group Directory Server (HGDS)
- Passport Communication Manager (FDTM)

See 241-6001-310 *Preside MDM Server Reference Guide* for more information.

Network Reporting System directory structure

Following is the list of directories related to the Network Reporting System:

- */opt/MagellanNMS/cfg*

  Contains the Network Reporting System configuration file NRS.cfg, the Service Integrity Audit configuration file SIS.cfg, and the NRS/NCD Population Manager configuration file PMT.cfg.

- */opt/MagellanNMS/data/nrs/data*

  Default Network Reporting System database location.

- */opt/MagellanNMS/data/nrs/rdf*

  RDF location. Subdirectories dpn, ppc, ppe, and custom store RDFs by switch type.

- */opt/MagellanNMS/data/nrs/rpt*

  Directory that can be used to store the Network Reporting System report programs created by your organization.

- */opt/MagellanNMS/lib/nrs/rpt*

  Contains the sample report programs delivered with the Network Reporting System.

- */opt/MagellanNMS/data/sis/data*

  Default location for Service Integrity Audit to store its output files.

- */opt/MagellanNMS/data/sis/work*

  Default location for Service Integrity Audit to store its control files.

- */opt/MagellanNMS/data/pmt/data*

  Default location for NRS/NCD Population Manager to store its output files.

- */opt/MagellanNMS/data/pmt/work*

  Default location for NRS/NCD Population Manager to store its control files.

Network Reporting System customization

Once the Network Reporting System is installed, it can be used without any further changes. By default, the system assumes that the NRS database is in the /*opt/MagellanNMS/data/nrs/data* directory and that the RDFs are in the subdirectories (“dpn”, “ppe”, “ppc”, “custom”) in the /opt/MagellanNMS/data/nrs/rdf directory.

The system was designed to be simple, flexible and to have few components. Each system can be customized to conform to individual application requirements and operating environments. The Network Reporting System administrator determines who has access and decides on the Network Reporting System setup and operating procedures.

Before attempting to perform the following tasks, you should be familiar with the concept of UNIX file servers and UNIX account setup. Refer to your UNIX documentation for more information.

Selecting the Network Reporting System database and RDF locations

The following is a recommendation on how to select the Network Reporting System database and RDF locations.

It is recommended that the default locations for the Network Reporting System database and RDFs be used. If the Network Reporting System database must be made accessible to other workstations on the LAN, the Network Reporting System database and RDF directories should be NFS mounted by the other workstations using the same directory names. This configuration simplifies Network Reporting System administration.

If you decide to change the location of the RDFs, the RDFs must be copied from the subdirectories (“dpn”, “ppe”, “ppc”, and “custom”) in /opt/MagellanNMS/data/nrs/rdf to the selected directory but must not be removed from the /opt/MagellanNMS/data/nrs/rdf subdirectory. The RDFs in the dpn subdirectory MUST also be copied when a new software release is installed. See “RDF backward compatibility” (page 148) for more details.

See “Configuring NRS” (page 313) for a description on how to configure the Network Reporting System in a LAN environment.

Including customer data in reports

Customer data can be included in Network Reporting System reports. To do so, the filename of the customer data database must be set in the Network Reporting System configuration file. Network Reporting System requires that this file be on a disk accessible by each workstation running reports.

See 241-6001-804 *Preside MDM Workstation Utilities User Guide* for more information on customer data.

Network Reporting System configuration file

Some of the Network Reporting System tools need to know the location of different directories. The Network Reporting System configuration file contains the parameters common across Network Reporting System tools. The Network Reporting System configuration file, NRS.cfg, can be found in the */opt/MagellanNMS/cfg* directory. All Network Reporting System tools which use this file will check for the NRS.cfg file in the *MagellanNMS* directory under the user’s home directory first. If the file is not found there, the file is taken from the */opt/MagellanNMS/cfg* directory. It is therefore possible to give different users different configuration files. For example, an application developer may want the Network Reporting System tools to use include files from their local development environment.

Configuration parameters

The following is a description of each Network Reporting System parameter contained in the Network Reporting System configuration file.

NRS_DATA_DIR

Directory containing the Network Reporting System data files. This directory is referred to as the Network Reporting System database directory. The default directory is */opt/MagellanNMS/data/nrs/data*. It is recommended to use this directory only when multiple Network Reporting System databases must exist on the same workstation.

NRS_RDF_DIR

Directory containing the Network Reporting System record definition files (RDF). These files describe the organization of the Network Reporting System data files and can be viewed as the schema for the Network Reporting System database. The default directory is /opt/MagellanNMS/data/nrs/rdf and contains the following subdirectories:

dpn --> DPN RDFs
ppc --> Passport carrier RDFs
ppe --> Passport enterprise RDFs
custom --> custom RDFs.

It is highly recommended not to change this parameter.

NRS_DEFAULT_PP_TYPE

The Passports are treated as two distinct families:

ppc -->Passports running a version of software for carriers
ppe -->Passports running a version of software for enterprises

Some reporting programs need to know which Passport type to use when a specific type is not specified.

NRS_INCLUDE_DIR

Directory containing include files used when creating *nawk* reports. Originally, the parameter is set to /opt/MagellanNMS/lib/nrs/rpt and some include files are provided in this directory.

NRS_CUST_DATA_DB_NAME

Customer data database filename. This parameter identifies the customer data database that is queried when requesting customer data to be incorporated into reports. By default, this parameter is left blank. If it remains blank, customer data is not incorporated into the reports.

NRS_CUST_DATA_DB_DIR

Customer data database location. This parameter identifies the directory where the customer data database is stored. The directory needs to be accessible through the file system in order to be used to generate reports. By default, this parameter is set to */opt/MagellanNMS/data/cdb*.

NRS_CUST_DATA_DB_HOST

Customer data database host location. This parameter identifies the host where the customer data database resides. By default, this parameter is set to localhost. This parameter is present for future use only.

How to edit the Network Reporting System configuration file

The following example shows the actual contents of the NRS.cfg file and how to edit it. Use an editor, such as vi, to edit this file.

```
# --------------------------------------------------#
#   Module Name:    NRS.cfg
#
#   Description:        This file contains the key configuration
#                       to the Network Reporting System (NRS).
#
#                       The fields are all on one line and are
#                       separated by tab/space characters.
#
#                       Line format is: NRS <parameter> <value>
#
#                       Comment lines start with the ’#’
#                       character.
#                       The set of configuration parameters and
#                       their functions are described lower in
#                       this file.
#
```

```
(...)
# ---------------------------------------------------#
# This parameter defines the full directory path where the
# NRS data files are stored/retrieved.
#
#NRS    NRS_DATA_DIR    <full path of directory>
 NRS    NRS_DATA_DIR    /opt/MagellanNMS/data/nrs/data
#
# This parameter defines the full directory path where the
# NRS Record Definition Files (RDFs) can be found.
#
#NRS    NRS_RDF_DIR     <full path of directory>
 NRS    NRS_RDF_DIR     /opt/MagellanNMS/data/nrs/rdf/
#
# This parameter defines the full directory path where the
# include files that the user wants to include in their
# reports are stored.
#
# This parameter defines the default Passport type.
# Depending on the version of software running on a
# Passport, a type (ppc or ppe) is associated with the
# Passport during the NRS population. When specifying a
# component for the purposes of reporting (example: using
# the nrsdatah report), the type specified here will be
# used if a Passport component name is used without a
# specific Passport type.
#
#NRS    NRS_DEFAULT_PP_TYPE  <default Passport type>
#NRS    NRS_DEFAULT_PP_TYPE  ppc
#
#NRS    NRS_INCLUDE_DIR  <full path of directory>
 NRS    NRS_INCLUDE_DIR  /opt/MagellanNMS/lib/nrs/rpt
#
# This parameter defines the Customer Data Database Name
# used when running the NRS reports.
#
#NRS    NRS_CUST_DATA_DB_NAME <database name>
 NRS    NRS_CUST_DATA_DB_NAME
#
```

```
# This parameter defines the full directory path where the
# Customer Data Database is located.
#
#NRS    NRS_CUST_DATA_DB_DIR   <database full path of
#                              directory>
 NRS    NRS_CUST_DATA_DB_DIR   /opt/MagellanNMS/data/cdb
#
# This parameter defines the host where the Customer Data
# can be retrieved (this is for future use).
#
#NRS    NRS_CUST_DATA_DB_HOST <database host name>
 NRS    NRS_CUST_DATA_DB_HOST localhost
#
```

The actual Network Reporting System configuration file records are lines that do not start with a # character. The format of the records, which contain three fields each are:

```
<sub-system> <environment_variable> <value>
```

Only the <value> should be changed in this file. MagellanNMS

Example

If the Network Reporting System administrator chooses to place the Network Reporting System report include files in a different directory, such as, /file_server1/nrs/include, then the NRS.cfg file line

```
 NRS NRS_INCLUDE_DIR /opt/MagellanNMS/lib/nrs/rpt
```

would be replaced by the following:

```
NRS NRS_INCLUDE_DIR /file_server1/nrs/include
```

Once the change has been made, the next Network Reporting System tool to be used will use the new Network Reporting System configuration values.

Account setup

There are no special requirements for Preside Multiservice Data Manager (MDM) users to access the Network Reporting System.

Chapter 3 NRS database population

This chapter describes the populator tools that you can use to populate the Network Reporting System (NRS) database. In this chapter, you can find the following information:

- “DPN NRS Populator tool” (page 39)
- “Using the DPN NRS Populator” (page 41)
- “Passport NRS Populator” (page 42)
- “Automating NRS database population” (page 43)

DPN NRS Populator tool

The DPN NRS Populator tool is used to upload module service data from the DPN-100 network. The tool then reformats the data and stores it in the Network Reporting System database. The actual filename for the DPN NRS Populator tool is nrspop and it is located in the /opt/MagellanNMS/bin directory. The tool can be invoked from any directory since it uses the NRS configuration file NRS.cfg to determine where the RDFs and the Network Reporting System database are located.

Service data retrieval is done through the Provisioning stack, consisting of the PM file access server (PFAS), the SDA and Envelope Access Server (SEA), and the field access server (FA).

Input for the Populator consists of a list of modules to be uploaded, which are always specified in the form of a command file and the NCS authentication which is specified on the Populator command line. A log file option is available for recording transactions.

Each command line in the command file contains enough information to uniquely identify the module to be uploaded, where it is to be uploaded from and which MCF to upload (active, committed or other).

Two Network Reporting System configuration file variables are used by the NRS Populator: NRS_DATA_DIR and NRS_RDF_DIR. NRS_DATA_DIR points to the directory where the data files uploaded by the NRS Populator tool will be stored and NRS_RDF_DIR points to the NRS RDFs used by the NRS Populator tool. These two variables are set in the NRS.cfg file. If a copy of the file does not exist in the *MagellanNMS* directory under the user's home directory, the NRS Populator tool will look in the /opt/MagellanNMS/cfg directory.

Multiple NRS Populators can execute simultaneously. The primary reason for running multiple DPN NRS Populators would be due to large data volumes. For large networks in which periodic uploading of data could take hours, the upload time can be reduced by using multiple DPN NRS Populators for uploading. While more than one DPN NRS Populator can be run on the same machine, it is recommended that the second or third DPN NRS Populator be invoked on another machine with a lower resource load. Several DPN NRS Populators executing on the same machine will slow each other down due to high resource consumption. The Service Integrity Audit tool can be used to run multiple populators on multiple machines. See "Service Integrity Audit tool" (page 51) for more information.

It is recommended that single or multiple DPN NRS Populators connect to the NCS hierarchy at the root. This way, the entire network becomes accessible.

Creating a DPN NRS Populator command file

Before the DPN NRS Populator is started, a command file of modules to be uploaded must be created. This file can be created using an editor, such as *vi*. Each record in the command file specifies one module to upload and the options. A record is terminated by a newline or return.

Comments may be included by adding a # character as the first non-blank character on the line.

Command file syntax:

```
<pm> -uploadmode <upload_mode>
  [<upload_bundle_or_key_or_date> [upload_namsid]]
  [-uploadlocation <location_mode> [<location_value>]]
```

See “DPN NRS Populator (nrspop)” (page 236) for details on the NRS Populator command file syntax.

Example

The following is an example of a DPN NRS Populator command file:

```
# This is a sample nrspop command file
R34 -uploadmode dated 930317 4034
R34 -uploadmode K 92WK25 4034 -uploadlocation NMS_DISK
R35 -uploadmode committed
R36 -uploadmode ACT -uploadlocation User_specified R40
R37 -uploadmode US ABCD -uploadlocation BACKUP_DISK
```

The DPN NRS PM Lister tool is also provided to help create command files. See “DPN NRS PM Lister (nrspmlst)” (page 258) for more details.

Using the DPN NRS Populator

The following is a description of what occurs when the nrspop tool is invoked.

When the DPN NRS Populator is invoked, it first displays the message The nrspop tool is starting. The command line is parsed and verified and a log file is opened if this option was specified. The NRS configuration variables for the NRS data directory and the RDF directory are checked. The request command file is read, parsed, and verified and the Provisioning stack is initialized. The tool is ready to proceed and the message Processing X request(s) is displayed. A request for module service data is sent to Configuration for the first module and the message Processing request 1 of X: followed by the command file parameters specified for that module are displayed.

When the data is received, if the module is successfully written to disk the message File <module filename> is created successfully is displayed. The next request is then processed.

Once all requests have been processed, just before the Populator exits, it prints the message The nrspop tool has completed successfully.

The following is an example of DPN NRS Populator successfully uploading one request:

```
15:57:50 NOTE: The nrspop tool is starting.
Date: 1994-02-07.
15:57:56 NOTE: Processing 1 request(s).

15:57:56 NOTE: Processing request 1 of 1:
R34 -uploadmode DATED 940301 -uploadlocation
USER_SPECIFIED R34

15:59:53 NOTE: File /opt/MagellanNMS/data/nrs/data/
dpn.R34.4034.94021703.940217.
data is created successfully.

15:59:53 NOTE: The nrspop tool has completed
successfully.
```

Passport NRS Populator

The Passport NRS Populator tool is used to upload module service data through Configuration from the Passport network. The tool then reformats the data according to the Passport RDFs and stores it in the NRS database.

Creating a Passport NRS Populator command file

Before the NRS Populator is started, a command file of modules to be uploaded can be created. This file can be created using an editor, such as vi. Each record in the command file specifies one module to be uploaded and the options. A record is terminated by a newline or return. Comments may be included by adding a # character as the first non-blank character on the line.

The command file syntax is as follows:

```
<pm> -uploadmode <mode> [<key>|<date>|<view filename>]
```

See “Passport NRS Populator (pnrspop)” (page 240) for more information on the Passport NRS Populator command file and command line syntax.

The following is an example of a Passport NRS Populator command file.

```
#Start of command file
NODER1 -uploadmode CURRENT
NODER2 -uploadmode K MYKEY
NODER3 -uploadmode COMMITTED
```

Automating NRS database population

Network Reporting System database population can be configured to be done automatically for DPN and Passport data.

The NRS Populator can be configured to ensure automatic population with minimal intervention required by the NRS administrator. This can be done by running the Service Integrity Audit (*sisauto*) tool on a regular basis from a cron job.

The Service Integrity Audit tool is used to populate the NRS database and, optionally, to execute the NRS-based Service Integrity checks (NSIC) and populate the Network Configuration Database (NCD). Since NSIC and NCD population are optional when running Service Integrity Audit, therefore, Service Integrity Audit can be run only to populate the NRS database. Service Integrity Audit populates only the *valid* MCF/View File per module thus saving time when populating the NRS database. The *valid* MCF/View File is determined by the variables set in the Service Integrity Audit configuration file. Also, Service Integrity Audit can be executed from one workstation and be configured such that the other processes are run on multiple workstations. For more information, see “Service Integrity Audit tool” (page 51).

Chapter 4
NRS database maintenance

This chapter explains how to maintain your Network Reporting System database. In this chapter, you can find the following information:

- “File cleanup” (page 45)
- “File backup and recovery” (page 45)

File cleanup

As the Network Reporting System (NRS) data is populated, the NRS database will continue to grow. Eventually the NRS database will contain data that is no longer relevant. The NRS Module File Deletor tool (nrsrm) is provided to help facilitate the deletion of DPN and Passport module data files.

See “NRS Module File Deletor (nrsrm)” (page 287) for a description of this tool.

File backup and recovery

No backup or recovery mechanism is provided since the NRS database is not the primary source of the service data. In case of problems, the data must be re-populated.

Chapter 5
Automatic NRS/NCD population after download

This chapter describes, for DPN modules, how to automatically populate NRS after a download. In this chapter, you can find the following information:

About NRS/NCD population

To keep the NRS database up-to-date, the best time for you to populate the database is right after a download. Automatic NRS/NCD population is accomplished using the API command filter. The application is easy to use and does not require you to know the API command filter syntax.

Once activated, the application is automatically called after a DPN Component Provisioning download has completed successfully. By default, it always populates the MCF to NRS, but not to NCD. It can be customized to:

- only populate the MCF if the dated naming convention is used and the date is less than or equal to the specified date
- populate the MCF from the backup disk when the MCF is downloaded to the module
- populate the MCF to NCD

- let you decide whether or not the MCF must be populated to NRS/NCD

Usage

The following is the complete command line syntax used to start the DPN Component Provisioning tool with this application.

```
/opt/MagellanNMS/bin/pui -filter_width 100000
   -filter_appl "/opt/MagellanNMS/bin/acf_post_download
        [-backgrnd]
   /opt/MagellanNMS/bin/acf_pd_nrspop
        [-dated <date or file containing a date>]
        [-optional]
        [-from_backup_disk_instead_of_pm]
        [-ncd <NCD database name>]
        [-quiet]"
```

where:

`-backgrnd`

With this option, the NRS/NCD population is started in the background after a successful download. This lets you start another provisioning session while the population is in progress. Without this option, the download is considered complete only when the population is complete.

`-dated <date or file containing a date>`

When specified, only a dated MCF with a date less than or equal to the specified date is populated. Other MCFs are ignored.
The date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.
The date can be specified on the command line or in a file. Using a file makes it easier to modify the date at a later time. Without this option, every MCF is populated.

`-optional`

After a successful download, prompt the user whether or not to proceed with the population. Without this option, the population is always done.

`-from_backup_disk_instead_of_pm`

Populate the MCF to NRS by uploading it from the NMS backup disk if the MCF was downloaded to the module. An MCF downloaded to the NMS disk is always populated from the NMS disk. Without this option, the MCF is populated from where it was downloaded (module or NMS disk).

`-ncd <NCD database name>`

With this option, populate the MCF to the specified NCD database after the NRS population. Without this option, no NCD population is performed.

`-quiet`

When specified, information messages are not displayed.

Activating the application

Since Component Provisioning can be started from different places, such as the Preside MDM Window, Network Viewer (NV), and Component Information Viewer (CIV), a configuration file is provided. The configuration file is called PUIDpnCmd.cfg and is located in */opt/MagellanNMS/cfg*. Every DPN Component Provisioning session reads this configuration file. For details, see “Examples” (page 49).

Examples

The following examples show different ways to configure the PUIDpnCmd.cfg file.

Example 1

```
-filter_width 100000
-filter_appl "/opt/MagellanNMS/bin/acf_post_download
   /opt/MagellanNMS/bin/acf_pd_nrspop -quiet"
```

After a successful download from the DPN Component Provisioning tool, this example always populates the resulting MCF in the Network Reporting System. Information messages are not displayed to the user.

Example 2

```
-filter_width 100000
-filter_appl "/opt/MagellanNMS/bin/acf_post_download
   /opt/MagellanNMS/bin/acf_pd_nrspop -optional -fr"
```

After a successful download from the DPN Component Provisioning tool, this example prompts you for whether or not you want to populate the resulting MCF in NRS. If you want to, this example populates the MCF from the NMS backup disk, if you had downloaded the MCF to the module.

Example 3

```
-filter_width 100000
-filter_appl "/opt/MagellanNMS/bin/acf_post_download
       -backgrnd
   /opt/MagellanNMS/bin/acf_pd_nrspop -dated
       mydate_file -ncd ncddb"
```

After a successful download from the DPN Component Provisioning tool, this example populates the resulting MCF in NRS only if it is of type dated and the date is less than or equal to the date specified in the file mydate_file. After the NRS population, this example populates the MCF in the NCD database called ncddb.

The NRS and NCD population is started in the background, which lets you proceed with the DPN Component Provisioning tool, while the population is running.

For further information on API command filters, see 241-6001-209 *Preside MDM Provisioning Command Filter API Reference Guide*.

Chapter 6
Service Integrity Audit tool

This chapter describes the Service Integrity Audit tool. In this chapter, you can find the following information:

About the Service Integrity Audit tool

The Service Integrity Audit tool is used to populate the NRS database for DPN and/or Passport modules, and, optionally, to execute the NRS-based Service Integrity checks (NSICs), and to populate the Network Configuration Database (NCD). With this tool, the MCFs/View Files are retrieved directly from the modules.

When the tool is executed the tool will determine for DPN modules the *valid* MCF for each PM and populate the MCF into the NRS database. Similarly for Passport modules, the tool will determine the valid View File for each module and populate it into the NRS database. It then performs the NSIC checks and populates the NCD. When the Service Integrity Audit is executed, an NRS database population is always performed, but NSIC checks and NCD population are optional and can be omitted.

Valid MCFs and View Files are determined by the variables set in the Service Integrity Audit configuration file. The MCF/View Files that best fit the criteria specified are chosen. Service Integrity Audit supports DATED mode for MCF/View Files. Customers using KEYED mode can only use the tool to retrieve the Active (current) or Committed MCF/View File.

The Service Integrity Audit can be executed from one workstation and be configured such that the MCF List, DPN NRS Population, View File List and Passport NRS Population processes run on multiple workstations. This allows for parallel processes to be run and maximizes computer resources.

Service Integrity Audit tool model

The Service Integrity Audit is comprised of two tools, sisauto and sisautui. The *sisauto* tool is command line based and can be executed from the command line or a cron job. The *sisautui* tool is a textual interface executed from the command line or selected from the MDM Toolsets menu under Configuration -> DPN Devices -> Network Reporting System or Configuration -> Passport Devices -> Network Reporting System and is called Service Integrity Audit. The tool prompts the user for key parameters and then executes the sisauto tool using these parameters.

The *sisauto* tool executes two processes for DPN modules: MCF List and DPN NRS Population; and two processes for Passport modules: View File List and Passport NRS Population.

For DPN, MCF List determines for each PM, the *valid* MCF from the PM disk. Depending on the configuration and the MCF retrieved, it creates a DPN NRS Population request or nothing if the MCF is already in NRS.

DPN NRS Population populates the NRS database with the specified MCF.

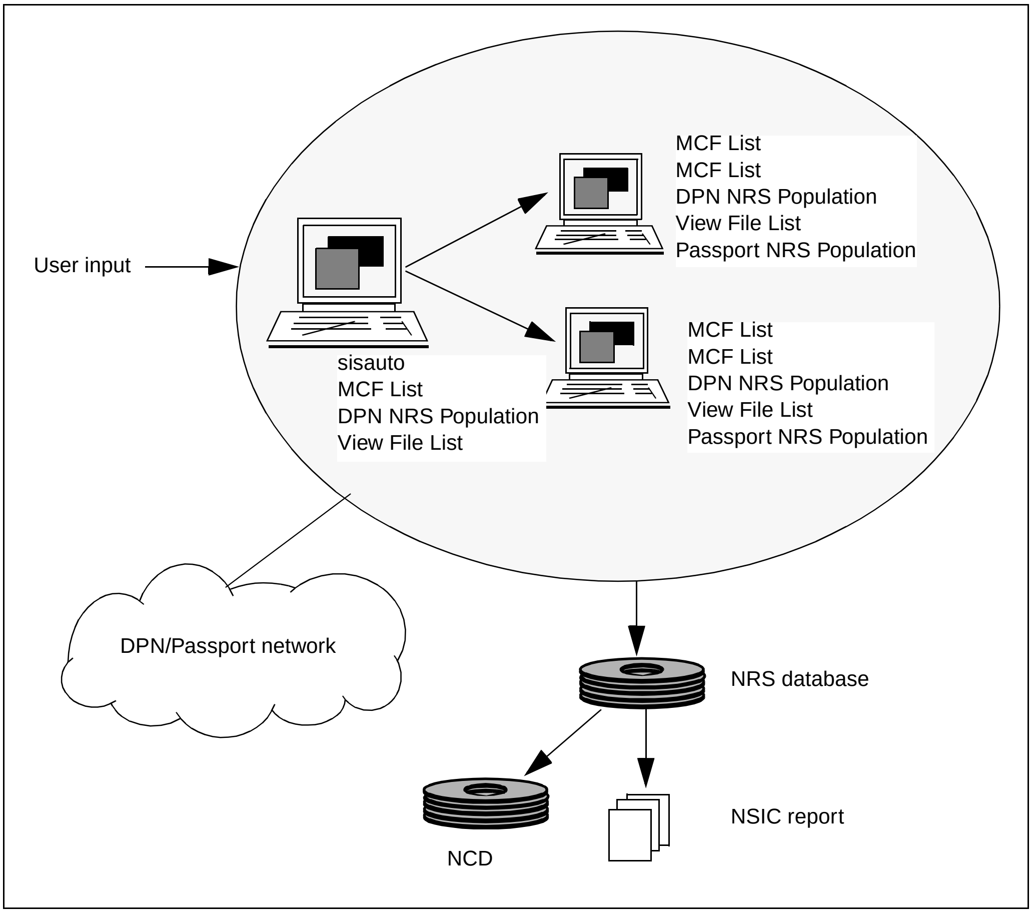
For Passport, View File List determines for each EM, the valid View File from the EM disk. Depending on the configuration and the View File retrieved, it creates a Passport NRS Population request or nothing if the View File is already in NRS.

Passport NRS Population populates the NRS database with the specified View File.

After these processes have been run for all the modules, all the NSIC checks are run on the NRS database. A summary of the NSIC errors or warnings are displayed on the screen. For more information on NSIC, see “NRS-based Service Integrity Check” on page 99.

NCD is then populated. For more information on NCD, see NTP 241-6001-308 *Preside MDM Network Configuration Database Administrator Guide*.

Figure 2
Service Integrity Audit workstation configuration example



DPN MCF naming

For DPN modules, the Service Integrity Audit tool processes only dated MCFs. A dated MCF is any MCF where the bundle name is a valid date in the format yymmdd, followed by a two digit index. For example, MC.94051000.4034.0, MC.94051099.4034.0, and MC.95123117.4034.0. See “Date Convention” on page 27 for more information on the date format.

Active and committed MCFs can also be processed by Service Integrity Audit regardless of the MCF naming convention used.

Passport View File Naming

For Passport modules, the Service Integrity Audit tool processes only dated View Files . A dated View File is any View File where the name is a valid date in the format yymmdd, followed by a two digit index. For example, 96051000.full.012, 96051099.full.056, and 96123117.full.329. See “Date Convention” on page 27 for more information on the date format.

Current and Committed View Files can also be processed by Service Integrity Audit regardless of the View File naming convention used.

Configuration

Before the Service Integrity Audit tool can be run, the following must be configured:

- Network Reporting System (NRS)
- Service Integrity Audit configuration file
- MCF List, DPN NRS Population, View File List and Passport NRS Population instances files

Network Reporting System (NRS)

NRS must be configured before using the Service Integrity Audit tool. When Service Integrity Audit is configured to use multiple workstations, NRS must be configured the same on all the workstations. The NRS data directory must be NFS mounted as well as the NRS RDF directory and the NRS configuration file on each workstation must refer to those directories.

See “Installation and configuration” on page 29 for more information on NRS, and NTP 241-6001-100 *Preside MDM Installer Guide* for more information on NFS.

Service Integrity Audit configuration file

The Service Integrity Audit configuration file is called *SIS.cfg* and can be found in the
/opt/MagellanNMS/cfg directory. This file can be copied to the *MagellanNMS*

directory under the user’s home directory or modified directly. The tool looks for this file in the user specific directory first. If not found, then the tool looks in the */opt/MagellanNMS/cfg* directory..

When the tool is configured to make use of multiple workstations, it uses only the configuration file where the sisauto tool is running. The same is true when the *sisautui* tool is started from a remote workstation.

The Service Integrity Audit configuration file must be modified to set the following global variables.

`SIS_DATA_DIR`

Service Integrity Audit requires a read/write directory to store the output files produced. This directory must be mounted by all the workstations used to run this tool.
The default directory is /opt/MagellanNMS/data/sis/data.

`SIS_WORK_DIR`

Service Integrity Audit requires a read/write directory where work and temporary files used or created by the tool can be found. This directory must be mounted by all the workstations used to run Service Integrity Audit.
The default directory is /opt/MagellanNMS/data/sis/*work*.

`SIS_MCF_DETERMINATION_MODE`

This variable is for DPN modules only.

There are different ways to perform the integrity checks. It is preferable to perform the integrity checks before activating the new service data. However, it is possible to perform the checks on the active or committed service data.

This variable is used to determine which MCF is to be considered the *valid* MCF to populate in NRS before performing the NSIC checks.

The following is a list of the possible values and their descriptions:

0
Active mode. Populate the ACTIVE MCF in NRS.

1
Committed mode. Populate the COMMITTED MCF in NRS.

2 (default)
NMS mode. If an MCF exists for the criteria defined by the SIS_NMS_MCF_MODE variable, it is populated to NRS. If no MCF exists, an error is generated for the PM.

3
No longer used.

4
No longer used.

`SIS_NAS_MCF_MODE`

This variable is no longer used and should be set to 0.

`SIS_NMS_MCF_MODE`

This variable is for DPN modules only.

When NMS MCFs are involved (when the variable SIS_MCF_DETERMINATION_MODE is set to 2), there are four ways to select the *valid* MCF:

0
Select the NMS MCF with the highest bundle date.

1
Select the NMS MCF with the highest file system date. This would be the NMS MCF that was downloaded last to the PM.

2 (default)
One way to activate new service data is to activate it on a certain day of the week. Service data is prepared days before the activation and, on the given day of the week, the new service data is activated on every PM. When preparing the service data in Preside Multiservice Data Manager (MDM), the DATED key is set to the date of activation. In order to validate the service data before the activation, these “future” MCFs must be selected by the tool.

The day of the week to be used is determined by the SIS_NMS_MCF_DAY variable. For example, if this variable is set to Sunday and the tool is called on Tuesday March 15 1994 (940315), the NMS MCF with a bundle date closest to, but not exceeding 940320 (the upcoming Sunday) will be selected. It is also possible to set the SIS_NMS_MCF_DAY variable to today, tomorrow, in 4 days, in 10 days etc.

3
For service data defined to be activated on a specific day or next month, option 2 does not work.

This mode is used to specify a date using the -*nmsdated* option on the *sisauto* command line. This date is used to select the NMS MCF with a bundle date closest to, but not exceeding, that date.

`SIS_MERGED_MCF_MODE`

This variable is no longer used and should be set to 0.

`SIS_NMS_MCF_DAY`

This variable is for DPN modules only.

This variable defines a day, relative to the current date, used when determining the date to use for selecting NMS MCFs. This variable is only used when the SIS_NMS_MCF_MODE variable is used and set to 2. Values are:

-7	Sunday
-6	Saturday
-5	Friday
-4	Thursday
-3	Wednesday
-2	Tuesday
-1	Monday
0	today (default)
1	tomorrow
2..1000	in X days

`SIS_VIEWFILE_DETERMINATION_MODE`

This variable is for Passport modules only.

Different customers have different ways of performing the integrity checks. It is preferable to perform the integrity checks before activating the new service data. However, it is possible to perform the checks on the current or committed service data.

This variable is used to determine which view file is to be considered the valid one to populate in NRS before performing the NSIC checks.

The following is a list of possible values and their descriptions:

0
Current mode. Populate the CURRENT view file in NRS.

1
Committed mode. Populate the COMMITTED view file in NRS.

2 (default)
NMS mode. If an NMS view file exists for the criteria defined by the SIS_NMS_VIEWFILE_MODE variable, it is populated to NRS. If none exist, an error is generated for the EM.

`SIS_NMS_VIEWFILE_MODE`

This variable is for Passport modules only.

When NMS view files are involved (when the variable SIS_VIEWFILE_DETERMINATION_MODE is set to 2), there are four ways to select the valid one:

0
Select the NMS view file with the highest date as part of the file name.

1
Select the NMS view file with the highest file system date. This means the NMS view file that was downloaded last to the PM.

2 (default)
You may decide to activate new service data on a certain day of the week. Prepare the service data days before the activation and, on the given day of the week, the new service data is activated on every EM. When preparing the service data in NMS, use the date representing the day of activation as the DATED key. In order to validate the service data before the activation, these “future” view files must be selected by the tool.

The day to use is determined by the SIS_NMS_VIEWFILE_DAY variable. If, for example, this variable is set to Sunday and the tool is called on Tuesday March 12 1996 (960312), the NMS view file with a bundle date closest to, but not exceeding, 960317 (the upcoming Sunday) will be selected. It is also possible to set the SIS_NMS_VIEWFILE_DAY variable to today, tomorrow,, in 4 days, in 10 days, etc.

3
Option 2 does not support things like retrieving the service data to be activated on a specific date or next month. This mode is used to allow the user to specify a date using the -nmsdated_em option on the command line. This date is used to select the NMS view file with a date as part of the file name closest to, but not exceeding, that date.

`SIS_NMS_VIEWFILE_DAY`

This variable is for Passport modules only.
This variable defines a day, relative to the current date, used when determining the date to use for selecting NMS view files. This variable is only used when the SIS_NMS_VIEWFILE_MODE variable is used and set to 2. Possible values are:

-7	Sunday
-6	Saturday
-5	Friday
-4	Thursday
-3	Wednesday
-2	Tuesday
-1	Monday
0	today (default)
1	tomorrow
2..1000 in X days	

`SIS_NSIC_EXTERNAL_DNA_FILE`

This file is also used by the NCD population tool.
With NSIC, it is possible to specify a file containing a list of DNAs that are external to the DPN-100 network. For example, NM DNAs, and DNAs of another network.

This parameter can be used to specify such a file or it can be left blank. No default value provided.

`SIS_NSIC_EXTERNAL_RID_FILE`

This file is also used by the NCD population tool.
With NSIC, it is possible to specify a file containing a list of RIDs that are external to the DPN-100 network. For example, NM RIDs, and RIDs of another network.

This variable can be used to specify such a file or it can be left blank. No default value provided.

`SIS_NSIC_EXTERNAL_IPADDR_FILE`

With NSIC, it is possible to specify a file containing a list of IP addresses that are external to the DPN100 /Passport network (e.g. IP addresses of another network).

This variable can be used to specify such a file. It can be left blank. There is no default value.

This file is also used by the NCD population tool.

`SIS_NCD_TARGET`

This variable defines the NCD database to be used for NCD population. It is optional. If left blank or if the -noncdpop option is used, NCD population is not performed.

`SIS_OUTPUT_FILES_KEEP_COUNT`

The *sisauto* tool produces a set of files (log, activation list, NSIC errors) in the SIS_DATA_DIR directory each time it is executed. This variable defines how many sets of files will be kept. All the files generated on the same day are considered in the same set. Only the latest sets are kept and all other sets are deleted. This allows for the SIS_DATA_DIR directory to be maintenance free. Acceptable values are 1-10000.

`SIS_NCS_DESTMNEM_DEFAULT`

With the *sisautui* tool, it is possible to provide a default for the NCS Destination Mnemonic parameter. No default value provided.

For example, MDI

`SIS_NCS_CAPID_DEFAULT`

With the *sisautui* tool, it is possible to provide a default for the NCS Capability ID parameter. No default value provided.

For example, ID

SIS_PM_FILE_DEFAULT

With the *sisautui* tool, when using the file PM selection mode, a file containing a list of PMs must be specified. This variable can be used to provide a default file name. No default value provided.

For example, /localdisk/sis/PM.list

`SIS_AUTH_GROUP_DEFAULT`

With the sisautui tool, it is possible to provide a default for the group name part of the authentication parameters for Passport. There is no default value.

For example: GRP

`SIS_AUTH_USERID_DEFAULT`

With the sisautui tool, it is possible to provide a default for the user ID part of the authentication parameters for Passport. There is no default value.

For example: ID

`SIS_EM_FILE_DEFAULT`

With the sisautui tool, when using the “file” mode of execution for Passport modules, a file containing a list of EMs must be specified. This variable can be used to provide a default file name. There is no default value.

For example: /localdisk/sis/EM.list

Example

The following is an example of an Service Integrity Audit configuration file. Comment lines, indicated by (...), have been removed but appear in the actual file.

```
#-----------------------------------------------------------------------#
#
#    Module Name:  SIS.cfg
#
#    Description:  This file contains the key configuration variables
#                  for the Service Integrity Audit tools.
#
#                  Line format is: SIS <variable> <value>
#
#                  Comment lines start with the '#' character.
#                  The set of configuration variables and their
#                  functions are described lower in this file.
(...)
SIS  SIS_DATA_DIR                     /opt/MagellanNMS/data/sis/data
(...)
SIS  SIS_WORK_DIR                     /opt/MagellanNMS/data/sis/work
(...)
SIS  SIS_MCF_DETERMINATION_MODE       2
(...)
SIS  SIS_NAS_MCF_MODE                 0
(...)
SIS  SIS_NMS_MCF_MODE                 3
(...)
SIS  SIS_MERGED_MCF_MODE              0
(...)
SIS  SIS_VIEWFILE_DETERMINATION_MODE  2
(...)
```

```
SIS  SIS_NMS_VIEWFILE_MODE           2
(...)
SIS  SIS_NMS_VIEWFILE_DAY            0
(...)
SIS  SIS_NMS_MCF_DAY                 0
(...)
SIS  SIS_NSIC_EXTERNAL_DNA_FILE      /localdisk/sis/ext.dnas
(...)
SIS  SIS_NSIC_EXTERNAL_RID_FILE      /localdisk/sis/ext.rids
(...)
SIS  SIS_NSIC_EXTERNAL_IPADDR_FILE   /localdisk/sis/ext.ipaddresses
(...)
SIS  SIS_NCD_TARGET                  NCDDB
(...)
SIS  SIS_OUTPUT_FILES_KEEP_COUNT     7
(...)
SIS  SIS_NCS_DESTMNEM_DEFAULT        MDI
(...)
SIS  SIS_NCS_CAPID_DEFAULT           ID
(...)
SIS  SIS_PM_FILE_DEFAULT             /localdisk/sis/pm.list
(...)
SIS  SIS_AUTH_GROUP_DEFAULT          GRP
(...)
SIS  SIS_AUTH_USERID_DEFAULT         ID
(...)
SIS  SIS_EM_FILE_DEFAULT             /localdisk/sis/em.list
```

Instances configuration files

Instances files are used to control the number of parallel processes that are performing the MCF List, DPN NRS Population, View File List and Passport NRS Population tasks. These files, *.mcflist.instances*, *.nrspop.instances,* .viewfilelist.instances, .pnrspop.instances, and *.mcfmerge.instances*, must be located in the directory referred to by the SIS_WORK_DIR variable in the Service Integrity Audit configuration file. Read permission is required for these files.

The Service Integrity Audit tool requires that each of the above instances files be present even if it is not required. For example, even though the *.mcfmerge.instances* file is no longer used, it still must be created.

Each line in the above files must contain a workstation name optionally followed by a user name, or the word *localhost*. Blank lines or lines that start with the # character are ignored.

Example

```
bcars146 berjean
localhost
bcars475
```

For each line, a process of the given type (MCF List, DPN NRS Population, View File List, or Passport NRS Population) will be started locally, if the workstation name is *localhost*, or on the specified workstation. The user name to be used when starting the process on a different workstation will be the user name as specified in the instances file or the same user name that started the *sisauto* tool. In order for this to work, the user running the *sisauto* tool must be allowed on the other workstations. These files do not require further editing if there are no changes in the number of parallel processes being run.

It is therefore possible to have the following configuration:

- The *sisauto* tool is running locally.
- 5 MCF List processes: 1 locally, 2 on workstation X and 2 on workstation Y
- 3 DPN NRS Population processes: 1 locally, 1 on workstation X and 1 on workstation Y
- 3 View File List processes: 1 locally, 1 on workstation X and 1 on workstation Y
- 2 Passport NRS Population processes: 1 on workstation X and 1 on workstation Y

See the figure “Service Integrity Audit workstation configuration example” (page 54) for an example of the above configuration. Changes made to these configuration files will automatically take effect the next time the tool is run. When the tool is configured to make use of multiple workstations, the data (SIS_DATA_DIR) and work (SIS_WORK_DIR) directories must be NFS mounted on all workstations.

The number of parallel processes that should be configured depends on many variables. For example, number of modules in the network, size of the modules, frequency of service data changes, number of workstations and the amount of workstation memory. These are only a few examples of variables that will determine the number of processes that can be run.

> ***Note:*** If any of these files are not present, *sisauto* will create them and set one line in the file to *localhost*.

Running the Service Integrity Audit tool

The Service Integrity Audit tool can be executed from a command line (*sisauto*) or from the textual user interface (*sisautui*).

Specifying PMs

For DPN, the tool accepts PMs specified in one of the following ways:

- All the PMs in the NCS hierarchy

  This is the default mode. All PMs reporting to the NCS OA or to any OA that exists in the hierarchy of that OA. This is referred to as *all PMs mode*.

- All the PMs specified in a file

  A file can be specified which contains a list of PMs. This is referred to as *file mode*.

- A specific PM

  A specific PM can be specified. This is referred to as *PM mode*.

- Problem PMs

  A list of PMs where a problem occurred while determining the valid MCF or populating NRS in the previous execution of the tool. The list is generated automatically and includes all the PMs that were not processed in the previous execution of the tool. This is referred to as *restart mode*.

Specifying EMs

For Passport, the Service Integrity Audit tool accepts EMs specified in one of the following ways:

- All the EMs in the group

  This is the default mode. All EMs which are part of the group. This is referred to as *all EMs mode*.

- All the EMs specified in a file

  A file can be specified which contains a list of EMs. This is referred to as *file mode*.

- A specific EM

  A specific EM can be specified. This is referred to as *EM mode*.

- Problem EMs

  A list of EMs where a problem occurred while determining the valid View File or populating NRS in the previous execution of the tool. The list is generated automatically and includes all the EMs that were not processed in the previous execution of the tool. This is referred to as *restart mode*.

Service Integrity Audit tool command line tool (sisauto)

The *sisauto* tool can be executed from the command line interface or from a cron job. The */opt/MagellanNMS/bin/siscron* script should be used if you want to run *sisauto* from cron.

For return codes, see the table “sisauto return codes” (page 72).

Enter the following command syntax as one continuous command.

For DPN and Passport:

```
sisauto -ncs <destination mnemonic> <capability id>
  <password> -auth <group name> <user id> <password>
  [-f_pm <file of PMs> | -pm <pm> | -restart_pm]
  [-nmsdated_pm <NMS MCF date>]
  [-f_em <file of EMs> | -em <em> | -restart_em]
  [-nmsdated_em <NMS view file date>]
  [-nrsdate <NRS selection date>] [-nonsic]
  [-noncdpop] [-quiet] [-h]
```

For DPN only:

```
sisauto -ncs <destination mnemonic> <capability id>
  <password>
  [-f_pm <file of PMs> | -pm <pm> | -restart_pm]
  [-nmsdated_pm <NMS MCF date>]
  [-nrsdate <NRS selection date>] [-nonsic]
  [-noncdpop] [-quiet] [-h]
```

For Passport only:

```
sisauto -auth <group name> <user id> <password>
  [-f_em <file of EMs> | -em <em> | -restart_em]
  [-nmsdated_em <NMS view file date>]
  [-nrsdate <NRS selection date>] [-nonsic]
  [-noncdpop] [-quiet] [-h]
```

where:

`-ncs <destination mnemonic> <capability id> <password>`

Mandatory option for NCS authentication when listing the MCFs from the various PMs or when performing NRS populations. This option is also used when the *-f_pm*, *-pm*, and *-restart_pm* options are not specified. In this case, all the PMs reporting to the NCS OA represented by the <destination mnemonic> parameter or reporting to any OA that exists in the hierarchy of the NCS OA are considered. This is referred to as *all PMs* mode.
Only OAs of type OA, OABU, CF and CFBU are supported with *all PMs* mode.

<destination mnemonic>	NCS mnemonic of the MDI interface
<capability id>	NCS capability id
<password>	Password for the NCS capability id

Note: The above parameters are not case sensitive.

`-auth <group name> <user id> <password>`

Option used for the authentication to Passport modules (EMs) when listing the view files and performing the NRS populations.

This option is also used to determine all the EMs under the specified group when the -f_em, -em and -restart_em options are not specified. This is referred to as the all EMs mode.

`<group name>`	Passport group name
`<user id>`	User ID
`<password>`	Password for User ID

The <password> parameter is case sensitive; the <group name> and <user id> parameters are not.

-f_pm <file of PMs>

Option to specify a file containing a list of PMs. This is referred to as the PM file mode.

The file must contain the PM mnemonics, one per line. These mnemonics are not case sensitive but the file name is.
File name example: /localdisk/admin/pm.list

`-pm <PM>`

Option to specify a particular PM (for example, r34). This is referred to as the PM mode. The <PM> is a specific PM mnemonic.

The <PM> parameter is not case sensitive.

-restart_pm

Option to specify that all the PMs where a problem occurred while determining the valid MCFs or populating NRS in the previous execution of the tool, are considered. This list also includes all the PMs that were not processed in the previous execution of the tool. This is referred to as PM restart mode. NSIC and/or NCD Population are run independently of this option.

This list of PMs is kept internally.

`-nmsdated_pm <NMS MCF date>`

Option used to select the NMS MCF. The NMS dated MCF with a date closest to, but not exceeding, the specified date is selected. This option is mandatory only when the configuration variable SIS_MCF_DETERMINATION_MODE is set to 2, 3 or 4 and variable SIS_NMS_MCF_MODE is set to 3. This option cannot be specified if these variables are set to different variables.

The <NMS MCF date> is in the format yymmdd. See “Date Convention” (page 27) for more information on the date format.

`-f_em <file of EMs>`

Option to specify a file containing a list of EMs. This is referred to as the EM file mode.

The file must contain the EM mnemonics, one per line. These mnemonics are not case sensitive but the file name is.
File name example: /localdisk/admin/passport.list

`-em <EM>`

Option to specify a specific EM (for example, node43). This is referred to as the EM mode.

The <EM> parameter is not case sensitive.

`-restart_em`

Option to specify that all the EMs where a problem occurred while determining the valid View File, or populating NRS in the previous execution of the tool, are considered. This list also includes all the EMs that were not processed in the previous execution of the tool. This is referred to as EM restart mode. NSIC and/or NCD Population are run independently of this option.

This list of EMs is kept internally.

`-nmsdated_em <NMS view file date>`

Option used to select the NMS view file. The NMS dated view file with a date closest to, but not exceeding, the specified date is selected. This option must be specified when the configuration variable SIS_VIEWFILE_DETERMINATION_MODE is set to 2 and variable SIS_NMS_VIEWFILE_MODE is set to 3. This option cannot be specified if these variables are set to different values.

The <NMS view file date> is in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-nrsdate <NRS selection date>`

Option to specify a date to be used when selecting the NRS files when running NSIC and when populating NCD.

The <NSR selection date> is in the format yymmdd. See “Date Convention” (page 27) for more information on the date format.

`-nonsic`

Option to specify that the NSIC checks should not be executed.

`-noncdpop`

Option to specify that the NCD population should not be performed.

`-quiet`

Option to suppress the generation of information messages. Information messages will not be sent to standard error or a log file.

`-h`

Option to display command line usage information.

> ***Note:*** If the *-nonsic* and *-nrsdate* options are not specified, activation date 791231 (December 31, 2079), meaning “the latest NRS file available for each module”, is used by default. If the -noncdpop and -nrsdate options are not specified, the date 791231 is also used to select the NRS files to be populated into NCD.

Table 1
sisauto return codes

Return code	Description
0	Successful completion. All the MCF Lists, View File Lists, and NRS Populations were successful and NSIC did not find any errors or warnings. The NRS files were populated successfully to NCD.
1	Error. Some of the MCF Lists, View File Lists, or NRS Populations were not successful, or errors were produced by NSIC, or the NCD Population found some duplication, or some major error(s) occurred with the tool.
2	User requested termination. A termination signal was received by the tool, for example, Control-C. *sisauto* terminated properly and the *-restart*_pm or -restart_em options can be used to restart the tool.
3	The tool is currently running elsewhere. This tool does not allow concurrent sessions to be running unless the SIS_WORK_DIR parameter (as defined in the SIS configuration file) is different for each session.
4	Warnings. All the MCF Lists, View File Lists, and NRS Populations were successful but warnings were produced by NSIC.

The following is an example of an sisauto session to process DPN and Passport modules. The value of *NMS priority mode*, that is assigned to MCF determination mode, is the same as *NMS mode*. The variable NAS MCF mode is no longer used.

```
/opt/MagellanNMS/bin/sisauto -ncs mdi id pw -auth grp uid
pwd

16:03:32 NOTE: The sisauto tool is starting.
Date: 1994-03-08.
16:03:33 NOTE: Following are the configuration options:
```

```
For DPN modules:
    Authentication parameters:MDI ID XXXXXXXX.
    PM selection mode:       All the PMs in the NCS OA
                             hierarchy of MDI.

    MCF determination mode:  NMS priority mode.
        NMS MCF mode:        MCF with latest bundle
                             date as of 960308 (today).
        NAS MCF mode:        MCF with highest bundle
                             number.

For Passport modules:
    Authentication parameters:GRP UID XXXXXXXX.
    EM selection mode:       All the EMs in Group GRP.
    View file determination mode:NMS.
     NMS view file mode:     View file with date as of
                             960308 (today).
NSIC execution:              Yes.
    NRS selection date:      791231.
NCD population:              Yes. (database: NCDDB)
    NRS selection date:      791231.
16:03:34 NOTE: Generating a list of all the PMs in the NCS
OA hierarchy of MDI.
16:03:42 NOTE: Generating a list of all the EMs in group
GRP.
16:04:17 NOTE: There are 7 out of 7 requests still to be
processed.
16:07:23 ERROR: The view file list failed for EM PP77.
Refer to the log file for details.
16:13:55 NOTE: There are 1 out of 7 requests still to be
processed.
16:16:18 NOTE: Starting to execute the NSIC checks.
16:17:44 ERROR: Some errors/warnings were found by NSIC.
Here is a summary:
   Module  Warnings  Errors  Total
   ------  --------  ------  -----
   PP14       0         1      1
   R70        0         1      1
           --------  ------  -----
```

```
              0         2      2
16:17:45 NOTE: Starting to populate the NCD database with
the NRS files.
16:18:52 NOTE: The NCD population completed successfully.
No errors/warnings found.
16:18:53 NOTE: File /opt/MagellanNMS/data/sis/data/
sisauto.19960308160330.ncdpop contains all NCD population
messages.
16:18:53 NOTE: File /opt/MagellanNMS/data/sis/data/
sisauto.19960308160330.nsic_unsorted contains all NSIC
messages as generated by NSIC.
16:18:54 NOTE: File /opt/MagellanNMS/data/sis/data/
sisauto.19960308160330.nsic_sorted contains all NSIC
messages sorted by module.
16:18:55 NOTE: File /opt/MagellanNMS/data/sis/data/
sisauto.19960308160330.validmcf contains the valid MCFs
generated.
16:18:55 NOTE: File /opt/MagellanNMS/data/sis/is_data/
sisauto.19960308160330.validviewfile contains the valid
view files generated.
16:18:56 NOTE: File /opt/MagellanNMS/data/sis/data/
sisauto.19960308160330.act contains the activation list
generated.
16:18:57 NOTE: Log file /opt/MagellanNMS/data/sis/data/
sisauto.19960308160330.log contains all messages
generated.
16:18:58 NOTE: The sisauto tool has completed normally.
```

Service Integrity Audit textual interface tool (sisautui)

The *sisautui* tool is a simple textual interface that can be executed from the command line or selected from the Preside Multiservce Data Manager (MDM) window. For DPN Devices you select Configuration -> DPN Devices -> Network Reporting System -> Service Integrity Audit. Port Passport Devices, you select Configuration -> Passport Devices -> Network Reporting System -> Service Integrity Audit If the Service Integrity Audit tool is started from the Preside MDM Toolset, *sisauto* is configured to run locally.

For DPN, you are prompted for the NCS parameters and PM selection mode.

For Passport, you are prompted for the authentication parameters and the EM selection mode.

You are then asked whether or not to execute the NSIC checks and whether or not to populate the NRS files into NCD; then the sisauto tool is run with these parameters.
The defaults specified in the sis.cfg file will appear in () brackets. Press return to accept the default values or to enter the new value.

For return codes, see the table “sisautui return codes” (page 77).

The following is the command syntax for *sisautui*:

```
/opt/MagellanNMS/bin/sisautui
  [-moduletype <module type>]
  [<host name> [<user name>]][-quiet] [-h]
```

where:

`-moduletype <module type>`

Option used to determine the type of modules being selected. The <module type> is one of Dpn, Passport or Both. If the option is not specified, the user is prompted for it.

`<host name>`

The name of the workstation where the *sisauto* tool is to be executed. If the <host name> is not specified, the *sisauto* tool will be executed locally.

`<user name>`

The user name to be used when executing the *sisauto* tool remotely. If the host name is specified but the user name is not, the user name will be the same as the one who started the *sisautui* tool.

`-quiet`

Option to suppress the generation of information messages by *sisauto*.

`-h`

Option to get the command line syntax and examples.

Example

The following is an example of an *sisautui* session for Passport.

```
/opt/MagellanNMS/bin/sisautui  -moduletype PASSPORT

Service Integrity Audit (Passport modules)
--------------------------------------------------

Enter the authentication parameters:

   Group name(GRP):
   User ID (ID):
   Password: XXXXXXXX

EM selection mode:

   1- All the EMs in group GRP
   2- List of EMs
   3- Specific EM
   4- Restart
   q- Quit

Enter your selection [1-4,q] (1): 3
Enter the EM name: pp301

Do you want to execute the NSIC checks? [y,n] (y):

Do you want to populate the NCD database? [y,n] (y):

The following parameters will be used:
For Passport modules:
   Authentication parameters:       GRP ID XXXXXXXX.
   EM selection mode:               EM pp301.
NSIC execution:                     Yes.
NCD population:                     Yes.

Do you want to start the execution? [y,n] (y):
```

```
(all the messages from sisauto)

Type c to continue or q to quit [c,q] (q):
```

Table 2
sisautui return codes

Return code	Description
0	Successful completion. The sisauto tool was called. The success or failure of the sisauto tool is not reflected here.
1	Error. A validation error occurred while retrieving some of the parameters.
2	User requested termination. A termination signal, Control-C, was received by the tool. The tool has terminated.

Service Integrity Audit output

The tool produces the following output:

- standard error
- valid MCF file
- valid View File file
- activation list file
- log file
- file with NSIC errors (sorted by module)
- file with NSIC errors (as generated by NSIC)
- NCD population file

Standard error The Service Integrity Audit tool prints progress messages, error messages and a summary of NSIC errors to the screen. A progress message is displayed approximately every ten minutes and indicates the number of requests still to be processed. At the end of the session a message is displayed indicating where to find the activation list files, log file and the NSIC error files. A termination message is also displayed.

Valid MCF file For each PM handled by *sisauto*, this file contains the name of the PM and the MCF that was found to be valid.

This file is sorted by PM and is used to determine which MCF was found to be the *valid* MCF for each PM.

If an error occurs while listing the MCFs from the PM, that PM will not be included in this file. However, if for a PM the NRS population fails or NSIC errors were found, the PM will be included in this file.

This file is generated in the directory referred to by the SIS_DATA_DIR variable in the SIS Service Integrity Audit configuration file. The format is *sisauto.<date>.validmcf,* where *<date>* is in the format *yyyymmddhhmmss*.

The purpose of this file is for the user to be able to tell which MCF was found to be the valid one for each PM. This is important when making sure that, based on the values set in the configuration file, the correct MCF was considered the valid one by the tool.

Valid View File file This file contains, for each EM handled by the current execution of the tool, a line that contains the EM name and the view file that was found to be the valid one. This file is sorted by EM.

If errors occurred while listing the view files from the EM, that EM will not be included in this file. However, if for an EM, the NRS population fails, NSIC errors were found, or the population to NCD failed, the EM will be included in this file.

This file is generated in the directory referred to by the SIS_DATA_DIR variable in the Service Integrity Audit configuration file. Its name is:

```
sisauto.<date>.validviewfile
```

where *<date>* is in the format *yyyymmddhhmms*s.

The purpose of this file is for the user to be able to tell which view file was found to be the valid one for each EM. This is important when making sure that, based on the values set in the configuration file, the correct view file was considered the valid one by the tool.

Activation list file Since the valid MCF and valid View File files are potentially missing some modules, the activation list file is produced by merging all the valid MCF files and all the valid View File files found in the directory referenced by the SIS_DATA_DIR variable in the Service Integrity

Audit configuration file. For each module, the entry is taken from the latest valid MCF file or valid
View File file where it is present.

This file is used to determine which MCF/View File should be activated for each module.

This file is generated in the directory referred to by the SIS_DATA_DIR variable in the configuration file. The format is *sisauto.<date>.act* where *<date>* is in the format yyyymmddhhmmss.

Log file This file contains all the messages that are printed to standard error as well as messages generated during the MCF List, DPN NRS Population, View File List, Passport NRS Population, NSIC and NCD Population phases. It is generated in the directory referred to by the SIS_DATA_DIR variable in the configuration file.
The log file name has the format *sisauto.<date>.log*, where *<date>* is in the format yyyymmddhhmmss.

NCD population file This file contains the output generated by the *ncdpop* tool. It is generated in the directory referred to by the SIS_DATA_DIR variable in the Service Integrity Audit configuration file. Its name is:

```
sisauto.<date>.ncdpop
```

where *<date>* is in the format yyyymmddhhmmss.

File with NSIC errors, sorted by module This file contains a summary of the NSIC errors and all the NSIC errors sorted by module. For example, all the messages for PM A1 are printed followed by the messages for EM A2.

This file is called *sisauto.<date>.nsic_sorted*, where *<date>* is in the format yyyymmddhhmmss. The file is generated in the directory referred to by the SIS_DATA_DIR variable in the Service Integrity Audit configuration file.

File with NSIC errors, unsorted This file contains the output of NSIC errors without any sorting.

This file is called *sisauto.<date>.nsic_unsorted*, where *<date>* is in the format yyyymmddhhmmss. The file is generated in the directory referred to by the SIS_DATA_DIR variable in the Service Integrity Audit configuration file.

Stopping the Service Integrity Audit tool

If the *sisauto* tool was started from the NT Toolset using the ASCII user interface or the UNIX command line, type *Control-C* to stop the session.

If the *sisauto* tool was started from a cron job, is running in the background or was started by another user, use the following procedure to stop the tool.

1 Try starting the tool again: Enter:

```
/opt/MagellanNMS/bin/sisauto
```

If the tool is currently running, the following message will be displayed:

```
14:17:13 ERROR: The sisauto tool is already running.
It was started on 1994-05-02 09:19:22 by user fred on
host NMS1. The process ID is 532.
```

2 The tool can be stopped by user fred on host NMS1 by specifying the process ID: Enter:

```
kill 532
```

It may take a few minutes for the tool to stop.

Note: Do not use the kill -9 <process ID> command. The tool will not be able to perform the necessary clean-up and create the output files in the SIS_DATA_DIR directory.

Recovering from abnormal termination

If sisauto is stopped abnormally, for example, using the kill -9 <process id> command or a system reboot occurs while the tool is running, the next time the tool is started it will think that it is already running and produce the following message:

```
14:17:13 ERROR: The sisauto tool is already running.
It was started on 1994-05-02 09:19:22 by user fred on
host NMS1. The process ID is 532.
```

This message is displayed because the *.sisauto.lock* file is still present in the SIS_WORK_DIR. This file should be removed using the UNIX *rm* command.

> ***Note:*** The *.sisauto.lock* file must not be removed if the sisauto tool is running. Removing this file will allow another sisauto session to start and may cause the first session to run indefinitely.

Chapter 7
NRS/NCD Population Manager

This chapter describes the NRS/NCD Population Manager tool (*popmgr*), which you can use to simplify the population of the NRS and NCD databases. In this chapter, you can find the following information:

About the NRS/NCD Population Manager

The NRS/NCD Population Manager tool (*popmgr*) periodically checks the BACKUP_DISK or NMS_DISK directory for the arrival of new MCFs. The NRS population tool then populates the MCF data into NRS database and automatically executes the NCD population tool to populate NRS data into the NCD database.

Provisioning changes will be populated automatically into the NRS and NCD databases. This reduces the latency between a download and the availability of data in NRS and NCD and allows NSIC to be run sooner after provisioning activities.

This tool does not support Passport modules. It will only populate the DPN NRS data into the given NCD database. If MCFs are manually populated into the NRS database or populated using the Service Integrity Audit tool, the population management tool will automatically detect the new NRS data and populate it into the NCD database if the tool is configured correctly.

Popmgr model

Each *popmgr* process supports one NCD database. Multiple *popmgr* processes with different configurations can be deployed to support multiple NCD databases for multiple views of the network.

The *popmgr* command executes four processes: MCF List, NRS Population, NRS List, and NCD Population.

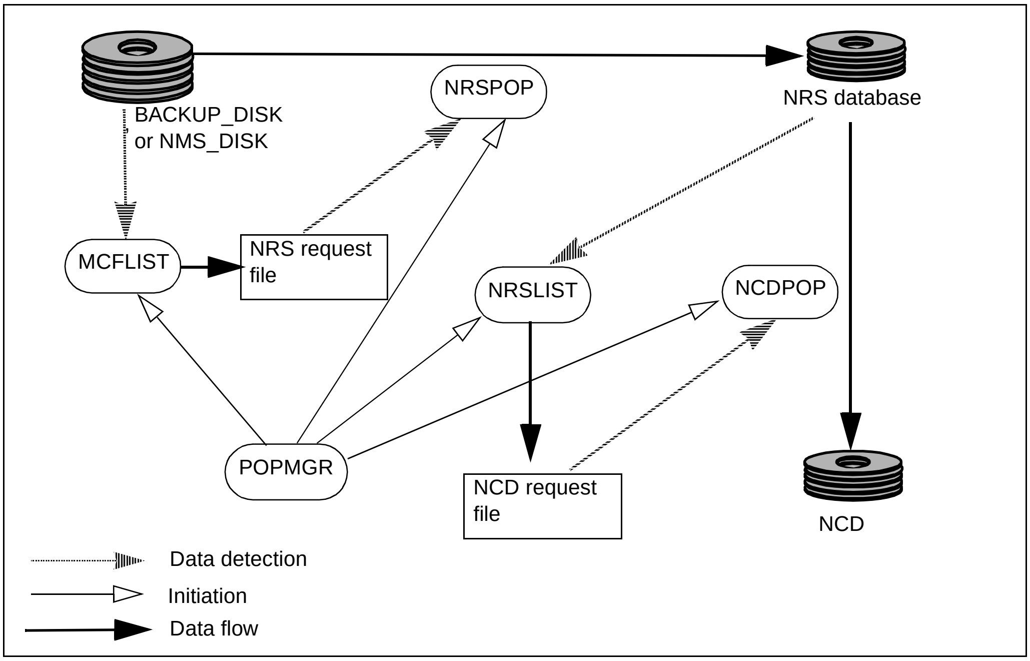
MCF List periodically monitors the arrival of the new MCFs for each MCF directory or location. It selects the latest keyed or dated MCFs from the specified location based on the MCF selection criteria specified on the command line.

For each NRS Population request, NRS Population takes care of populating the MCF data into the NRS database. This process will idle when there are no more NRS Population request files to be processed.

NRS List periodically monitors the arrival of the new NRS data files in the NRS database. It selects the latest keyed or dated NRS data files based on the same selection criteria as MCF List.

For each NCD Population request, NCD Population takes care of populating the NRS data into the NCD database. This process will idle when there are no more NCD Population request files to be processed.

Figure 3
popmgr data flow



Configuration

See the following sections for information on what you need to configure before running the popmgr tool:

- “Network Reporting System” (page 86)
- “Network Configuration Database” (page 86)
- “Provisioning File Access Server” (page 86)
- “Configuration file” (page 86)
- “Instance files” (page 90)
- “Popmgr mapping file” (page 92)

Network Reporting System

The Network Reporting System (NRS) must be configured before using the popmgr tool. When the tool is configured to make use of multiple workstations, the NRS configuration must be the same on all the workstations. The NRS data directory must be NFS mounted, as well as the NRS RDF directory and the NRS configuration file must refer to these directories.

For more information on configuring NRS, see “Installation and configuration” (page 29).

Network Configuration Database

The Network Configuration Database (NCD) must be configured before using this tool if the -noncdpop option is not specified in the command line. The specified NCD database must be created and the NCD server must be running.

For more information on NCD, see 241-6001-308 *Preside MDM Network Configuration Database Administrator Guide*.

Provisioning File Access Server

Provisioning File Access Server (PFAS) must be configured before using this tool. If the location mode is NMS_DISK or BACKUP_DISK, the corresponding directory must be NFS mounted and the PFAS configuration file must refer to this directory.

For more information on PFAS, see 241-6001-304 *Preside MDM Configuration Management Administrator Guide*.

Configuration file

The configuration file is called *PMT.cfg* and can be found in the */opt/ MagellanNMS/cfg* directory. This file can be copied to the MagellanNMS directory under the user’s home directory or modified directly. The tool looks for this file in the user specific directory first. If not found, then in the */opt/ MagellanNMS/cfg* directory.

The -*cfg* option specified on the command line allows you to specify the *configuration* file name and location if the defaults are not used.

When the tool is configured to make use of multiple workstations, only the configuration file where the *popmgr* tool is running is used.

The following is a list of global variables contained in the configuration file. These variables must be changed to suit the network configuration.

POPMGR_DATA_DIR

popmgr requires a read/write directory to store output files produced. This directory must be mounted by all the workstations used to run this tool. The default value is */opt/MagellanNMS/data/pmt/data*.

For example, /opt/MagellanNMS/data/pmt/data

POPMGR_WORK_DIR

popmgr requires a read/write directory where work and temporary files used or created by the tool can be found. The directory must be mounted by all the workstations used to run this tool. The default value is */opt/MagellanNMS/data/pmt/work*.

For example, /opt/MagellanNMS/data/pmt/work

POPMGR_PERIOD

This variable defines the period (in seconds) to check the specified location for newly arrived MCFs. The default value is 600.

For example, 600

POPMGR_MODE_LOCATION

This variable defines the mode and directory to be monitored. It is in the format <mode>[:<location>], where <mode> is BACKUP_DISK or NMS_DISK and <location> is a directory. The default mode is BACKUP_DISK. The default location for NMS_DISK mode is the value of UNIX_ROOTDIR in the *PFA.cfg* file; for BACKUP_DISK mode it is the value of UNIX_BACKUP_DIR in the *PFA.cfg* file. The directory must be PFA accessible.

For example, BACKUP_DISK

POPMGR_OUTPUT_FILE_SETS_KEEP_COUNT

The *popmgr* tool produces a set of files (log, activation list, errors) in the POPMGR_DATA_DIR directory every time the tool is executed. This variable defines how many sets of files will be kept. All the files generated on the same day are considered in the same set. The latest sets are kept and the others are deleted. This allows the POPMGR_DATA_DIR directory to be maintenance free. The default value is 100.

For example, 100.

POPMGR_NCD_TARGET

This variable defines the target NCD database to be populated. This variable is optional if the -*noncdpop* option is specified.

For example, ncddb

POPMGR_NCD_MODE

This variable defines the NCD population mode. Values can be UPDATE or REPLACE. The default value is REPLACE.

For example, REPLACE

POPMGR_NCD_FORCE

This variable defines whether or not to use the FORCE option when populating the NCD database. The default value is NO.

For example, NO

POPMGR_EXTERNAL_DNA_FILE

This variable is used when the *popmgr* tool calls the NCD population tool. With the NCD Populator, it is possible to specify a file containing a list of DNAs that are external to the DPN-100 network.

This parameter can be used to specify such a file or it can be left blank.

For example, /usr/data/dna_file

POPMGR_EXTERNAL_RID_FILE

This variable is used when the *popmgr* tool calls the NCD population tool. With the NCD Populator, it is possible to specify a file containing a list of RIDs that are external to the DPN-100 network.

This parameter can be used to specify such a file or it can be left blank.

For example, /usr/data/rid_file

Example

The following is an example of an *PMT.cfg* file. Comments lines indicated by (...), have been removed but appear in the actual file.

```
# ---------------------------------------------------------------------
#
#   Module Name:      PMT.cfg
#
#   Description:      This file contains the configuration variables
#                     for the NRS/NCD POPulation ManaGeR tools.
#
#                     Line format is: POPMGR <variable> <value>
#
#                     Comment lines start with the ‘#’ character.
#                     The set of configuration variables and their
#                     functions are described in this file.
#
(...)
#POPMGR POPMGR_DATA_DIR            <full path of directory>
#POPMGR POPMGR_DATA_DIR            /opt/MagellanNMS/data/pmt/data
(...)
#POPMGR POPMGR_WORK_DIR            <full path of directory>
#POPMGR POPMGR_WORK_DIR            /opt/MagellanNMS/data/pmt/work
(...)
#POPMGR POPMGR_PERIOD              <the number of seconds>
#POPMGR POPMGR_PERIOD              600
(...)
```

```
#POPMGR POPMGR_MODE_LOCATION       <mode>[:<location>]
#POPMGR POPMGR_MODE_LOCATION       BACKUP_DISK
(...)
#POPMGR POPMGR_OUTPUT_FILE_SETS_KEEP_COUNT   <1, 2147483647>
#POPMGR POPMGR_OUTPUT_FILE_SETS_KEEP_COUNT   100
(...)
#POPMGR POPMGR_NCD_TARGET          the name of NCD database>
#POPMGR POPMGR_NCD_TARGET          ncd_DB
(...)
#POPMGR POPMGR_NCD_MODE            <REPLACE | UPDATE>
#POPMGR POPMGR_NCD_MODE   REPLACE
(...)
#POPMGR POPMGR_NCD_FORCE           <YES | NO>
#POPMGR POPMGR_NCD_FORCE           NO
(...)
#POPMGR POPMGR_EXTERNAL_DNA_FILE   [<full path of a file>]
#POPMGR POPMGR_EXTERNAL_DNA_FILE
(...)
#POPMGR POPMGR_EXTERNAL_RID_FILE   [<full path of a file>]
#POPMGR POPMGR_EXTERNAL_RID_FILE
#
```

Instance files

Instance files are used to control the number of parallel processes that are performing the MCF List, NRS List, NRS Population and NCD Population tasks. The instance files *.mcflist.instances*, *.nrslist.instances*, *.ncdpop.instances*, and *.nrspop.instances*, should be created in the directory referred to by the POPMGR_WORK_DIR variable in the *PMT.cfg* file. If not, *popmgr* will create the corresponding instance files with default value, *localhost*, in each of the files.

The number of processes configured to run in parallel will depend on many factors. For example, number of PMs in the network, size of the PMs, frequency of service data changes, number of workstations and the amount of workstation memory.

Each line in the above files must contain a workstation name optionally followed by a user name, or the word *localhost*. Blank lines or lines that start with the # character are ignored.

Example
```
bcars706 joe
localhost
bcars475
```

For each line, a process of the given type (MCF List, NRS List, NRS Population, or NCD Population) will be started locally, if the workstation name is *localhost,* or on the specified workstation. The user name to be used when starting the process on a different workstation will be the user name as specified in the instances file or the same user name that started the *popmgr* tool.

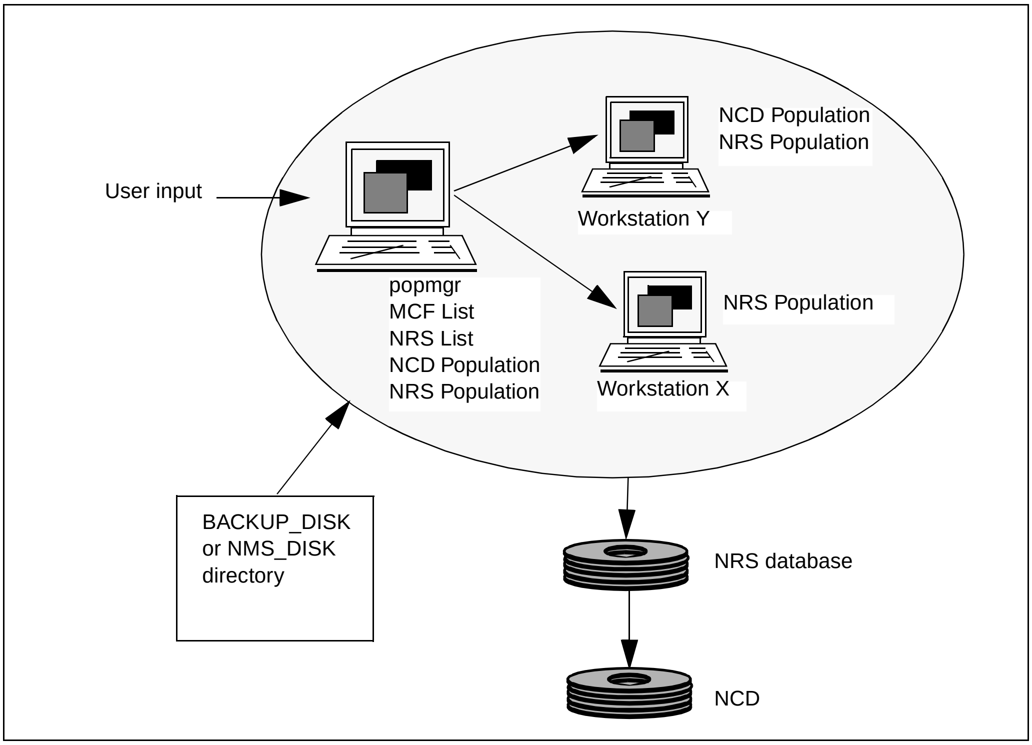
Note: Only one location (BACKUP_DISK directory or NMS_DISK directory) can be specified for MCF List to monitor. The specified site will be monitored for the latest MCFs. If more than one name is specified in this file, the *popmgr* tool will use the first one and ignore the rest.

It is therefore possible to have the following configuration:

- popmgr tool is running locally
- 1 MCF List process: 1 locally
- 1 NRS List process: 1 locally
- 2 NCD Population processes: 1 locally and 1 on workstation Y
- 3 NRS Population processes: 1 locally, 1 on workstation X, and 1 on workstation Y.

See the figure “popmgr workstation configuration example” (page 92) for an example of the above configuration.

Figure 4
popmgr workstation configuration example



Changes made to any of these configuration items will automatically take effect the next time the tool is run.

When the tool is configured to make use of multiple workstations, the data (POPMGR_DATA_DIR) and work (POPMGR_WORK_DIR) directories must be NFS mounted on all workstations.

Popmgr mapping file

The mapping file, *popmgr.mappin*g, under POPMGR_WORK_DIR is used to map the NAMSID to PM name. You can manually update the file before starting *popmgr* or *popmgr* will update the mapping file as necessary (such as for a new NAMSID). If no mapping file exists under

POPMGR_WORK_DIR, *popmgr* will create an empty file and then update the mapping file. Each line in the file has the format: <NAMSID> <PM name>. Blank lines or lines that start with a # character are ignored.

Example

```
# <NAMSID> <PM name>
4034 R34
4011 A20
```

Running popmgr

The *popmgr* process can be started by entering the parameters on the command line.

MCF/NRS data file selection

For each specified directory or location, *popmgr* periodically monitors the arrival of the new MCF/NRS data files.

Only one MCF List instance is used to monitor the BACKUP_DISK or NMS_DISK site. Therefore, only one MCF List request file will be generated which contains the upload mode (BACKUP_DISK or NMS_DISK; the default is BACKUP_DISK) and the location (directory of BACKUP_DISK or NMS_DISK). For BACKUP_DISK, the default location is specified by the variable UNIX_BACKUP_DIR in the PFAS configuration file, *PFA.cfg* under the */opt/MagellanNMS/cfg* directory. For NMS_DISK, the default location is specified by the variable UNIX_ROOT_DIR in the PFAS configuration file.

The *popmgr* tool processes MCF/NRS data files that fall into one of the following categories:

- Latest MCF/NRS data files

  Only the latest MCF/NRS data per PM. Using the current view mode, popmgr select the latest MCF/NRS data (one per PM). All data files will be populated into the NRS/NCD database that reflects the latest view of network. This is the default mode.

- Dated or keyed

  This mode selects the latest dated or keyed MCF/NRS data per PM. Only the latest data files will be populated into the NRS/NCD database. Both dated and keyed are supported, but not at the same time. User specified mode is not supported.

Depending on the configuration and the MCF(s) retrieved, *popmgr* creates an NRS Population request or nothing. In order to populate the MCF into the NRS database, the PM name is required for nrspop. *popmgr* will automatically generate/update the mapping file to map PM names and NAMSID.

Command syntax

The following command syntax is used to start the *popmgr* tool. Enter the following command syntax as one continuous command. To display online help for this command, use the *-h* option.

If the *-dated* or *-keyed* options are not specified on the command line, the latest MCF/NRS data per PM will be selected.

```
/opt/MagellanNMS/bin/popmgr -ncs <destination
mnemonic>
  <capability id> <password> | -manual
  [-dated <date> | -keyed <key>] [-log [<log file>]]
  [-restart] [-noncdpop] [-quiet] [-h]
```

where:

-ncs <destination mnemonic> <capability id> <password>

Option for NCS authentication when performing NRS populations. This option cannot be used with the *-manual* option.

<destination mnemonic>	NCS mnemonic of the MDI interface
<capability id>	NCS capability id
<password>	NCS capability id password

-manual

Option to use without NCS authentication when performing NRS population. This option cannot be used with the -*ncs* option.

-dated <date>

Option used to select the MCF/NRS data. The MCF/NRS data with a bundle matching the specified date is selected. This option cannot be specified with the -*keyed* option.

<date> A valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-keyed <key>

Option used to select the MCF/NRS data. The MCF/NRS data with the highest version of the specified key is selected. This option cannot be specified with the -*dated* option.

-log [<log file>]

Option to have all messages recorded in a log file. The log file will be updated under the POPMGR_DATA_DIR directory.

<log file> The log file is named *popmgr.log* unless otherwise specified.

-restart

Option to restart the tool from last stop point. All uncompleted requests will be reprocessed if a problem occurred while selecting MCF/NRS data files(s), populating NRS, or populating NCD in the previous execution of the tool. If the user wants to apply the same configuration as the previous command, they must specify exactly the same options and configuration as well as, specify the -*restart* option on the command line. Otherwise, *popmgr* will use the default options and configuration. For example, data selection will be the latest, without the -*quiet* option and -*noncdpop*, etc.

-noncdpop

Option to specify that NCD population should not be executed. NCD population is executed by default.

-quiet

Option to suppress the generation of information messages.

-h

Option to display command line usage information.

Table 3
popmgr return codes

Return code	Description
0	The tool has finished normally without error.
1	The tool has finished normally but errors were found.
2	The user terminates the tool.

Note: *popmgr* usually runs all the time. The tool only exists in the following situations:
1. command line syntax error
2. terminated by user (Control C or kill)
3. computer reboot.

Example

The following command will populate the latest MCF data files with a key of KEY into the NRS database (if needed). It will then populate the latest NRS data files with a key of KEY into the NCD database. All messages are stored in a file called *log_file* and in log files in the POPMGR_WORK_DIR and POPMGR_DATA_DIR directories.

```
/opt/MagellanNMS/bin/popmgr -ncs mdi id pw -keyed KEY
-log log_file
```

Output

The *popmgr* tool produces the following output:

- OAM log
- log file

OAM log When a fatal error occurs (such as one of the instance processes has stopped), the *popmgr* tool sends a message to the OAM log to notify the user. Corrective action should be taken.

Log file The master log file specified by the user with -*log* option contains all the messages generated to standard error as well as messages generated during the MCF List, NRS Population, and NCD Population if *popmgr* is terminated. This file is generated in the directory referred to by the POPMGR_DATA_DIR variable in the *popmgr* configuration file. The name of the log file is *popmgr.log* unless otherwise specified by the -*log* option on the command line.

When *popmgr* is running, a large number of log files (instance log files) under the POPMGR_WORK_DIR directory are generated by MCF List (*mcflist.log.<number of instance>*), NRS Population (*nrspop.log.<number of instance>*), NRS List (*nrslist.log.<number of instance>*), NCD Population (*ncdpop.log.<number of instance>*). However these files will be merged into master log file if the *popmgr* tool is terminated. While *popmgr* is running, if *nrspop* or *ncdpop* fails, the corresponding log files *popmgr.nrspop.<time>.log* or *popmgr.ncdpop.<time>.log*, are generated under POPMGR_DATA_DIR directory.

You should periodically check the master log file and instance log files under POPMGR_WORK_DIR and the log files under POPMGR_DATA_DIR to guarantee that the *popmgr* tool is running normally and make any necessary corrections.

Stopping popmgr

If the *popmgr* tool was started from a UNIX command line, type *Control-C* to stop the session.

If the *popmgr* tool was started from a cron job, running in the background or was started by another user, use the following procedure to stop the tool.

1 Try starting the tool again, enter:

```
/opt/MagellanNMS/bin/popmgr -ncs ncs userid passwd
```

If the tool is currently running, the following message will be displayed:

```
14:17:13 ERROR: The popmgr tool is already running. It
was started on 1995-05-02 09:19:22 by user fred on host
NMS1. The process ID is 532.
```

2 The tool can be stopped by user fred on host NMS1 by specifying the process ID, enter:

```
kill 532
```

It may take a few minutes for the tool to stop.

Note: Do not use the *kill -9 <process ID>* command. The tool will not be able to perform the necessary clean-up and create the output files in the POPMGR_WORK_DIR and POPMGR_DATA_DIR directory.

Chapter 8
NRS-based Service Integrity Check

This chapter describes the NRS-based Service Integrity Check (NSIC), which you can used to detect errors in the service data. In this chapter, you can find the following information:

About the NRS-based Service Integrity Check

NRS-based Service Integrity Check (NSIC) is a UNIX command line application used to detect inconsistencies in the service data that could cause service problems during network operation. These inconsistencies are not detected during a normal provisioning session because data from several or even all modules in the network may be required to determine the nature of the problem.

NSIC is a diagnostic tool that does not change or affect service data in any way. It is a passive tool that is run after the service data has been generated. It provides a means of identifying inconsistencies in service data prior to activation and reduce the possibility of service disruption.

NSIC provides semantic checks for the following:

- DNA uniqueness
- MID and NAMS ID uniqueness
- module mnemonic uniqueness
- IP address uniqueness
- call redirection
- hunt groups
- direct calls
- PVC
- RID map
- PM mnemonic
- GID (gateway IDs)
- OA (NCS OAs)
- voice networking consistency
- VXID (Virtual XID)
- ATM PVCs

Prerequisites to running NSIC

The NSIC utility relies on the data files created and maintained by the Network Reporting System (NRS). NRS should be installed and configured prior to attempting to use the NSIC utility.

In order to provide the most accurate detection of potential errors, NSIC requires an NRS data file for each packet module in the network. The service data for those modules must be captured in the NRS data files to provide accurate NSIC error detection. See “NRS toolset reference” (page 235) for a description of the sisauto, nrspop and pnrspop tools which are used to create the necessary data files for each module in your network.

To simplify the use of NSIC, it is recommended you use the dated convention for MCF/View File naming. See 241-6001-304 *Preside MDM Configuration Management Administrator Guide* for a description of this MCF naming convention.

When to run NSIC

NSIC should be run any time changes are made to any of the components identified in the preceding section. In particular, when adding DNAs, adding new modules, changing member lists for hunt groups, changing direct calls, PVCs, call redirection lists or RID mappings. Note that it may not be necessary to run the checks on every service data change.

If changes are batched for your network, whereby many service data changes are entered for a scheduled activation at some point in the future, then it is advisable to complete all the data entry changes to the modules prior to performing an NSIC validation.

Service data integrity check cycle

The process for updating and checking service data is as follows:

1. Change service data using the Component Provisioning tool.
2. Download data either to the module or the MDM workstation disk.
3. Run the Service Integrity Audit tool with NSIC checks selected to populate the DPN and Passport modules. NSIC checks will be run on both the DPN and Passport modules in the database.
4. Use the Component Provisioning tool to fix any errors detected by NSIC, and repeat the cycle.

Customized network view

NSIC allows you to run the semantic checks on a customized view of the network.To do this you must generate a subset of NRS data files generated by the command line parameters. It is important that the subset of the network define completely all aspects of the feature being checked because of the potential for unresolved references to the rest of the network. Creating a subset of data files will shorten the time required to run NSIC.

The following example runs the semantic checks against a customized view of the database called myfile.

1 Generate a base list of NRS data files using the nrslist command. The -dated option will select only one file per NAMS ID. The file with the activation date closest to (without exceeding) the specified date is selected and placed in a file called myfile.

```
/opt/MagellanNMS/bin/nrslist -dated 920101 >myfile
```

2 Edit myfile and delete any NRS data files that are not required for this customized check of the network.

3 Run NSIC with the edited file as the set of input NRS data.

```
/opt/MagellanNMS/bin/nsic -nrs_list myfile
```

Running NSIC

Enter the following command syntax as one continuous command.

```
/opt/MagellanNMS/bin/nsic [<nrs_selector> | <nrs_list>]
   [<check_options>]
   [<other_options>]
   [-h]
```

where:

```
<nrs_selector>  = [-type <module_type>]
           [-pm <pm>]
           [-namsid <namsid>]
           [-bundle <bundle> | -viewfile <viewfile>]
           [-activation_date <activation_date>]
           [-dated <date> | -keyed <key>]

<nrs_list>      = -nrs_list <file>

<check_options> = [-dna_uniqueness] [-gid_uniqueness]
           [-mid_uniqueness] [-module_name_uniqueness]
           [-nodeid_uniqueness] [-Namsid_uniqueness]
           [-oa_uniqueness] [-vxid_uniqueness]
           [-ipaddr_uniqueness] [-redirection]
           [-hunt_group] [-pvc] [-direct_call] [-rid_map]
           [-call_forward] [-accounting] [-voice]
           [-external_dna_file <dna_file>]
           [-external_rid_file <rid_file>]
           [-external_ipaddr_file <ipaddr_file>]
           [-atm]

<other_options> = [-noconfirm]
           [-sorted <sorted output filename>]
           [-debug]
```

No options or the -h option prints help information.

If no feature options are specified then all checks are performed, which is the recommended way to run NSIC. The time spent by NSIC to perform all checks in a single run, is less than the total time spent to perform different checks in multiple runs.

For a description of the above parameters, see “NRS-based Service Integrity Checks (NSIC)” (page 262).

Example

The following is an example of a user request for a dated check on DNA uniqueness followed by the system response. All output is directed to standard output.

```
/opt/MagellanNMS/bin/nsic -dated 940510
-dna_uniqueness
```

The response is:

```
NOTE: The following NRS data files have been selected:
/opt/MagellanNMS/data/nrs/data/
dpn.R58.4058.99051003.990510.data
/opt/MagellanNMS/data/nrs/data/
dpn.R34.4034.99051012.990510.data

Initiate nsic checks (y/n)?
```

The system responds with all the NRS data selected and a confirmation prompt before running the checks. If the -noconfirm option is specified the checks run without confirmation.

Customized user checks

The following are examples of NSIC commands that can be used to gather specific information.

Dated access, all checks, an external DNA file is supplied.

```
/opt/MagellanNMS/bin/nsic -dated 920612
-external_dna_file ext.dnas
```

Keyed access with DNA uniqueness and PVC checks only.

```
/opt/MagellanNMS/bin/nsic -keyed R7BASE -dna_uniqueness
-pvc
```

Bundle access, all checks, do not confirm, generate messages sorted by node name in file nsic.sorted.

```
/opt/MagellanNMS/bin/nsic -bundle R7 -noconfirm
-sorted nsic.sorted
```

Dated access, generate an NRS list file, edit the file and run RID map only checks.

```
/opt/MagellanNMS/bin/nrslist -dated 941225 >mynrslist
```

```
vi mynrslist
```

```
/opt/MagellanNMS/bin/nsic -nrs_list mynrslist -rid_map
```

Viewfile access, all checks, do not confirm.

```
/opt/MagellanNMS/bin/nsic -viewfile NODER16 -noconfirm
```

Example

The following is an example of how to use NSIC to check for DNA uniqueness across multiple Passport modules.

1 Create a list of Passport modules (type ppc and ppe) and put them in the specified file, enter:

```
/opt/MagellanNMS/bin/nrslist -type '^pp' > nrs.list
```

2 Edit the file and remove all unwanted modules/view files.

```
vi nrs.list
```

3 Run NSIC.

```
/opt/MagellanNMS/bin/nsic -nrs_list nrs.list
-dna_uniqueness
```

Response:

```
NOTE: The following NRS data files have been selected:
/opt/MagellanNMS/data/nrs/data/
ppc.NODER53.2187.DMB01,full,001.800101.data
/opt/MagellanNMS/data/nrs/data/
ppc.NODER65.2201.DMB06,full,003.800101.data

Initiate nsic checks (y/n)?  y

NSIC error number 1:
Error:  Duplicate DNA X30210211870130.

Location: NODER53 FrameRelayUni/130
DataNetworkAddress/X30210211870130.

Location: NODER65 FrameRelayUni/62 DataNetworkAddress/
X30210211870130.

NSIC error number 2:
Warning: Duplicate DNAs have been found. Subsequent
messages referencing DNAs may be wrong or misleading
if the DNA is a duplicate.

Fix: Fix the duplicate DNA problems and re-run nsic.

NRS Service Data Integrity Check complete, 2 errors or
warnings detected.
```

Semantic checks

This section details the various semantic checks offered by NSIC. Each semantic check details the error message, the location and a possible fix. Although NSIC allows you to specify features (for example, call redirection, PVC) you cannot specify a particular check within that feature.

The NAMS ID uniqueness, MID uniqueness, DNA uniqueness and PVC options include checks for the Passport Frame Relay service.

The ATM checks are limited to permanent virtual circuits (PVC). In the case of virtual path terminators (VPT), NSIC only checks for consistency of the virtual path identifier (VPI) at the VPT level. NSIC does not follow the virtual path connection (VPC) and/or virtual channel connection (VCC) subcomponents provisioned under a VPT to check for consistency in the VPI.VCI values across interfaces. See "ATM PVC checks" (page 126).

The accounting checks apply to both DPN and Passport modules. All other checks support DPN modules only.

> ***Note:*** The DNA uniqueness check is always performed when call redirection, hunt group, direct call, voice networking, or PVC checks are requested. In addition, the node ID uniqueness check is always performed when doing the voice networking check.

Call forward checks

1 In Destination Call Router, all call forward DNAs must be assigned.

Error: A call forward DNA is unassigned.

Location: <pm> Destination_Call_Router DCR_Call_Forward_Group/n ForwardDNAs DNA/<dna>

Fix: Change the call forward DNA to an assigned DNA.

2 In Destination Call Router, all call forward RIDs must be assigned.

Error: A call forward RID is not assigned.

Location: <pm> Destination_Call_Router DCR_Call_Forward_Group/n ForwardDNAs DNA/<dna> Forward_RID <rid>

Fix: Change the call forward RID to an assigned RID.

Call redirection checks

1 In Access to Call Redirection, all RID redirection targets must be assigned.

Error: Access_to_Call_Redirection, RID_Redirection refers to unassigned RID.

Location: <pm> Access_To_Call_Redirection RID_Redirection RID n: <target_rid>.

Fix: Assign RID, or change reference to an assigned RID.

2 In Access to Call Redirection, the RID which is redirected must be assigned.

Warning: An unassigned RID is redirected.

Location: <pm> Access_To_Call_Redirection RID_Redirection RID n.

Fix: Assign RID, or remove redirection.

3 In Access to Call Redirection, the Direct Calls must refer to the DNA of a Call Redirection Server.

Error: Access_to_Call_Redirection, Direct_Call, does not refer to a Call Redirection Server.

Location: <pm> Access_To_Call_Redirection Direct_Call_List Server DNA n <dna>

Fix: Refer to an assigned Call Redirection Server, or assign Call Redirection Server with the given DNA.

4 In Access to Call Redirection direct call, the DNA must be in the same CUG as the Call Redirection Server.

Error: Access_to_Call_Redirection, Direct Call DNA does not have the
same CUG ID.

Server: <pm> PE/x Call_Redirection_Server DNA_CUG/<dna>.

Fix: Configure both DNAs with a common CUG ID.

5 The DNA should be referenced at least once by Access to Call Redirection in Call Redirection Server.

Warning: Call_Redirection_Server is not referenced by Access_to_Call_Redirection.

Location: <pm> PE/x Call_Redirection_Server DNA_CUG/<dna>.

Fix: Configure Access_to_Call_Redirection to reference server, or remove the server.

6 Primary DNAs must be assigned in the Call Redirection List.

Error: A Call_Redirection_List Primary DNA is not assigned.

Location <pm> Call_Redirection_List Call_Redirection_ DNAs Primary_DNAs Primary/<dna>.

Fix: Remove primary DNA, or assign it.

7 Primary DNAs must be valid in the Call Redirection List. A valid DNA is any DNA except X25 Gateway.

Error: A Call_Redirection_List Primary DNA is not valid.

Location: <pm> Call_Redirection_List Call_Redirection_DNAs Primary_DNAs Primary/<dna>.

Fix: Remove Primary DNA from the Redirection List.

8 The Call Redirection secondary DNA must be assigned.

Error: A Call_Redirection_List Secondary DNA is not assigned.

Location: <pm> Call_Redirection_List Call_Redirection_ DNAs Primary_DNAs Primary/<dna> DNA n: <dna>.

Fix: Change the secondary DNA.

9 Redirection RIDs should be assigned in the Call Redirection List.

Error: A Call Redirection RID is not assigned.

Location: <pm> Call_Redirection_list Call_Redirection_DNAs Primary_DNAs Primary/<dna> RID n:<rid>.

Direct call checks

Direct call checks apply to all direct calls including ICON and spooling.

1 The remote DNA of a direct call must be an assigned DNA.

Error: The remote DNA of a Direct Call is unassigned.

Location: <pm> PE/x PI/y PO/z <service> Direct Call Remote DNA <dna>.

Fix: Change the remote DNA to an assigned DNA.

2 The remote DNA of a direct call cannot be a PVC.

Error: The remote DNA of a Direct Call is a PVC.

Location: <pm> PE/x PI/y PO/z <service> Direct Call Remote DNA <dna>.

Fix: Change the remote DNA to a different DNA.

3 The remote DNA of a direct call cannot be of certain service types.

Error: The remote DNA of a Direct Call cannot be of service type <service>.

Location: <pm> <component>/x <component>/y Remote DNA <dna>.

Fix: Change the remote end of the Direct Call to a DNA that is NOT one of the following service types: <service>, <service>, ...

DNA uniqueness checks

1 A check is run on all DNAs defined in Preside Multiservice Data Manager (MDM) and the external DNA file. All duplicate DNAs are reported.

Error: Duplicate DNA <dna>.

Location: <pm> PE/x PI/y PO/z <service>.

Location: <pm> PE/x PI/y PO/z <service>.

2 When at least one duplicate DNA is found, a message is generated to warn the user about the other messages that are referencing duplicate DNAs.

Warning: Duplicate DNAs have been found. Subsequent messages referencing DNAs may be wrong or misleading if the DNA is a duplicate.

Fix: Fix the duplicate DNA problems and re-run NSIC.

Gateway ID uniqueness check

1 A check is run on all Gateway IDs. All duplicate Gateway IDs are reported.

Error: Duplicate Gateway ID<gid>.

Location: <pm> PE/x PI/y PO/n X25_Gateway X25Gateway_ID/<gid>.

Location: <pm> PE/x X25_Multilink_Gateway_Agent/ n X25GateWay_ID/<gid>.

Location: <pm> PE/x PI/y PO/n X75_Singlelink_Gateway Gateway_ID/ <gid>.

Location: <pm> PE/x X75_Multilink_Agent/n GateWay_ID/<gid>.

IP address uniqueness check

1 A check is run on all IP addresses. All duplicate IP addresses are reported.

Error: Duplicate IP address <ipaddr>.

Location: <em> IpInterfaceOverFrameRelay ipAddress: <ipaddr>.

Location: <em> IpInterfaceOverVc ipAddress: <ipaddr>.

Location: <em> VirtualRouter/<vr> ProtocolPort/<pp> IpPort IpLogicalInterface/<ipaddr>.

Hunt group checks

Hunt group primary/backup configuration and service type is determined by checking the port configuration of the first member of the hunt group. If the first member is configured incorrectly, a number of messages will be generated.

In the following messages, PE_Hunt_Group_server may be replaced by PE_386_Combination_Server, depending on the actual configuration.

1 The secondary hunt group (if configured) must be on a different PE.

Error: A secondary hunt group server is on the same PE as the primary
server.

Primary: <pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>.

Secondary: <pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>.

Fix: Configure one of the hunt group servers on a different PE.

2 The primary and secondary hunt groups must have the same number of members.

Error: Primary and secondary hunt groups have different number of members.

Primary: <pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>.

Secondary: <pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>.

Fix: Configure the primary and secondary to be identical.

3 The primary and secondary hunt groups must have the same members and be in the same order.

Error: Primary and secondary hunt groups have a different member.

Primary: <pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna> Member DNA n <dna>.

Secondary: <pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna> Member DNA n <dna>.

Fix: Configure the primary and secondary to be identical.

4 All hunt group member DNAs must exist.

Error: A hunt group member DNA is not assigned.

Hunt Group Member:
<pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna> Member DNA n <dna>.

Fix: Configure DNA, or remove member from hunt group.

5 The members of a hunt group must be of a valid service type. The valid service types for hunt group members are X25, SBSC, POS, ITI, SNA and Token Ring SNA PAD.

Error: Hunt group membership not supported for DNA.

Hunt Group Member:
<pm> PE_Hunt_Group_Server/x Hunt_Group_List_Server/y Hunt_Group_DNA/<dna> Member DNA n <dna>.

Port: <pm> PE/x PI/y PO/z <service> DNA/<dna>.

Fix: Remove member from the hunt group.

6 The members of a hunt group must be of compatible service types.

Note: All services are compatible only with themselves, except ITI and X25 which are compatible with each other.

Error: Members of a hunt group are not compatible.

Location: <pm> PE_Hunt_Group_Server/x
Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>
Member DNA n <dna>.

Hunt Group Service type:
<service>.

Member Service Type:

<service>.

Fix: Remove member from hunt group, or change its service type.

7 All members of hunt groups are provisioned at the port level.

Error: Hunt group member configured incorrectly at the port level.

Hunt Group Member:
<pm> PE_Hunt_Group_Server/x
Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>
Member DNA n <dna>.

Location: <pm> PE/x PI/y PO/z <service> DNA/<dna>.

Fix: Configure the port to be part of the hunt group.

8 For ports, the primary and secondary hunt group DNAs must represent the correct hunt group server DNAs.

Error: Port hunt group member DNA is not a hunt group server DNA.

Location: <pm> PE/x PI/y PO/z <service> DNA/<dna>
Hunt_DNA <Primary/Secondary> hunt group DNA <dna>.

Fix: Correct port hunt group provisioning.

Error: Port hunt group primary DNA is invalid.

Location: <pm> PE/x PI/y PO/z DNA/<dna> Hunt_DNA
Primary hunt group DNA <dna>.

Expected Server:
<pm> PE_Hunt_Group_Server/x
Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>.

Fix: Correct port hunt group provisioning.

9 Hunt group member must be provisioned with the secondary DNA of the hunt group.

Error: Port hunt group secondary DNA is invalid.

Location: <pm> PE/x PI/y PO/z <service> DNA/<dna>
Hunt_DNA Secondary hunt group DNA <dna>.

Expected Server:
<pm> PE_Hunt_Group_Server/x
Hunt_Group_List_Server/y Hunt_Group_DNA/<dna>.

Fix: Correct port hunt group provisioning.

10 Hunt group member must be provisioned with the secondary DNA of the hunt group.

Error: Port secondary hunt group is unknown.

Location: <pm> PE/x PI/y PO/z <service> DNA/<dna>
Hunt_DNA Secondary hunt group DNA <dna>.

Fix: Correct secondary hunt group.

11 All DNAs provisioned as hunt group members at the port level are in the primary hunt group.

Warning: Port provisioned as hunt group member is not
configured in that hunt group server.

Location: <pm> PE/x PI/y PO/z <service> DNA/<dna>
Hunt_DNA Primary hunt group DNA <dna>.

Fix: Add port to hunt group server, or remove hunt DNA provisioning.

MID uniqueness checks

1 A check is run on MIDs defined in the Module_Data component. All duplicate MIDs are reported. MIDs have to be unique within a routing zone.

Note that stub-AMs do not have a MID and are therefore not checked.

Error: Duplicate Module ID n in Routing Zone n.

Location: <pm> Module_Data Module ID: n.

Location: <pm> Module_Data Module ID: n.

2 For Passport modules, a check is run on Module ID (MID) defined in the Routing DpnAddressPlan component. All duplicate MIDs must be unique within a RoutingID (RID).

Error: Duplicate ModuleId n in RoutingId n.

Location: <em> Routing DpnAddressPlan moduleId: n.

Location: <em> Routing DpnAddressPlan moduleId: n.

Module mnemonic uniqueness check

1 A check is run on all module mnemonics defined in the DPN Module_Data component and on all node names defined in the Passport ModuleDate component. All duplicates are reported.

Error: Duplicate Module Mnemonic <pm>.

Location: <pm> Module_Data Module Mnemonic: <pm>.

Location: <pm> Module_Data Module Mnemonic: <pm>.

Location: <em> ModuleData nodeName: <em>.

NAMS ID uniqueness check

Location: <pm> Module_Data NAMS ID: n.

1 A check is run on NAMS IDs defined in the Module_Data component and in the Link component of the NCS coordinator application. All duplicate NAMS IDs are reported.

Error: Duplicate NAMS ID n.

Location: <pm> PE_NCS_Server/x OA/<oa> Link OA NAMS ID: n.

Fix: Refer to an assigned RID.

Node ID uniqueness check

1 A check is run on the node IDs defined in the Passport ModuleData component. All duplicate node IDs are reported.

Error: Duplicate Node ID <nodeId>.

Location: <em> ModuleData nodeId: <nodeId>.

Location: <em> ModuleData nodeId: <nodeId>.

OA uniqueness check

1 A check is run on the OA names and OA types defined in the OA Link components. Primary OA names must be unique and backup OA names must be unique. There can not be more than one backup to a primary OA. All duplicate OAs are reported.

Error: Duplicate OA <oa>.

Location: <pm> PE_NCS_Server/n OA/<oa> Link OA type: <oatype>.

Location: <pm> PE_NCS_Server/n OA/<oa> Link OA type: <oatype>.

PVC checks

1 The remote DNA of a PVC must be an assigned DNA.

Error: The remote DNA/LCN of a PVC is unassigned.

Location: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote_DNA <dna> Remote LCN <lcn>.

Fix: Change the remote DNA to an assigned DNA.

2 The remote end of a PVC is not configured as a PVC.

Error: PVC remote end is not configured as a PVC.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote_DNA <dna> Remote LCN <lcn>.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC.

Fix: Configure the remote DNA as a PVC.

3 The remote DNA of a PVC is incompatible.

Error: The service types of the DNAs of a PVC are different. Both ends of a PVC involving either Frame Relay or HTDS must have the same service type. (Remote DNAs may be external).

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote_DNA <dna>.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n.

Fix: Correct local/remote DNA.

Error: The service types of the DNAs of a PVC are incompatible. The IpInterfaceOverFrameRelay DNA is compatible with a FrameRelayUni DNA only. (Remote DNAs may be external.)

Local: <em> FrameRelayUni/<n> DataNetworkAddress/<dna> PVC/<n> Remote_DNA <dna>.

Remote: <em> IpInterfaceOverFrameRelay DataNetworkAddress/<dna>
PVC/<n>.

Fix: Correct local/remote DNA.

4 When a PVC is defined between 2 DNAs and a backup PVC is defined, call redirection must be defined between the remote DNA and the backup DNA.

Error: Incomplete backup PVC configuration.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote_DNA <dna> Remote LCN <lcn>.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote_DNA <dna> Remote LCN <lcn>.

Fix: Call redirection must be defined between DNA <dna> and DNA <dna> or the remote PVC should be deleted.

5 Both ends of the PVC must refer to each other.

Error: Both ends of the PVC should refer to each other.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote_DNA <dna> Remote LCN <lcn>.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Remote DNA <dna> Remote LCN <lcn>.

Fix: Configure remote DNAs to refer to each other.

6 One end of the PVC must be configured as master and the other end configured as the slave.

Error: Slave/Master configuration of ends of PVC is inconsistent.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Master end: <Local/Remote>.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Master end:<Local/Remote>.

Fix: Configure one end as master, and the other end as slave.

7 For a backup PVC, the “Slave Calls Master” field must be set to “No”.

Error: PVC “Slave Calls Master” must be set to “No” for a backup PVC.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Slave Calls Master: Yes.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n.

Fix: Configure the backup PVC with “Slave Calls Master” set to “No”.

8 Both ends of a PVC must have the same priority.

Error: PVC Priority does not match.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n
Priority: <Normal/High>.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n
Priority: <Normal/High>.

Fix: Correct priority.

9 PVC send window size should equal remote receive window size.

Error: PVC Send Window Size does not equal remote Receive Window
Size.

Local: <pm> PE/z PI/y PO/z <service> DNA/<dna> PVC/n
Send Window Size: m.

Remote: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n
Receive Window Size: m.

Fix: Correct window size.

10 PVC send throughput class should equal remote receive throughput class.

Error: PVC Send Throughput Class does not equal remote Receive
Throughput Class.

Local: <pm> PE/x PI/y PO/z <service> DNA/<dna> PVC/n Send
Throughput Class: m.

Remote: <pm> PE/z PI/y PO/z <service> DNA/<dna> PVC/n Receive
Throughput Class: m.

Fix: Correct throughput class.

11 PVC send packet size should equal remote receive packet size.

Error: PVC Send Packet Size does not equal remote Receive Packet Size.

Local: <pm> Frame Relay Uni/z DNA/<dna> PVC/n
Send Packet Size: m.

Remote: <pm> Frame Relay Uni/z DNA/<dna> PVC/n
Receive Packet Size: m.

Fix: Correct packet size.

RID map checks

1 The prime RID of this module should be ticked in Gateway Common Data.

Error: The prime RID of this module is not ticked in Gateway Common Data.

Location: <pm> Gateway_Call_Router Gateway_Common_Data RID n.

Fix: Tick the RID in the Gateway Common Data.

2 The prime RID of this module should be ticked in System Attributes.

Error: The Prime RID of this module is not ticked in System Attributes.

Location: <pm> RID_Routing System_Attributes RID n.

Fix: Tick the RID in System Attributes.

Voice networking consistency checks

1 The DNA attribute of the VoiceRoute DNA component cannot be empty when the VoiceRoute must support DNA-based call setup.

Warning: The DNA will have to be assigned in order for the VoiceRoute to
handle a DNA based call setup.

Location: <em> VoiceRoute/n DNA.

Fix: Assign a valid DNA.

2 Check the DNA attribute of the VoiceNetworkingCallServer DiallingPlan DirectoryNumber component to see if it refers to an existing VoiceRoute DNA (referred DNA).

a) When the referred DNA does not exist and the Destination Node ID (DNI) and Destination Component ID (DCI) are not empty, the VoiceRoute defined in VoiceNetworkingCallServer is used.

Warning: The referred DNA does not exist and will be ignored.
The VoiceRoute defined in
<em> VoiceNetworkingCallServer/a DiallingPlan/b
DirectoryNumber/c will be used.

Location: <em> VoiceNetworkingCallServer/a DiallingPlan/b
DirectoryNumber/c DNA/d.

Fix: Assign a valid DNA.

b) When the referred DNA does not exist and the Destination Node ID (DNI) and Destination Component ID (DCI) are empty, a valid DNA must be assigned.

Error: The referred DNA does not exist.

Location: <em> VoiceNetworkingCallServer/a DiallingPlan/b DirectoryNumber/c DNA/d.

Fix: Assign a valid DNA.

3 The DNA attribute of the VoiceNetworkingCallServer DiallingPlan DirectoryNumber component should reference VoiceRoute or external service type.

Error: DNA is of wrong service type. Valid types are VoiceRoute or external.

Location: <em> VoiceNetworkingCallServer/a DiallingPlan/b DirectoryNumber/c DNA/d.

Fix: Assign a valid DNA.

4 The Voice Route reference between the Destination Node ID, Destination Component ID and the DNA cannot be different.

Error: Inconsistent VoiceRoute reference found between DestinationNodeID/DestinationComponentID and the DNA.

Location: <em> VoiceNetworkingCallServer/a DiallingPlan/b DirectoryNumber/c DNA/d.

Location: <em> VoiceRoute/n DNA/m.

Fix: Fix the inconsistency.

5 The Destination Node ID (DNI) of the VoiceNetworkingCallServer DiallingPlan DirectoryNumber component must exist in the network.

Warning: The referred DestinationNodeID does not exist in the network.

Location: <em> VoiceNetworkingCallServer/a DiallingPlan/b DirectoryNumber/c DNI/d.

Fix: Assign a valid DNI.

6 The remote name of a Permanent Logical Connection must exist.

Error: The remote Permanent Logical Connection does not exist.

Local: <em> remoteName <remoteName>.

Fix: Assign a valid remote name.

7 The remote name of a Permanent Logical Connection cannot be empty.

Warning: The Permanent Logical Connection remoteName is empty.

Local: <em>.

Fix: Assign a valid remote name.

8 The service type in the local name and the remote name of a Permanent Logical Connection must match.

Error: The Permanent Logical Connection remoteName refers to a wrong service type.

Local: <em> remoteName <remoteName>.

Fix: Assign a valid remote name.

9 The local name of one end must match the remote name of the other end of a Permanent Logical Connection.

Error: Unmatched Permanent Logical Connection remoteName found.

Local: <em> remoteName <remoteName>.

Remote: <em> remoteName <remoteName>.

Fix: Assign a valid remote name.

10 The remote name of the remote end of a Permanent Logical Connection cannot be empty.

Warning: The Permanent Logical Connection remoteName on <localName> is empty. It should be <remoteName>.

Local: <em> remoteName <remoteName>.

Remote: <em>.

Fix: Assign a valid remote name.

Note: For the Permanent Logical Connection checks, the syntax for both <localName> and <remoteName> is
EM/<Node Name> <Service Type>/<id>
where <Service Type> is VS, HTDS, or BTDS.

VXID uniqueness check

1 A check is run on all non-zero Virtual XIDs (VXIDs) defined in the DNAs. All duplicate VXIDs are reported.

Error: Duplicate Virtual XID <vxid>.

Location: <pm> PE/x PI/y PO/z <service> Virtual XID: <vxid>.

Location: <pm> PE/x PI/y PO/z <service> Virtual XID: <vxid>.

Accounting parameters consistency checks

1 A check is run on the segment size attributes defined in the Network_Data and VirtualCircuitSystem components. The segment sizes should be consistent in the whole network. If an inconsistency is detected, all segment sizes are reported.

Error: Inconsistent accounting segment sizes.

Location: <em> ModuleData VirtualCircuitSystem segmentSize: <x>.

Location: <pm> Network_Data Account segment size: <x>.

2 A check is run on the generation mode attributes defined in the Network_Data and VirtualCircuitSystem components. The generation modes should be consistent in the whole network.

A value of *Both ends* for DPN Account generation mode is equivalent to *bothEnds* for Passport generationMode; and *Charged end* is equivalent to *singleEnd*.

If an inconsistency is detected, all generation mode attributes are reported.

Error: Inconsistent accounting generation modes.

Location: <pm> Network_Data Account generation mode: <mode>.

Location: <em> ModuleData VirtualCircuitSystem generationMode: <mode>.

3 A check is run on the accounting units counted attributes defined in the Network_Data and VirtualCircuitSystem components. The values of the attributes should be consistent in the whole network.

A value of *Yes* for DPN Frame count used attribute is equivalent to *frames* for Passport unitsCounted attribute; and *No* is equivalent to *segments*.

If an inconsistency is detected, all Frame count used and units counted attributes are reported.

Error: Inconsistent accounting units counted.

Location: <pm> Network_Data Frame count used: <units>.

Location: <em> ModuleData VirtualCircuitSystem unitsCounted: <units>.

4 A check is run on the max subnet packet size attributes defined in the VirtualCircuitSystem component of Passport and the Module_Data component of DPN. The max subnet packet sizes should be consistent in the whole network. If an inconsistency is detected, all max subnet packet sizes are reported.

Error: Inconsistent max subnet packet size.

Location: <em> ModuleData VirtualCircuitSystem maxSubnetPacketSize: <x>.

Location: <pm> Module_Data Maximum subnet packet size: <x>.

5 A check is run on the peak water mark interval attributes defined in the Passport Collector components. The list of defined intervals must be consistent in the network. If an inconsistency is detected, all peak water mark intervals are reported.

Error: Inconsistent accounting peak water mark intervals.

Location: <em> Collector/ACCOUNTING peak water mark interval: <x>.

Location: <em> Collector/ACCOUNTING peak water mark interval: <y>.

6 A check is run on the accounting collection times defined in the Passport Collector components. The list of defined collection times must be consistent in the network. If an inconsistency is detected, all collection times are reported.

Error: Inconsistent accounting collection times.

Location: <em> Collector/ACCOUNTING collection times: (<time1> <time2> ...).

Location: <em> Collector/ACCOUNTING collection times: (<time1> <time2> ...).

7 A check is run on the NCS Time of Day Accounting schedules defined in the DPN NCS_Time_Of_Day_Accounting components. Schedules with the same name must be consistent in the network. If an inconsistency is detected, all NCS Time of Day Accounting schedules are reported.

Note 1: If 24 X 1-hour TODA schedule is *Enable*, the following schedule is forced on the PM:

```
0000   0100   0200   0300   0400   0500   0600   0700
0800   0900   1000   1100   1200   1300   1400   1500
1600   1700   1800   1900   2000   2100   2200   2300
```

Note 2: If 24 X 1-hour TODA schedule is *Disable*, only the normal schedules are checked, the alternate schedules are NOT checked.

Error: Inconsistent time of day accounting schedules.

Location: <pm> NCS_Time_Of_Day_Accounting TODA_Schedule/<schedule>

Sunday <time1> <time2> <time3> <time4>
Monday <time1> <time2> <time3> <time4>
Tuesday <time1> <time2> <time3> <time4>
Wednesday <time1> <time2> <time3> <time4>
Thursday <time1> <time2> <time3> <time4>
Friday <time1> <time2> <time3> <time4>
Saturday <time1> <time2> <time3> <time4>

Location: <pm> NCS_Time_Of_Day_Accounting TODA_Schedule/<schedule>

Sunday <time1> <time2> <time3> <time4>
Monday <time1> <time2> <time3> <time4>
Tuesday <time1> <time2> <time3> <time4>
Wednesday <time1> <time2> <time3> <time4>
Thursday <time1> <time2> <time3> <time4>
Friday <time1> <time2> <time3> <time4>
Saturday <time1> <time2> <time3> <time4>

Location: <pm> NCS_Time_Of_Day_Accounting TODA_Schedule/<schedule>

Note: 24 X 1-hour TODA schedule is *Enable*. Therefore, the following schedule is forced:
0000 0100 0200 0300 0400 0500 0600 0700 0800 0900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100 2200 2300

8 A check is run on the TODA Schedule name attributes defined in the Time_Of_Day_Accounting component under the Control_Device_Manager of the OAs, and the Destination mnemonic (OA mnemonic) attributes of the control ICON_Mnemonic console managers.

The same TODA schedule name should be used by all control ICON_Mnemonic destination OAs.

If an inconsistency is detected, the control ICON_Mnemonic Destination mnemonics, and Control_Device_Manager Time_Of_Day_Accounting Schedule names are reported.

Warning: Inconsistent use of time of day accounting schedules.

Location: <pm> Console_Manager ICON_Mnemonic/<control icon> Destination mnemonic:<OA1>.

Location: <pm> Console_Manager ICON_Mnemonic/<control icon> Destination mnemonic:<OA2>.

Location: <pm> PE_NCS_Server/<x> PE_Server NCS_Server OA/<OA1> Control_Device_Manager/<device>Time_Of_Day_Accounting Schedule name: <schedule1>.

Location: <pm> PE_NCS_Server/<x> PE_Server NCS_Server OA/<OA1> Control_Device_Manager/<device> Time_Of_Day_Accounting Schedule name: <schedule1>.

Location: <pm> PE_NCS_Server/<x> PE_Server NCS_Server OA/<OA2> Control_Device_Manager/<device> Time_Of_Day_Accounting Schedule name: <schedule2>.

Location: <pm> PE_NCS_Server/<x> PE_Server NCS_Server OA/<OA2> Control_Device_Manager/<device> Time_Of_Day_Accounting Schedule name: <schedule2>.

9 A check is run on all modules to see if they all have Time of Day Accounting (TODA) change-over times provisioned. If some modules have TODA while others do not, then a warning is produced, and the modules without TODA are reported. Note that there is no warning if all modules do not have TODA.

Warning: Only some modules have Time of Day Accounting (TODA) change-over times provisioned. The following modules do not have TODAs provisioned:

Location: <module1>.

Location: <module2>.

10 If the Passport collection times are consistent, and the DPN NCS Time of Day Accounting schedules are consistent, a check is run on the NCS Time of Day Accounting. The TODA changeover times must be same for every day of the week in a network containing Passports and DPNs.

If an inconsistency is detected, the Time of Day Accounting schedule is reported.

Error: Inconsistent Time of Day Accounting schedule during week.TODA changeover times must be same for every day of the week in a Passport-DPN network.

Location: <pm> NCS_Time_Of_Day_Accounting TODA_Schedule/<schedule>

Sunday <time1> <time2> <time3> <time4>
Monday <time1> <time2> <time3> <time4>
Tuesday <time1> <time2> <time3> <time4>
Wednesday <time1> <time2> <time3> <time4>
Thursday <time1> <time2> <time3> <time4>

Friday <time1> <time2> <time3> <time4>
Saturday <time1> <time2> <time3> <time4>

11 If the Passport collection times are consistent, and the DPN NCS Time of Day Accounting schedules are consistent, a check is run between the Passport collection times and the DPN NCS Time of Day Accounting schedules.

If an inconsistency is detected, the Passport collection times and DPN NCS Time of Day Accounting schedules are reported.

Error: Inconsistent Passport collection times and DPN Time of Day Accounting (TODA) schedules:

Passport collection times: (<time1> <time2> ...)
DPN TODA schedule <schedule>: (<time1> <time2> ...)

Location: <em> Collector/ACCOUNTING collection times: (<time1> <time2> ...).

Location: <em> Collector/ACCOUNTING collection times: (<time1> <time2> ...).

Location: <pm> NCS_Time_Of_Day_Accounting TODA_Schedule/<schedule>

Sunday <time1> <time2> <time3> <time4>
Monday <time1> <time2> <time3> <time4>
Tuesday <time1> <time2> <time3> <time4>
Wednesday <time1> <time2> <time3> <time4>
Thursday <time1> <time2> <time3> <time4>
Friday <time1> <time2> <time3> <time4>
Saturday <time1> <time2> <time3> <time4>

OR

Location: <pm> NCS_Time_Of_Day_Accounting TODA_Schedule/<schedule>

Note: 24 X 1-hour TODA schedule is *Enable*. Therefore, the following schedule is forced:
0000 0100 0200 0300 0400 0500 0600 0700 0800 0900 1000
1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100
2200 2300

Location: <pm> Console_Manager ICON_Mnemonic/<control icon> Destination mnemonic:<OA>.

Location: <pm> PE_NCS_Server/<x> PE_Server NCS_Server OA/<OA> Control_Device_Manager/<device> Time_Of_Day_Accounting Schedule name: <schedule>.

ATM PVC checks

NSIC provides ATM checks. It performs semantic checks to detect inconsistencies or problems in the service data for ATM permanent virtual circuits (PVC) across Passport nodes.

To follow the PVC, NSIC uses the information provided in the remoteAtmInterfaceLabel field to locate the remote end of each hop of the PVC.

> ***Note:*** To perform ATM PVC checks, the remoteAtmInterfaceLabel (located under the AtmIf component) must be provisioned. If this field is blank, NSIC assumes that the remote end is properly provisioned or that it is part of an external network. Consequently, no checks are performed for that remote end.

The format of the remoteAtmInterfaceLabel is

```
EM/<PassportName> ATMIF/<n> or
EM <PassportName> ATMIF<n>
```

where:

<PassportName>

is the name of the Passport node on which the remote end of the Atmif has been provisioned.

<n>

is the instance of the remote AtmInterface component.

1 The virtual path identifiers (VPI), virtual path identifier/virtual channel identifier (VPI/VCI) values between the two ends of the PVC hop should match. If the local end is a virtual path terminator (VPT), the remote end should be a virtual path connection (VPC) with the same VPI as that for the VPT. If the local end is a VPC, the remote end should be either a VPT, VPC, or virtual channel connection (VCC) with the same VPI. If the local end is a VCC, then the remote end should either be a VCC with the same VPI/VCI, or a VPC with the same VPI value.

Error: A compatible service with VPI.VCI <x.y> does not exist at the remote end.

Local: <em> ATMIF/<a> <service>/<x.y>

Remote: <em> ATMIF/<b>

Fix: Add a compatible service with the matching VPI.VCI number at the remote end.

2 The interfaces should point to each other.

Error: The remote AtmInterface “<em> ATMIF/<a>” does not point to “<em> ATMIF/<b>”.

Local: <em> ATMIF/<b> remoteAtmInterfaceLabel “em/<em> atmif/<a>”

Remote: <em> ATMIF/<a> remoteAtmInterfaceLabel “em/<em> atmif/<c>”

Fix: Either point the remote AtmInterface to the local one or point the local AtmInterface to the valid remote AtmInterface.

3 The remote AtmInterface specified in the RemoteAtmInterfaceLabel field should exist.

Error: The referred remote AtmInterface “em/<em> atmif/<a>” does not exist.

Location: <em> ATMIF/2 remoteAtmInterfaceLabel “em/<em> atmif/<a>”

Fix: Add the remote AtmInterface.

4 Only one interface can point to another interface.

Error: Another AtmInterface is already pointing to the remote AtmInterface “<em> ATMIF/<a>”.

Local: <em> ATMIF/<b> remoteAtmInterfaceLabel “em/<em> atmif/<a>”

Remote: <em> ATMIF/<a> remoteAtmInterfaceLabel “em/<em> atmif/<c>”

Fix: Change the remote AtmInterface in <em> ATMIF/20.

5 The format of the remoteAtmInterfaceLabel should be valid.

Error: The format of the remoteAtmInterfaceLabel attribute “EM/<em>” is invalid.

Location: <em> AtmInterface/<x> remoteAtmInterfaceLabel “EM/<em>”.

Fix: Correct the format. It should be
“EM/<PassportName> ATMIF/<n>” or
“EM <PassportName> ATMIF <n>”

where:
service can be VirtualChannelConnection, VirtualPathConnection or VirtualPathTerminator.

Chapter 9
Record definition and module data file formats

This chapter describes the RDF file syntax and the NRS data format for service data. In this chapter, you can find the following information:

About record definition files

Record definition files (RDF) define the record layout for components. RDFs are used by both the DPN and Passport NRS Populators to determine record formats when creating the module data files. RDFs are also used by the NRS reporting tools to allow Network Reporting System report programmers to reference service data fields by name rather than by offset in the component record.

RDFs can be found in the /opt/MagellanNMS/data/nrs/rdf directory. The dpn subdirectory contains DPN RDFs. The ppc subdirectory contains Passport carrier RDFs. The ppe subdirectory contains Passport enterprise RDFs.

The file naming convention for RDFs is *<component>.rdf*.

For DPN, *<component>* is referred to as a category name and is unique across all versions.

For Passport (ppc or ppe), *<component>* is referred to as a component id. Component ids will change from one version to the next. In order to have one set of Passport RDFs which can be used for multiple versions, the component or attribute ID is used to identify the RDF. There are a number of structured attributes that are used in Passport provisioning. These attributes may have more than one value and be identified by one or two indices. In the normal RDF representation, the attribute is represented by a field in the RDF and this representation cannot be used to describe the structure attribute. To represent the structured attribute, two RDFs are required, the first RDF, *<id>.rdf*, is used to describe the attribute itself, owner list, hierarchy level, etc... . The second RDF, *<id>_x.rdf*, is used to describe the actual value of the field and its associated index. For example, the windowSize (id = 245) is an array and it is described by the 245.rdf and 245_x.rdf. The owner list and hierarchy level are specified in the 245.rdf but the value of the field and its two indices (packetSize and throughputClass) are actually specified in the 245_x.rdf.

DPN RDFs are provided as part of the Preside Multiservice Data Manager (MDM) software and are installed in the /opt/MagellanNMS/data/nrs/rdf/dpn subdirectory.

Passport RDFs are generated based on the model description file that is uploaded from the Passport node. The RDFs are then stored in the /opt/MagellanNMS/cfg/PassportSchema/RDF_<version> directory. They are then installed in either the ppe or ppc subdirectories in /opt/MagellanNMS/data/nrs/rdf/. The type (ppc or ppe) is determined based on the version of software running on the Passport.

See “RDF backward compatibility” (page 148) for more information on RDFs.

The NRS application developer needs to be familiar with the RDF syntax to be able to retrieve fields or to create new component types. When developing a report it is often necessary to restructure the service data. This is accomplished by developing a data extractor which pulls certain data from the module data files and restructures it into a new format.

Merging Passport RDFs

For each version of Passport software, there is an equivalent version of the model description file. The model description file is stored on the Passport disk.

After successfully connecting to a Passport, if the version of the model description file does not exist on Preside Multiservice Data Manager (MDM), the Passport provisioning stack (fps) uploads and stores the model description file on MDM. After uploading the file, fps generates the RDFs, activation files and forms, and loads the provisioning activation files in memory.

Within a Passport family (ppc or ppe), the new RDFs are then merged with the existing RDFs by the Populator in order to create a superset of RDFs which contains all provisionable components and their attributes at different versions. This allows the new RDFs to support multiple Passport versions without re-populating the NRS database.

If the model description file cannot be uploaded, or the freshly created activation files cannot be loaded into shared memory, the provisioning stack will try to use the files that are already activated in shared memory. If there is no provisioning activation file activated, the Passport NRS Populator terminates. If a provisioning activation file already exists in shared memory, a warning message is generated by the Passport NRS Populator indicating that the Passport data version and model version do not match. In this case, if the existing RDFs already support the version that is used by the provisioning stack, merging of the RDFs will not be performed.

The Passport model version is recorded in the data file using the MODEL_VERSION field described in RDF *<type>/2.rdf, where <type> is ppc or ppe*.

RDF syntax and file naming conventions

Every component RDF consists of a description of the fields that belong to that particular component. The position of the field defined in the RDF corresponds to the position of that field's data value in the module data file. If the third field of an RDF is DNIC, then the DNIC value will be the third field in the module data file. Since the NRS tools do not require the application developer to use field positions in their code, the fields are not explicitly numbered in the RDF.

At the beginning of all RDFs, comment lines are provided:

```
# RDF_VERSIONS: <list of RDF versions>
# COMPONENT: <component name>
# UI_PROMPT: <component UI long name>
# UI_ABBREVIATION: <component UI short name>
# OWNERS: <list of possible owner components>
# SUBORDINATES: <list of possible subordinate components>
```

The RDF_VERSIONS comment can only be found in the PM.rdf for DPN and the *2.rdf* for Passport. It indicates the valid RDF version supported by these RDFs. This comment is used by the NRS Populator and NRS Sanity Checker tools.

The component, UI prompt, and UI abbreviation comments identify the name of the component. The component name, also known as the *category name* (DPN) and *component id* (Passport), represents the unique name for the component. The UI prompt is the name of the component as seen in the Component Provisioning tool. The UI abbreviation is the short name of the component. Components without a short name have the UI prompt and UI abbreviation set the same.

The owner and subordinate lists are provided to ensure that report developers do not forget any component associations. The RDFs do not include whether the component associations are mandatory or optional. For example, the X25 and X25LINK component association is mandatory. When an X25 component is present, the X25LINK component will also be present as one of the components following the X25 component.

Following the comment lines are key and data fields. Key fields are used to uniquely identify the component. Key field names always start with an underscore character, for example, _PE. Every NRS component has two common keys, plus its own key field and any keys it might have inherited from its ancestors. The common key fields are _COMPONENT and _HIERARCHY_LEVEL. For components that do not really have a key (NETWORK_ENV, LOADER_ENV, ITI, ITILINK, etc) a key of type NOKEY is provided anyway. The value is $ if the component is present, and it is empty if the component is not present. This is particularly useful for components that can be under more than one hierarchy. For example, when looking at an ITILINK record, it is possible to determine if that record is part of an ITI or POS service by looking at keys _ITI and _POS. One of them will be set to $ while the other will be emply.

Each field in an RDF has a set of attributes associated with it. The table “Field attributes” (page 133) provides a list of these attributes.

Table 4
Field attributes

Attribute name	Meaning
_name	Service data field name.
_type	Field data type. One of: - NUMERIC - LISTINDEX - BOOLEAN - DNA - HEXADECIMAL - BIT_STRING - STRING - INVISIBLE - NOKEY
_width	Number of characters needed to represent, in ASCII, the largest value for that field.
_title	Field title string, from the Component Provisioning prompt for that field.
(Sheet 1 of 2)	

Table 4 (continued)
Field attributes

Attribute name	Meaning
_abbreviation	Short form of the field title string. Fields without a short name have the _title and _abbreviation set the same.
_group	Group name, taken from the Component Provisioning groupings of logically related service data. Some fields are not part of a group and the group name will be blank.
(Sheet 2 of 2)	

Example

The following example is the numbering plan indicator (164) field definition from the data network address (161) component of Passport:

```
_name: 164
        _type: STRING
        _width: 4
        _title: numberingPlanIndicator
        _abbreviation: npi
        _group: Address
```

The data fields (non_key) start with an alphanumeric character, for example, 164 or PRIMARYHGDNA. Some components may not have any data fields. Components with only key fields generally serve as place holders and navigation components. For example, the X25 component, which has no data fields, serves as the top of the X25 service hierarchy and ties the next level of X25 service components together.

Example

The following is an example of the RDF for the SNA Hunt DNA component, which has three possible owners, SNAXPAD, SNAMULTIHOST, and SNAT21, and no subordinate components. The key fields are:

_COMPONENT, _HIERARCHY_LEVEL, _PM, _PE, _PI, _PO, _SNA, _SNAPU, _SNAT21, _SNAMULTIHOST, _SNAXPAD and _SNAHUNTDNA.

```
# COMPONENT: SNAHUNTDNA
# UI_PROMPT: Hunt_DNA
# UI_ABBREVIATION: Hunt_DNA
# OWNERS: SNAXPAD SNAMULTIHOST SNAT21
# SUBORDINATES:
_name: _COMPONENT
  _type: STRING
  _width: *
  _title: Component
  _abbreviation: Component
  _group:
_name: _HIERARCHY_LEVEL
  _type: NUMERIC
  _width: 2
  _title: Hierarchy_level
  _abbreviation: Hierarchy_level
  _group:
_name: _PM
  _type: STRING
  _width: 12
  _title: PM
  _abbreviation: PM
  _group:
_name: _PE
  _type: NUMERIC
  _width: 2
  _title: PE
  _abbreviation: PE
  _group:
_name: _PI
  _type: NUMERIC
  _width: 2
  _title: PI
  _abbreviation: PI
  _group:
_name: _PO
  _type: NUMERIC
  _width: 2
  _title: PO
```

```
    _abbreviation: PO
    _group:
  _name: _SNA
    _type: NOKEY
    _width: 1
    _title: SNA
    _abbreviation: SNA
    _group:
  _name: _SNAPU
    _type: HEXADECIMAL
    _width: 3
    _title: PU
    _abbreviation: PU
    _group:
  _name: _SNAT21
    _type: NOKEY
    _width: 1
    _title: Type_2.1_Router
    _abbreviation: Type_2.1_Router
    _group:
  _name: _SNAMULTIHOST
    _type: NOKEY
    _width: 1
    _title: Multi-Host
    _abbreviation: Multi-Host
    _group:
  _name: _SNAXPAD
    _type: NOKEY
    _width: 1
    _title: XPAD
    _abbreviation: XPAD
    _group:
  _name: _SNAHUNTDNA
    _type: DNA
    _width: 16
    _title: Hunt_DNA
    _abbreviation: Hunt_DNA
    _group:
  _name: OAM
    _type: STRING
    _width: 10
    _title: Ownership
    _abbreviation: Ownership
    _group:
  _name: PRIMARYHGDNA
    _type: DNA
    _width: 16
    _title: Primary hunt group DNA
    _abbreviation: Primary hunt group DNA
    _group: Hunt DNAs
  _name: SECONDARYHGDNA
    _type: DNA
    _width: 16
    _title: Backup hunt group DNA
    _abbreviation: Backup hunt group DNA
    _group: Hunt DNAs
```

Example

The following is an example of a Passport RDF for components 320, 323 and 323_x . The 320 component has a subordinate called 323 and the 323 component has a subordinate called 323_x.

```
# COMPONENT: 320
# UI_PROMPT: Software
# UI_ABBREVIATION: Sw
# OWNERS: 2
# SUBORDINATES: 323 336
_name: _COMPONENT
  _type: STRING
  _width: *
  _title: Component
  _abbreviation: Component
  _group:
_name: _HIERARCHY_LEVEL
  _type: NUMERIC
  _width: 2
  _title: Hierarchy_level
  _abbreviation: Hierarchy_level
  _group:
_name: _2
  _type: STRING
  _width: 12
  _title: EM
  _abbreviation: EM
  _group:
_name: _320
```

```
  _type: NOKEY
  _width: 1
  _title: Software
  _abbreviation: Sw
  _group:
_name: OAM
  _type: STRING
  _width: 10
  _title: Ownership
  _abbreviation: Ownership
  _group:


# COMPONENT: 323
# UI_PROMPT: avList
# UI_ABBREVIATION: avl
# OWNERS: 320
# SUBORDINATES: 323_X
_name: _COMPONENT
  _type: STRING
  _width: *
  _title: Component
  _abbreviation: Component
  _group:
_name: _HIERARCHY_LEVEL
  _type: NUMERIC
  _width: 2
  _title: Hierarchy_level
  _abbreviation: Hierarchy_level
  _group:
_name: _2
  _type: STRING
  _width: 12
  _title: EM
  _abbreviation: EM
  _group:
_name: _320
  _type: NOKEY
  _width: 1
  _title: Software
  _abbreviation: Sw
  _group:
_name: _323
```

```
  _type: NOKEY
  _width: 1
  _title: avList
  _abbreviation: avl
  _group:
_name: OAM
  _type: STRING
  _width: 10
  _title: Ownership
  _abbreviation: Ownership
  _group:


# COMPONENT: 323_X
# UI_PROMPT: avList
# UI_ABBREVIATION: avl
# OWNERS: 323
# SUBORDINATES:
_name: _COMPONENT
  _type: STRING
  _width: *
  _title: Component
  _abbreviation: Component
  _group:
_name: _HIERARCHY_LEVEL
  _type: NUMERIC
  _width: 2
  _title: Hierarchy_level
  _abbreviation: Hierarchy_level
  _group:
_name: _2
  _type: STRING
  _width: 12
  _title: EM
  _abbreviation: EM
  _group:
_name: _320
  _type: NOKEY
  _width: 1
  _title: Software
  _abbreviation: Sw
  _group:
_name: _323
```

```
      _type: NOKEY
      _width: 1
      _title: avList
      _abbreviation: avl
      _group:
    _name: _323_X
      _type: LISTINDEX
      _width: 5
      _title: avList
      _abbreviation: avl
      _group:
    _name: OAM
      _type: STRING
      _width: 10
      _title: Ownership
      _abbreviation: Ownership
      _group:
    _name: 0
      _type: STRING
      _width: 30
      _title: value
      _abbreviation: value
      _group:
```

Component and field names

NRS names used to represent components and fields are not the same as the names seen in the Component Provisioning tool. The NRS names are known as component names while the Component Provisioning names are known as UI names. NRS uses the *component name* since it is unique while the UI name is not. The mapping between component name and UI name is provided in the RDF for both components and fields.

For components, the component name is provided by the following comment in the RDF:

# COMPONENT: <component name>

It is also the name of the RDF:

<component name>.rdf

The UI names (long and short) are provided by the following comments in the RDF:

# UI_PROMPT: <UI long name>
# UI_ABBREVIATION: <UI short name>

For fields, the component name is provided in the RDF by the following field attribute:

_name: <component name>

The UI names are provided in the RDF by the following field attributes:

_title: <UI long name>
_abbreviation: <UI short name>

UNIX tools can be used to get the category name to UI name mapping.

Example

You want to find the field Mnemonic that is part of the Module_Datacomponent.

You can search for the string # UI_PROMPT: Module_Data in all RDFs using the UNIX grep tool. This tool indicates that this string is found in the RDF MODULE_ENV.rdf file

Using the vi editor, look at the MODULE_ENV.rdf file and search for the string _title: Mnemonic. Three lines above, the field category name, MODULEMNEMONIC is found.

This field is part of component MODULE_ENV and is called MODULEMNEMONIC.

The NRS RDF Hierarchy Generator tool can be used to determine the relationship between the various NRS components as well as the mapping between category name and UI long and short names for all components and fields. See “NRS RDF Hierarchy Generator (nrsrdfh)” (page 272) for a description of this tool.

Module data files

A module data file is produced by the populator when it uploads a module. One module data file is created for every module that is uploaded. Each file represents a complete bundle even if the bundle was created from an incremental download. All the module data files together form an NRS database. Service data, which is input for reports and other tools, comes from the NRS database.

Module data file naming convention

Module data files have a standard naming convention. This format ensures that file names are unique for different modules or different bundles of a given module. It also provides an easy way to use pattern matching on the different parts of the file name. The format is as follows:

```
<type>.<pm>.<namsid>.<bundle/view
filename>.<date>.data
```

where:

type

To identify the type of module:
    dpn -->DPN module
    ppc -->Passport carrier module
    ppe -->Passport enterprise module

`pm`

The DPN or Passport module name.

`namsid`

The NAMS ID of the module obtained from the MCF name for DPN and module data component for Passport.

`bundle/view filename`

The name of the bundle or view file uploaded. The *bundle* is displayed in upper case and the *view filename* is displayed in mixed case.

`date`

For DPN, this field is the activation date of the MCF created by Component Provisioning using *dated* download mode. This field is a valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

It is recommended that *dated* download mode be used when using Component Provisioning. If the mode used is keyed or user specified, the date is carried forward from the existing MCF. To override this date, the *Change Activation Date* parameter must be specified in the Download Preferences when using Component Provisioning.

Example

The following are examples of DPN module data file names:

```
dpn.R34.4034.99031712.990317.data
dpn.R34.4034.MYBUNDLE.990317.data
dpn.ACTN30.2030.WK2101.990403.data
dpn.A20.2020.156.990317.data
```

The following are examples of Passport module data file names:

```
ppc.PASSPORT1.2105.MyViewFile.991216.data
ppe.PASSPORT5.2631.WK2101.991216.data
```

Module data file format

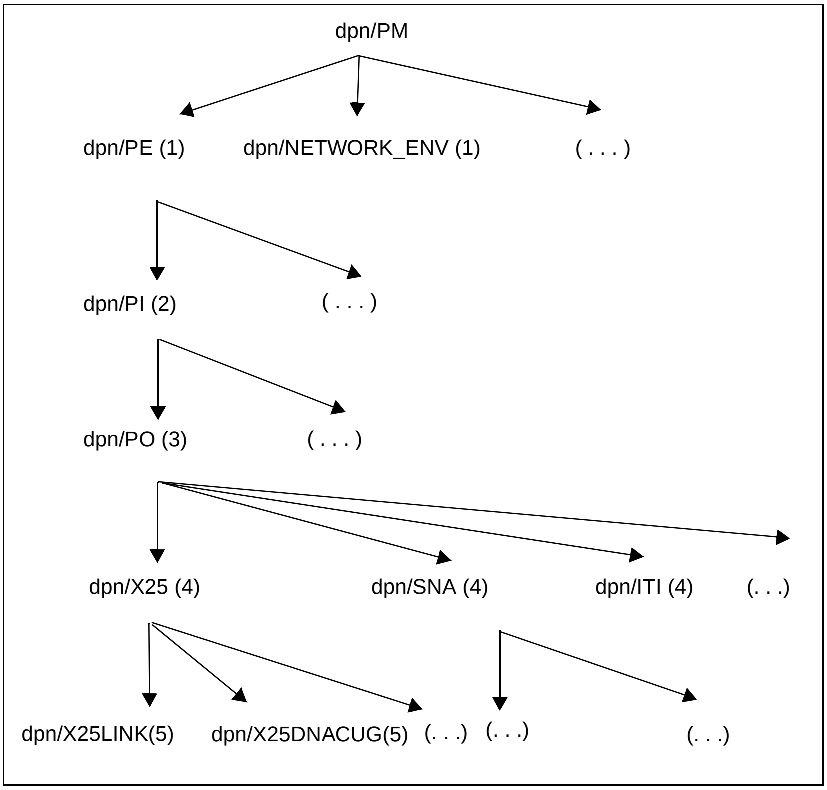
The internal format of these NRS module data files is based on the data model. One module data file represents all visible service data that can be provisioned on a particular module.

The organization of the components in a module data file is based on the internal service data hierarchy. The position of a record within the file shows its relationships to other components. For example, for DPN the dpn/PM component is the top of the hierarchy and will always be the first record in the file and the only occurrence of the dpn/PM record in that file. A dpn/PM component has several subordinate components, one of them is the dpn/PE. The dpn/PE component is owned by the dpn/PM component. Similarly, a dpn/PI is owned by a dpn/PE component. The owner of a particular dpn/PI is always the last dpn/PE record which precedes that dpn/PI.

For each component, the second field represents the level of hierarchy. It can be used to determine the hierarchy.

An example of the DPN service data hierarchy is shown in the figure “DPN service data hierarchy” (page 144). The dpn/PM record has a hierarchy level of 0, its children (for example, dpn/PE(s), NETWORK_ENV) have a hierarchy level of 1, their children have a hierarchy level of 2, and so on. The hierarchy level is shown beside the component in parenthesis.

Figure 5
DPN service data hierarchy



Each record in a module data file represents a service data component and has a corresponding RDF which describes the record layout. Each module data file is in ASCII format. Attributes of a component, such as line speed, are represented as fields within the record. Field names are not stored in the module data file, only data values; field names are stored in RDFs. Fields are delimited by the delete character (0x7F). In the following data file format examples the delete character is represented by the character sequence <7F> and records have been truncated. If a module data file is viewed using an editor, the delete character is different. The first field in each record is the component name and the second field is the hierarchy level.

Example

The following is an example of a DPN module data file.

```
dpn/PM<7F>0<7F>R34<7F>4034<7F>98111900<7F>981119<7F>NMS110Dai<7F>...
dpn/NETWORK_ENV<7F>1<7F>R34<7F>$<7F>OWNER_IWS<7F>BOCNET Network<7F>...
dpn/LARGE_VC_WINDOW_ENV<7F>1<7F>R34<7F>$<7F>OWNER_IWS<7F>3<7F>4<7F>...
dpn/ACCOUNTING_ENV<7F>1<7F>R34<7F>$<7F>OWNER_IWS<7F>1<7F>1<7F>1<7F>...
dpn/SWITCH_DATA<7F>1<7F>R34<7F>$<7F>OWNER_IWS<7F>34<7F>No<7F>DPN100
dpn/MODULE_ENV<7F>1<7F>R34<7F>$<7F>OWNER_IWS<7F>4034<7F>R34<7F>...
dpn/PE<7F>1<7F>R34<7F>8<7F>OWNER_IWS
dpn/LOADER_ENV<7F>2<7F>R34<7F><7F><7F><7F><7F><7F>8<7F>$<7F>...
dpn/PI<7F>2<7F>R34<7F>8<7F>8<7F>OWNER_IWS
dpn/HARDWARE_ENV<7F>3<7F>R34<7F>8<7F>8<7F>$<7F>OWNER_IWS<7F>...
dpn/PE_MNEMONIC_ENV<7F>2<7F>R34<7F><7F><7F><7F><7F><7F>8<7F>$<7F>...
dpn/PAM_SDA<7F>2<7F>R34<7F>8<7F>$<7F>OWNER_IWS
dpn/PE<7F>1<7F>R34<7F>9<7F>OWNER_IWS
dpn/LOADER_ENV<7F>2<7F>R34<7F><7F><7F><7F><7F><7F>9<7F>$<7F>...
dpn/PI<7F>2<7F>R34<7F>9<7F>9<7F>OWNER_IWS
(...)
```

Example

The following is an example of a Passport (ppc) module data file.

```
ppc/2<7F>0<7F>NODER16<7F>2105<7F>99051000,full,141<7F>990510<7F>...
ppc/117<7F>1<7F>NODER16<7F>ACCOUNTING<7F>OWNER_IWS<7F>
ppc/120<7F>2<7F>NODER16<7F>ACCOUNTING<7F>$<7F>OWNER_IWS
ppc/2384<7F>2<7F>NODER16<7F>ACCOUNTING<7F>$<7F>OWNER_IWS
ppc/123<7F>2<7F>NODER16<7F>ACCOUNTING<7F>$<7F>OWNER_IWS<7F>on<7F>200
ppc/117<7F>1<7F>NODER16<7F>ALARM<7F>OWNER_IWS<7F>
ppc/120<7F>2<7F>NODER16<7F>ALARM<7F>$<7F>OWNER_IWS
ppc/2384<7F>2<7F>NODER16<7F>ALARM<7F>$<7F>OWNER_IWS
ppc/123<7F>2<7F>NODER16<7F>ALARM<7F>$<7F>OWNER_IWS<7F>off<7F>30
ppc/117<7F>1<7F>NODER16<7F>LOG<7F>OWNER_IWS<7F>
ppc/120<7F>2<7F>NODER16<7F>LOG<7F>$<7F>OWNER_IWS
```

```
ppc/2384<7F>2<7F>NODER16<7F>LOG<7F>$<7F>OWNER_IWS
ppc/123<7F>2<7F>NODER16<7F>LOG<7F>$<7F>OWNER_IWS<7F>off<7F>10
ppc/117<7F>1<7F>NODER16<7F>DEBUG<7F>OWNER_IWS<7F>
ppc/120<7F>2<7F>NODER16<7F>DEBUG<7F>$<7F>OWNER_IWS
ppc/2384<7F>2<7F>NODER16<7F>DEBUG<7F>$<7F>OWNER_IWS
ppc/123<7F>2<7F>NODER16<7F>DEBUG<7F>$<7F>OWNER_IWS<7F>off<7F>2
ppc/117<7F>1<7F>NODER16<7F>SCN<7F>OWNER_IWS<7F>
ppc/120<7F>2<7F>NODER16<7F>SCN<7F>$<7F>OWNER_IWS
ppc/2384<7F>2<7F>NODER16<7F>SCN<7F>$<7F>OWNER_IWS
ppc/123<7F>2<7F>NODER16<7F>SCN<7F>$<7F>OWNER_IWS<7F>off<7F>10
(...)
```

RDF to module data file mappings

For each unique component type in a module data file, there exists a record definition file (RDF) which describes the structure of the component. This is shown in the figure “RDF to module data file mapping” (page 146) which illustrates the association between the record or component and its RDF. The sample module data file contains a single dpn/PM record, a single dpn/NETWORK_ENV record and two dpn/PE records.

Figure 6
RDF to module data file mapping

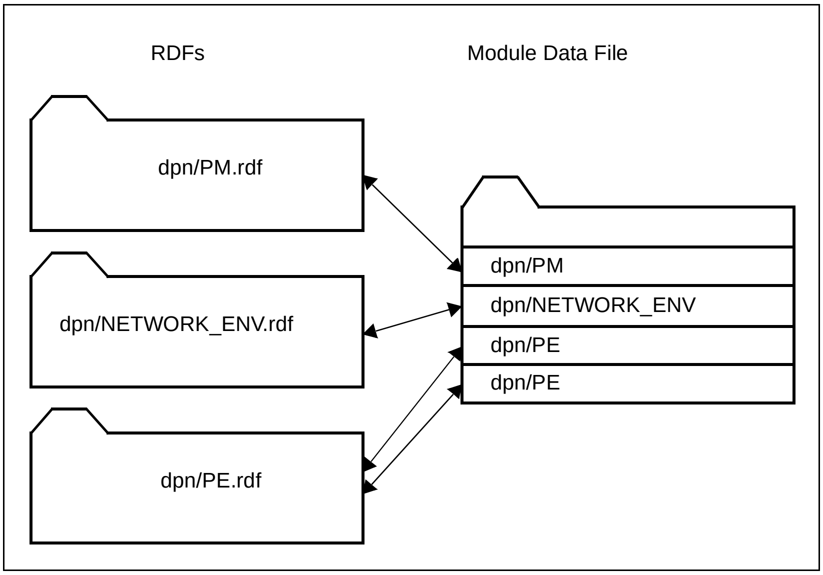
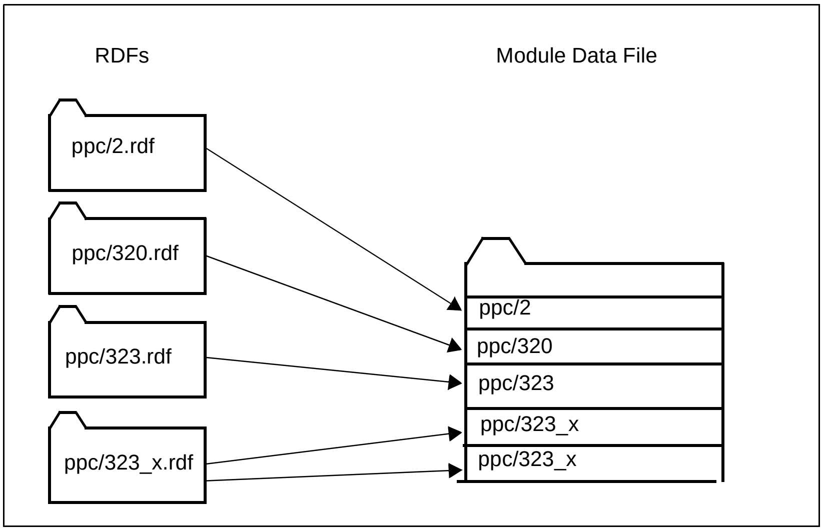


Figure 7
Passport (ppc) module data file mapping



Creating new RDFs

New components can be created through the creation of a new RDF. The most common way to create a new RDF is to extract service data fields from two or more existing components and combine them into a new component.

The NRS RDF naming conventions should be followed to name the new RDFs to ensure that the NRS toolset will be able to handle the new RDF. The component name must start with an underscore. For example, _MYCOMP.rdf.

When a new component is created its keys should be identified and included in the RDF. It is recommended that the key field naming convention which uses underscores as the first character of a key field be used for consistency when creating new RDFs. These RDFs must be stored in the custom subdirectory under the NRS_RDF_DIR directory.

> ***Note:*** It is recommended that the DPN and Passport NRS RDFs not be modified. Any changes made to the existing NRS RDFs would have to be propagated each time a new release of Preside Multiservice Data Manager (MDM) software was installed.

Extended component names

RDFs cannot have more than 199 fields because the nawk interpreter cannot handle more than 199 data fields in an input line. If a component has more than 199 fields, it is broken into two or more separate components each with its own RDF. The DPN ROUTING_DATA is an example of an extended component. In the routing data the first 199 fields make up the component called ROUTING_DATA and the remaining data fields make up a component called ROUTING_DATA_2. The naming convention for extended components is to append _2 to the name of the original component for the second 199 fields, and _3 for the next 199 data fields, and so on.

RDF backward compatibility

When a new release is installed, new DPN RDFs are provided in the */opt/MagellanNMS/lib/nrs/rdf/dpn* subdirectory. As part of the installation, these RDFs are either copied to the */opt/MagellanNMS/data/nrs/rdf/dpn* subdirectory (first time installation) or are merged with RDFs already present in that directory to create a superset. This is done automatically.

Passport RDFs are generated based on the model description file that is uploaded from the Passport node and stored in directory /opt/MagellanNMS/cfg/PassportSchema/RDF_<version>. These RDFs are then merged with the existing RDFs for the corresponding Passport family (ppc or ppe) to create a superset. This is triggered automatically by the pnrspop tool. This superset can then be used to support multiple Passport versions.

This new set of RDFs is backwards compatible with the existing set. This means that the new fields are added to a component at the end of the RDF for that component and that deleted fields are kept.

This allows for an easy NRS upgrade. The NRS database does not need any conversion and does not need to be deleted.

> ***Note:*** The RDFs prior to Release 11.3 are not compatible with RDFs of Release 11.3 and beyond. Any NRS database containing data populated prior to Release 11.3 must be deleted and repopulated.

Large fields in NRS

Some of the service data fields can be very large. The largest ones encountered so far can be as long as 2045 characters. When two or more of these fields are included in the NRS data files, nawk is unable to handle the record. nawk cannot handle records larger than 3072 characters (3K).

The NRS Populator builds data records that do not exceed 3072 characters. When it encounters a field that would make the record exceed the limit, a warning is displayed and no value is stored in the record for that field.

Currently, only the DPN ITIVIDEOTEXMENU (Menu_Screen) and VIDEOTEXSERVICESIGNAL (Videotex_Service_Signals) components can have large fields.

Multi line string fields

Some service data fields have strings that may contain line feed characters. These characters are replaced by the character with ASCII code 0x1C.

Currently, only the DPN MI8_CUS_MODEM_PROF_ENV (Custom_Modem_Profile) and MI8_PROFILEGROUPPARAMETERS (Profile) components have multi line string fields.

Chapter 10
Creating hierarchical reports

This chapter describes how you can create hierarchial reports. In this chapter, you can find the following information:

- “Overview and prerequisites” (page 151)
- “NRS Hierarchical Report Generator tool” (page 152)
- “Creating a different format of output” (page 156)

Overview and prerequisites

Before creating hierarchical reports, you should be familiar with RDFs. See “Record definition and module data file formats” (page 129) for more information.

Service data is organized into a hierarchy of components. Reports can be created to report on different sub-hierarchies of the service data. Here are a few examples of DPN reports that can be expressed as a sub-hierarchy:

- X25 service report
- ITI service report
- SNA service report
- X25 link component report
- X25 CUG report
- All CUG report
- PE list report
- ICON report

- NCS report
- data spooling report

The NRS Hierarchical Report Generator tool is provided to help create these and many more hierarchical reports for DPN and Passport data.

This chapter provides an overview of this tool. It walks you through an example from the creation to the execution of a report. The format of the output generated by the reports is described as well as a brief section on how to create a different output format.

NRS Hierarchical Report Generator tool

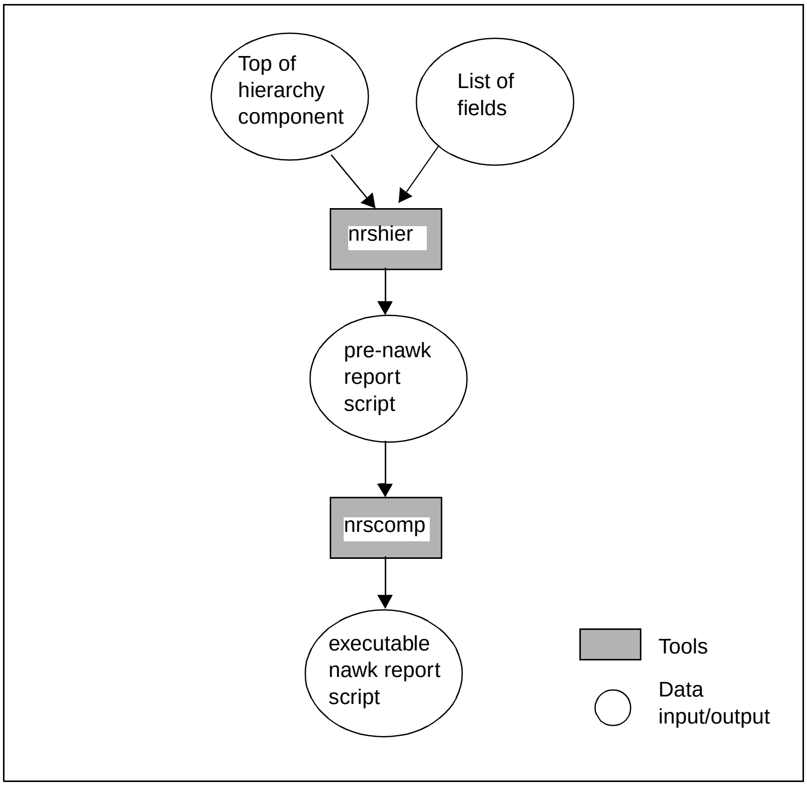
The NRS Hierarchical Report Generator (nrshier) is a tool that can be used to generate reports on DPN and Passport components organized as a sub-hierarchy of the service data.

As shown in the figure “Creating a hierarchical report” (page 153), this tool requires the component that marks the top of the sub-hierarchy be included in the report as well as a list of fields to be printed in the report.

A pre-nawk report script is created corresponding to the specifications. It must be compiled with the NRS Preprocessor (nrscomp) tool to create a nawk report script.

See “NRS Hierarchical Report Generator (nrshier)” (page 269) for a complete description of this tool.

Figure 8
Creating a hierarchical report



Creating a report

This procedure walks you through the steps that are used to generate a report on some of the DPN X.25 fields.

1 Choose the component marking the top of the sub-hierarchy.

In our example, dpn/X25 is the component that represents the top of the sub-hierarchy for the X.25 service. For an ICON report, dpn/ICON_SDA or dpn/CONMAN_SDA would be used. See “Component and field names” (page 140) for information on how to retrieve components from the RDFs.

2 Choose the fields to be included.

The -create_profile option can be specified in the nrshier tool to generate a profile instead of generating the pre-nawk report script.

```
nrshier dpn/X25 -create_profile > x25fields
```

This command generates a file called x25fields which includes all the fields of all the components below X25 in the service data hierarchy. Edit the x25fields file and remove the fields that are not needed in the report. It might be necessary to refer back to the RDFs to better identify the fields.

Assume for the remainder of this procedure that the following fields were kept in the x25fields file.

```
PM _PM
PM NAMSID
PM BUNDLE
PM ACTIVATION_DATE
X25 _PE
X25 _PI
X25 _PO
X25DNACUG _X25MDNA
X25DNACUG BILLING
X25DNACUG TEST
X25DNACUG STUDY
X25DNACUG AUDIT
X25LINK HARDWAREMODE
X25LINK FULLDPLX
X25LINK BUSYOUT
X25LINK LOOP3SYN
X25LINK CLOCKCONFIG
X25LINK LINESPEED
```

3 Create the pre-nawk report script.

The nrshier tool can now be used to generate the pre-nawk report script.

```
nrshier dpn/X25 -profile x25fields > x25.pre_nawk
```

4 Compile the pre-nawk report script.

The NRS Preprocessor (nrscomp) must be used to change the pre-nawk report script into a nawk report script.

```
nrscomp x25.pre_nawk x25.report
```

5 Run the report.

The NRS Reporter (nrsrpt) and all its selection options can be used to run the report.

```
nrsrpt x25.report -dated 930522 > myresult
```

The myresult file contains the report output.

Output format

This section describes the format of the output generated by the reports that are created by the nrshier tool. Here is what the file myresult, from the above example, could look like:

Example

```
Date: Aug 4, 1993

                        Hierarchy report for component dpn/X25
                        ______________________________________

X25 fields
__________

           PM/A21 NAMSID/2101 Bundle/93050101 Activation_Date/930501
           PE/6 PI/6 PO/1

Link fields
___________

           Hardware mode: DTE    Duplex: Full Busy out: 0
           Loop REQ preceded by SYN: 0      Clock config: Normal
           Line speed: 9600

DNA_CUG fields
______________

           DNA/X30212101406100

           Billing: 1            Test: 0    Study: 0
           Audit: 0

Date: Aug 4, 1993

                        Hierarchy report for component dpn/X25
                        ______________________________________


X25 fields
__________
```

```
          PM/A21 NAMSID/2101 Bundle/93050101 Activation_Date/930501
          PE/6 PI/6 PO/2
(...)
```

For each sub-hierarchy being reported on, the current date and title are printed. The title shows the top component that is reported on.

For every component for which fields are requested, a comment is printed (X25 fields). All the key fields that have a value are printed in the form <title>/<value> (except for key fields of type NOKEY. If they have a value, they are printed in the form <title>). Key fields that do not have a value are not printed. For the top component (dpn/X25), fields from the dpn/PM component are printed as if they were keys of that component. All the data fields are then printed in the form <title>: <value> over three columns. Fields that are wider than one column take two columns and fields that are wider than two columns take three.

> ***Note:*** It is possible to change the indentation used when printing key and data fields, the number of columns used for data fields and the maximum line length by modifying the value of the variables INDENTATION, MAX_NO_OF_COLUMNS and MAX_LINE_LEN in the nawk report script before running it.

Customer data

It is possible to include customer data in hierarchical reports. See “Profile rules” (page 271) for more information.

Creating a different format of output

Creating a different format of output requires knowledge of the nawk programming language. The existing print_fields function should be looked at carefully before attempting to create a new one. It is possible to change the format of the report output. Inside the nawk report script, the printing is done by a function called print_fields. This function is included in the nawk report script, using the statement “INCLUDE <nrsprint.inc>” in the pre-nawk report script. It is possible to write a different print_fields function in a file called nrsprint.inc and change the NRS configuration file parameter NRS_INCLUDE_DIR to look at the directory containing that file. The nrsprint.inc file in the /opt/MagellanNMS/lib/nrs/rtp directory should not be modified.

Chapter 11
Programming with NRS

This chapter is for application developers who want to create reports for DPN and Passport data, where the components involved do not represent a hierarchy as described in “Creating hierarchical reports” (page 151). In this chapter, you can find the following information:

Overview

The Network Reporting System (NRS) toolset is designed around the *nawk* pattern scanning and processing language. This language was chosen for its simple C-like structure and ease of use. In fact, some of the NRS tools are written in *nawk*.

We provide a brief introduction to the *nawk* language and NRS extensions for report and tool development use as well as an overview of the report creation process followed by a walk through of two NRS sample reports.

NRS application developers also need to be familiar with the following areas:

- NRS directory structure: See “Installation and configuration” (page 29).
- RDF syntax and module data file format: See “Record definition and module data file formats” (page 129).
- NRS toolset: See “NRS toolset reference” (page 235).

NRS module data files are simple ASCII files that can be processed by any programming language and many third party application packages. However, it is recommended that you become familiar with the existing system before attempting to develop reporting capabilities with other software packages.

nawk pattern matching and processing language

The nawk language is an interpreted pattern scanning and processing language with a syntax much like a simplified C language. Normally a nawk program scans each line of input, which could be a file, standard in, or piped input, to determine if the line matches any patterns specified in the program. When a pattern match is found an action is performed as specified in the program. Any results of the actions that are printed are sent to standard output.

Nawk programs consist of a main loop, shown below in between the BEGIN and END statements, in which input records are read and processed, and optional BEGIN and END statements.

```
BEGIN { ... }
{ ... }
END { ... }
```

All statements defined within the BEGIN statement will be executed once prior to any input records being read. This section can be used to perform operations which are not input data dependent, such as variable initialization. The END statement will be executed once after the last input record is read. The main section of the program, which is not preceded by a BEGIN or END, will be executed once for each input record.

Example

The following example is a complete nawk program which takes as input module service data file(s) and prints any X25 direct call records to standard out, all other records are ignored.

```
BEGIN {
        FS = ”\177”;
}
{
        if( $1 == ”dpn/X25DIRECTCALL” ) print $0
}
```

The variable FS is a nawk built-in variable which defines the input record field separator. In the NRS database the field separator used is the delete character represented in the example as octal 177.

In the program above the nawk interpreter takes care of all the details of reading the input and outputting the results. In the nawk language an entire input record is referenced by $0, each field of the current input line is referenced by $1 ... $n. The code print $0 simply prints the current line, whereas the code print $3 prints the value of the third field of the current input line. The first field in the module data file record is always the component type and the if statement in the example program checks to determine if the first field of each record read is equal to
dpn/X25DIRECTCALL.

For more information on how to run nawk programs, refer to the Sun OS documents listed in the front of this document. Some other sources of information on nawk are the UNIX man pages on nawk, NRS sample reports and the source for the NRS Preprocessor nrsrptpp.

Nawk is useful for developing reports, general queries and building small tools. For example, you can find out how many SNA ports are in your network.

```
BEGIN {
        FS = ”\177”;
        count = 0;
}
```

```
{
        if( $1 == ”dpn/SNA” ) count++
}
END {
        print count
}
```

This example is very similar to the last one with the addition of a variable called count which is incremented every time a dpn/SNA record is encountered. After all input records have been read, the END statement is executed and the value of count is printed. It is a good practice to initialize variables explicitly in the BEGIN section even though their default value is zero or null.

NRS report specification language

To facilitate report writing some new data types and directives have been added to the nawk language. An NRS report program is created by first writing the report program in nawk with the added NRS data types and directives included in the program. The NRS data types allow the programmer to reference service data fields by name rather than offset within a record as well as having access to the parameter attributes (for example, title, width, and so on). This nawk program with the NRS language extensions is then read by the NRS Preprocessor which converts the NRS data types and directives into standard nawk code. The program can then be run.

NRS predefined data types

NRS data types act as identifiers for specific component field attributes.

All NRS data types consist of a component name and a field name separated by a period. All but one NRS identifiers also have an attribute name preceded by a period. If no attribute name is supplied then the identifier references the field’s data value. The component, field and attribute names must be in uppercase. Syntax for referencing a field’s data value is:

```
#<component_name>.<field_name>#
```

For example,

```
#dpn/NETWORK_ENV.X121NTWKDNIC#
```

All other NRS identifiers take the form:

#<component_name>.<field_name>.<attribute_name>#

For example,

#dpn/NETWORK_ENV.X121NTWKDNIC.TITLE#

The attribute name must be one of the supported attributes listed in the table “NRS data types” (page 161).

Table 5
NRS data types

Attributes	Meaning
no attribute	References the data field’s value
POS	Data field’s position in the RDF/module data file
TYPE	Data field’s type
WIDTH	Data field’s maximum width in bytes
TITLE	Data field’s title string
ABBREVIATION	Data field’s abbreviation string
GROUP	Data field’s group string

The following is the potential content of the RDF for the DPN NETWORK_ENV component. This information will be used in examples in this section.

```
# COMPONENT: NETWORK_ENV
# UI_PROMPT: Network_Data
# UI_ABBREVIATION: Network_Data
# OWNERS: PM
# SUBORDINATES:
_name: _COMPONENT
  _type: STRING
  _width: *
  _title: Component
  _abbreviation: Component
  _group:
_name: _HIERARCHY_LEVEL
  _type: NUMERIC
  _width: 2
  _title: Hierarchy_level
  _abbreviation: Hierarchy_level
  _group:
_name: _PM
  _type: STRING
  _width: 12
  _title: PM
  _abbreviation: PM
  _group:
_name: _NETWORK_ENV
  _type: NOKEY
  _width: 1
  _title: Network_Data
  _abbreviation: Network_Data
  _group:
_name: OAM
  _type: STRING
  _width: 10
  _title: Ownership
  _abbreviation: Ownership
  _group:
_name: NETWORKHERALDASCII
  _type: STRING
  _width: 15
  _title: Network herald
  _abbreviation: Network herald
  _group: Network identification
_name: X121NTWKDNIC
  _type: NUMERIC
  _width: 4
  _title: X121 network DNIC
  _abbreviation: X121 network DNIC
  _group: Network identification
_name: E164TELECOUNTRYCODE
```

```
  _type: NUMERIC
  _width: 3
  _title: E164 telephone country code
  _abbreviation: E164 telephone country code
  _group: Network identification
(...)
```

The data field value reference

The data field value reference is one of the most commonly used NRS identifiers. It allows access to a data field's value through the use of the component and field name. The syntax for this is a special case because no attribute name is used. For example to reference the Network Envelopes' DNIC field the NRS identifier name would be:

```
dnic = #dpn/NETWORK_ENV.X121NTWKDNIC#
```

When this statement is processed by the NRS Preprocessor the

#dpn/NETWORK_ENV.X121NTWKDNIC# identifier will be replaced by $x, where x denotes the field position of the X121NTWKDNIC field in both the NETWORK_ENV data record and the NETWORK_ENV RDF.

In the case of the dnic assignment statement above, the resulting code would be:

```
dnic = $5
```

POS attribute

The POS attribute is used to determine a particular field's position in the RDF and data files. This can be used to assign a field's position to a variable. This attribute is very similar to the default field reference attribute, the POS attribute being the actual field position whereas the field reference is the numeric position preceded by a $ character which allows access to the data value.

The POS attribute is used in an assignment of the X121NTWKDNIC field position to a variable. For example, the following string:

```
dnic_position = #dpn/NETWORK_ENV.X121NTWKDNIC.POS#
```

would be replaced by

dnic_position = 5

TYPE attribute

The TYPE attribute refers to the type of values a field will accept. See the table “Field attributes” (page 133) for the list of possible values. For example, the following string:

dnic_type = “#dpn/NETWORK_ENV.X121NEWKDNIC.TYPE#”

would be replaced by

dnic_type = “NUMERIC”

WIDTH attribute

The WIDTH attribute represents the number of characters needed to represent, in ASCII, the longest value for that field. For example, the following string:

dnic_width = #dpn/NETWORK_ENV.X121NTWKDNIC.WIDTH#

would be replaced by

dnic_width = 4

This attribute is used frequently for formatting purposes.

TITLE attribute

The TITLE attribute represents the field name as seen in Component Provisioning. For example, the following string:

dnic_title = “#dpn/NETWORK_ENV.X121NTWKDNIC.TITLE#”

would be replaced by

dnic_title = “X121 network DNIC”

ABBREVIATION attribute

The ABBREVIATION attribute represents the field name as seen in Component Provisioning. For example, the following string:

```
dnic_abbreviation =
“#dpn/NETWORK_ENV.X121NTWKDNIC.ABBREVIATION#”
```

would be replaced by

```
dnic_abbreviation = “X121 network DNIC”
```

GROUP attribute

The GROUP attribute represents the group name as seen in Component Provisioning. For example, the following string:

dnic_group = “#dpn/NETWORK_ENV.X121NTWKDNIC.GROUP#”

would be replaced by

dnic_group = “Network identification”

NRS Preprocessor directives

The NRS Preprocessor recognizes one directive. The include statement which reads an external file into the report source. It must be in uppercase and follow the syntax outlined below.

Include directive

The include statement is a NRS Preprocessor directive to include an external file in the report program. This allows generic functions such as get_date() and page break logic to be included in all programs without manually copying the code into each report. Therefore, if the get_date() function were changed it would only have to be changed in one file. The syntax for the include statement is:

INCLUDE <include_filename>

or

INCLUDE “include_filename”

The include filename must be delimited by brackets (< >) or double quotes (“ ”). Brackets and quotes have different meanings. When brackets are used the NRS Preprocessor uses the NRS configuration file (NRS.cfg) variable NRS_INCLUDE_DIR to find the include file. Quotes can be used to specify the complete filename for the include file.

```
INCLUDE “/nrs/includes/nrsutil.inc”
```

In this case, the NRS Preprocessor looks in the /nrs/includes directory to find the file called nrsutil.inc.

```
INCLUDE <nrsutil.inc>
```

Since brackets are used in this example the NRS Preprocessor will use the NRS.cfg file’s variable NRS_INCLUDE_DIR to determine which directory to search for the file nrsutil.inc.

For a more detailed example, if there was a file called corp_title which contained the following code:

```
function print_corp_title()
{
        print ”\t\t\t\t\t\t” get_date()
        print ”\t\t\t” ”ABC Corporation”
        print
}
```

This function declaration could be included in all reports by adding the line:

```
INCLUDE <corp_title>
```

When the NRS Preprocessor reads the include statement, it writes out the include statement as a comment followed by the include filename’s contents.

```
# INCLUDE ”corp_title”
function print_corp_title()
{
        print ”\t\t\t\t\t\t” get_date()
        print ”\t\t\t” ”ABC Corporation”
        print
}
```

If the NRS Preprocessor does not find the include file an error message is printed and processing stops. There is a sample include file in the /opt/ MagellanNMS/lib/nrs/rdf directory called nrsutil.inc.

Using the NRS Preprocessor

The NRS Preprocessor is invoked with the UNIX Shell script nrscomp found in the /opt/MagellanNMS/bin directory. The preprocessor takes as input an NRS report program containing NRS identifiers and preprocessor directives and creates an executable nawk program.

The NRS Preprocessor is invoked with the following command line syntax:

```
nrscomp <report_file> [<output_file>]
```

See “NRS Preprocessor (nrscomp)” (page 280) for a description on the NRS Preprocessor tool syntax.

The NRS Preprocessor uses the NRS configuration file (NRS.cfg) to find the include file directory and the RDF directory. If no directory name is specified as part of the report program filename in the NRS Preprocessor command line the current working directory is assumed.

Creating a report

Reports generally fall into one of three categories.

- Simple reports: where all the data comes from one component type, such as, the network data report.
- Hierarchical reports: where the data comes from several components types and the components are stored in the data file in their proper hierarchy, such as, the DPN X25 service report.
- Complex reports: where data is extracted form different components, and reorganized into a format suitable for a particular type of report.

This section describes the creation of a simple report, namely the network data report. This sample report is included in the NRS installation. The network data report will be described line by line and serves as an introduction to the nawk language and the NRS extensions.

> ***Note:*** Simple and hierarchical reports can be created using the NRS Hierarchical Report Generator tool.

Sample network data report

This example shows how a program may be written to report on service data fields that are all contained within a single component. Only a few data fields are used in this example in order to keep the number of source lines to a minimum. The actual report, which is stored in the /opt/MagellanNMS/lib/nrs/rpt directory, is called nrsnetsh.rpt.

The first line in the report is:

```
#!/bin/nawk -f
```

which is not a comment line but invokes the nawk interpreter passing the contents of the report file.

Following the nawk invocation line are comment lines giving a report description and copyright notice. The first line of actual source code is the include statement:

```
INCLUDE <nrsutil.inc>
```

which is a directive to the preprocessor to incorporate the content of the nrsutil.inc file into the report source. This file contains the get_date(), underline() and max() functions.

Within the BEGIN statement, which will be executed once before any input records are read, the nawk variable for the input record field separator FS and the output record separator ORS is set to the delete character and null respectively. The nawk default setting for the input field separator (FS) is blank, and all fields in the module service data files are separated by the delete character, therefore the FS variable has to be set to the delete character. The default for ORS is the newline character.

```
FS = ”\177”
ORS = ””;
```

The input record separator (RS) and the output field separator (OFS) variables are left as their defaults of newline and space respectively. The output record separator is set to null to give the program control over when to start a newline.

The next two statements assign the format conversion specification to variables. The format specifications are used to print column headings, underline the column headings, and print data items. The variables in turn are used as parameters to the sprint () function to create a format based on the width of the data to be printed.

```
HDR_FMT = “    %%-%ss”;
HDR_FMT_UNDRLN = “    %%-%s.%ss”;
```

Following the format specifications the variable date is assigned the result of the get_date() function which would be the runtime date from the system.

For each field included in the report, PM, NAMS ID, BUNDLE, NETWORKHERALDASCII, and X121NTWKDNIC, a set of five formatting statements is used to create the format specification for each column title, title underline, and the field’s data value. The format widths are based on the maximum width of the field and title widths from the RDF files. These format specification statements are executed only once before any input records are read because they are all contained within the BEGIN statement. An index variable (fld_cnt) is used to facilitate the addition of new fields.

```
fld_cnt = -1;

# --------- #
# PM fields #
# --------- #

title[ ++fld_cnt ] = “#dpn/PM._PM.TITLE#”;
len_fld            = #dpn/PM._PM.WIDTH#;
max_len            = max( len_fld, length( title
                     [ fld_cnt]));
fmt_both[ fld_cnt ]= sprintf( HDR_FMT, max_len );
fmt_undrln[ fld_cnt ]= sprintf( HDR_FMT_UNDRLN,
                      max_len, max_len );

(. . .)

title[ ++fld_cnt }= “#dpn/NETWORK_ENV.X121NTWKDNIC
                     .TITLE#”;
len_fld          = #dpn/NETWORK_ENV.X121NTWKDNIC.
                    WIDTH#;
max_len            = max( len_fld, length( title
                    [ fld_cnt ] ));
fmt_both[ fld_cnt ]= sprintf( HDR_FMT, max_len );
fmt_undrln[ fld_cnt ]=sprintf( HDR_FMT_UNDRLN,
                       max_len, max_len );
```

Following the format specifications and still within the scope of the BEGIN are a series of print statements which output the report heading which consists of a date line, a report title, and the underlined column headings. In this simple program the heading is printed once at the beginning of the report and is not printed at the top of each new page.

```
printf( “    Date: %s\n”, date );

print( “\n\n” );
print( “\t\t\tNetwork_Data: Short Report\n” );
print( “\t\t\t___________________________\n\n\n” );

for( i = 0, i <= fld_cnt; i++ )
            printf( fmt_both[ i ], title[ i ] );

print( “\n” );

for( i = 0; i <= fld_cnt; i++ )
            underline( fmt_undrln[ i ] );

print( “\n\n” );
```

The main loop of the program starts with an *if* statement which checks the first field of each record to determine if it is a dpn/PM component. In that case, the values for the fields _PM, NAMS ID and BUNDLE are saved. For dpn/NETWORK_ENV, the values for the fields NETWORKHERALDASCII and X121NTWKDNIC are saved and then all 5 values are printed using the format specifications created in the BEGIN section. For components other than dpn/PM and dpn/NETWORK_ENV, the record is ignored.

```
{
  if( $1 == “dpn/PM” )
  {
          fld[ 0 ] = #dpn/PM._PM#;
          fld[ 1 ] = #dpn/PM.NAMSID#;
          fld[ 2 ] = #dpn/PM.BUNDLE#;
  }
  else if( $1 == “NETWORK_ENV” )
  {
          fld[ 3 ] = #dpn/NETWORK_ENV.NETWORKHERALDASCII#;
          fld[ 4 ] = #dpn/NETWORK_ENV.X121NTWKDNIC#;

          # ---------------- #
          # print the fields #
          # ---------------- #

          for ( i = 0; i <= fld_cnt; i++ )
              printf( fmt_both[ i ], fld[ i ] );

          print( “\n” );
  }
}
```

In the following example the nrsnetsh.rpt is transformed into executable nawk code using the NRS Preprocessor and the output is written to a file called *nrsnetsh*. The report is run with the input being what the network looked like on June 12, 1997.

```
nrscomp nrsnetsh.rpt nrsnetsh
nrsrpt nrsnetsh -dated 970612 > result
```

The output redirected to the *result* file might look like this:

```
Date: Jun 12, 1997

    Network_Envelope: Short Report
    ______________________________

PM     NAMSID   BUNDLE    Network herald  X121 network DNIC
___________________________________________________________
ACNT30 2030    97060313   DPN NETWORK       3021
R34    4034    97061004   DPN NETWORK       3021
R70    4070    97051201   DPN NETWORK             3021
```

Chapter 12 Sample reports

This chapter contains sample reports for both DPN and Passport that are provided as part of the Network Reporting System. They serve as examples on how to create other, more useful, reports. In this chapter, you can find the following information:

- "DPN reports" (page 173)
- "Passport reports" (page 174)

DPN reports

This section defines DPN reports.

Network_Data: short report

File: /opt/MagellanNMS/lib/nrs/rpt/nrsnetsh.rpt

This report prints a few fields from the DPN Network_Data (dpn/NETWORK_ENV) component and is presented in a tabular format.

This report must be compiled with the nrscomp tool before being used.

Network_Data: long report

File: /opt/MagellanNMS/lib/nrs/rpt/nrsnetlg.rpt

This report prints all the fields from the DPN Network_Data (dpn/NETWORK_ENV and dpn/NETWORK_ENV_2) component and is presented in a non-tabular format. It also prints the customer data associated with each PM if the customer data file is specified in the NRS configuration file.

This report must be compiled with the nrscomp tool before being used.

X25 Service: short report

File: /opt/MagellanNMS/lib/nrs/rpt/nrsx25sh.rpt

This report prints a few fields from some of the components related to the DPN X.25 service (X25, X25LINK, X25MDNA, X25PVC, X25DIRECTCALL) and is presented in a tabular format.

This report must be compiled with the nrscomp tool before being used.

X25 Service: long report

File: /opt/MagellanNMS/lib/nrs/rpt/nrsx25lg.rpt

This report prints some fields from all the components related to the DPN X.25 service (X25, X25LINK, X25DNACUG, X25INTERFACE, CUG, ICUG, X25PVC, and so on) and is presented in a non-tabular format. It also prints the customer data associated with each X.25 port and X.25 DNA if the customer database file is specified in the NRS configuration file.

This report must be compiled with the nrscomp tool before being used.

Passport reports

This section defines Passport reports. These reports are only extracting data from Passports of type ppc.

Software Configuration report

File: /opt/MagellanNMS/lib/nrs/rpt*/nrsppswc.rpt*

This report prints application version list, logical process type, logical processor and cards. This report must be compiled with the nrscomp tool before being used.

Routing/DPN Address Plan report

File: /opt/MagellanNMS/lib/nrs/rpt/*nrsppadp.rpt*

This report prints the Routing/DPN Address plan which consists of routing ID and module ID in a tabular format. This report must be compiled with the nrscomp tool before being used.

Module Data Short report

File: /opt/MagellanNMS/lib/nrs/rpt/*nrsppmod.rpt*

This report prints node identifier, node name, NAMS ID, network ID code and network ID type in a tabular format. This report must be compiled with the nrscomp tool before being used.

Chapter 13 Running NRS reports

This chapter describes how you can run an Network Reporting System (NRS) report. In this chapter, you can find the following information:

Overview

The Network Reporting System (NRS) toolset provides a simple tool called nrsrpt for running reports. The nrsrpt tool will check the NRS configuration file to find out where the NRS database is located. The program also offers a simple to use command line interface for selecting module data for input and starts the report program. The term *report* refers to any program which uses the NRS service data as input to produce some formatted output. Procedures shown are examples only, your NRS setup and procedures may differ. Your NRS administrator will know what reports are available and the procedures for running them.

Where to find NRS report programs

It is up to your Network Reporting System (NRS) administrator to define the location of the NRS reports. Directory */opt/MagellanNMS/data/nrs/rpt* can be used to keep the NRS report programs.

Running NRS reports

The nrsrpt tool is a convenient way to start report programs and specify the data selection criteria. The nrsrpt tool should be started from your working directory. The default output for the sample report programs is to the screen. The output can be redirected to a file.

The nrsrpt tool allows module data files to be selected for input by specifying command line parameters which correspond to the five parts of the NRS module data filenames. These are the type, PM, NAMS ID, bundle and activation_date. The nrsrpt tool selects the module data files which match the parameters specified. For example, if you wanted all bundles of only module R34, then you would invoke nrsrpt with only the -pm ‘^R34$’ parameter.

Full regular expressions are supported on the selections. For example, to select all module data files for modules beginning with the letter R, you would invoke nrsrpt with the -pm \^R\ .* or -pm ‘^R.*’ parameters. Regular expressions must be in quotes or each special character of the regular expression must be preceded by a backslash (\). See “Regular expressions” (page 325) for more information.

Also, three special options are provided to further reduce the number of files selected. These three options are mutually exclusive:

-dated <date>

To select only one file for every NAMS ID. The file with the activation date closest to (without exceeding) the specified date is selected. The date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-keyed <key>

To select only one file for every NAMS ID. The file with the highest version of the specified key is selected.

-unique

To select only one file for every NAMS ID. The file with the highest bundle (in ASCII) is selected.

The syntax for the nrsrpt command is:

```
/opt/MagellanNMS/bin/nrsrpt <report_program>
[-type <module_type>]
[-pm <pm>]
[-namsid <namsid>]
[-bundle <bundle>|-viewfile <view_filename>]
[-activation_date <activation_date>]
[-dated <date> | -keyed <key> | -unique]
[-h]
[<parameters or options to be passed to the report>]
```

See “NRS Reporter (nrsrpt)” (page 284) for a complete description of these options.

The following command line examples are shown on multiple lines to make them easier to read. These command lines should be entered on a single line. The reports used are for example only and are not provided as part of this feature. The examples assume that the reports are found in the */opt/MagellanNMS/data/nrs/rpt* directory.

Example

In this example the report called x25report is run. Input data is any module data file with a bundle value of MYBUNDLE. Report output will appear on the screen.

```
/opt/MagellanNMS/bin/nrsrpt
  /opt/MagellanNMS/data/nrs/rpt/x25report -bundle
  ‘^MYBUNDLE$’
```

Example

In this example the same X25 report is run but the data consists of the modules with NAMS IDs starting with 40 followed by any two digits. The dated option specifies that only one bundle of each PM be chosen. The one with activation date closest to but not exceeding 930321 (March 21, 1993) is chosen. This example shows that full regular expressions can be used to match module data file names. The output in this example is redirected to a file called x25report.output.

```
/opt/MagellanNMS/bin/nrsrpt
  /opt/MagellanNMS/data/nrs/rpt/x25report -type
  ’^dpn$’ -namsid ‘^40[0-9][0-9]$’ -dated 930321 >
  x25report.output
```

Example

In this example a report called myreport is run. Input is module service data files with an activation date of March 1993. For each module matching that criteria, only the file with the highest ASCII bundle string is selected. This example assumes that the report is expecting a value for the parameter PE. In this case, the PE value is 2.

```
/opt/MagellanNMS/bin/nrsrpt
  /opt/MagellanNMS/data/nrs/rpt/myreport -act
  ‘^9303..’ -unique PE=2
```

Example

In this example, the DC_report is run against the entire database.

```
/opt/MagellanNMS/bin/nrsrpt
  /opt/MagellanNMS/data/nrs/rpt/DC_report
```

Data selections based on organizational data

NRS files do not contain information about the organization of the modules in the network. For example, the concept of regions is not present in the data. However, it is possible to build regular expressions to reproduce these regions.

Consider the following small network organized into two regions:

NAMS ID	Region
4034	1
3401	1
1712	1
4051	2
2020	2

It is possible to run the report called x25report on the modules in region 1 with the following command:

```
/opt/MagellanNMS/bin/nrsrpt
  /opt/MagellanNMS/data/nrs/rpt/x25reports -namsid
  ‘^4034$|^3401$|^1712$’
```

Specifying the modules that are part of the region every time a report is executed can be tedious; especially if a large number of modules are part of a given region. A shell variable can be defined once and then used when running reports.

```
setenv REGION_1 ’^4034$|^3401$|^1712$’
setenv REGION_2 ’^4051$|^2020$’

/opt/MagellanNMS/bin/nrsrpt
  /opt/MagellanNMS/data/nrs/rpt/x25reports -namsid
  $REGION_1
```

More complex data selection techniques

Reports do not need to be started using the nrsrpt tool. Any report can be started by using the report name and specifying the input data. The nrsrpt command finds the location of the NRS database, so if the nrsrpt program is not used the user must specify exactly which data files are to be used as input and where they are located.

Example

In this example the X25report is run without the nrsrpt tool. One module data file, dpn.R34.2001.MYBUNDLE.990412.data, which is located in the /opt/MagellanNMS/data/nrs/data directory is used as input to the X25report. The word cat on the command line is a UNIX command that displays the contents of a file. In this particular case the content of the file is not displayed on the screen but is piped into the report program. The pipe is represented by the | character in the command line. A UNIX pipe is a mechanism which allows the output of one command to be the input to another command. In this case the output of the cat command is piped into the report program. Refer to the UNIX manuals for more details on the pipe function.

```
cat
/opt/MagellanNMS/data/nrs/data/
dpn.R34.2001.MYBUNDLE.990412.data
| /opt/MagellanNMS/data/nrs/rpt/x25report
```

Example

The following example is equivalent to Example 1, except this example uses the *nrsrpt* tool:

```
nrsrpt /opt/MagellanNMS/data/nrs/rpt/x25report
    -type ‘^dpn$’ -pm ‘^R34$’ -namsid ‘^2001$’ -bundle
    ‘^MYBUNDLE$’ -activation_date ‘^990412$’
```

Another method of selecting module data files as input to a report is to create a file with all the module data filenames. This file can be created manually or, depending on the selection criteria, may be created using the *nrslist* tool or with the UNIX tools such as the ls command, which lists files in directories. The following example shows how a module list file could be created using the ls command. The module list file can also be manually created by simply using the vi editor and typing in the file.

Example

This example assumes the data selection criteria is all PMs that start with R3.

```
ls /opt/MagellanNMS/data/nrs/data/dpn.R3*.*.*.*.data
> input_data
```

This creates a file, called input_data, of module data filenames in the following format:

```
/opt/MagellanNMS/data/nrs/data/dpn.R34.4034.519.990411.data
/opt/MagellanNMS/data/nrs/data/dpn.R35.4035.ABCD12.990419.data
/opt/MagellanNMS/data/nrs/data/dpn.R36.4036.93041100.990411.data
/opt/MagellanNMS/data/nrs/data/dpn.R36.4036.93041101.990411.data
/opt/MagellanNMS/data/nrs/data/dpn.R37.4037.93WK1421.990327.data
```

Module data files can be manually changed, deleted, or added to this file by using an editor such as vi.

The input_data file module list can now be used to create the input data stream for a report program.

cat `cat input_data` | /opt/MagellanNMS/data/nrs/data/x25report

This command line involves a cat on the results of another cat command. The second cat command is executed first and it prints the contents of the files in the input_data file. Using each file name in turn the first cat prints the contents of the module data file and this in turn is piped into the report program. As was stated earlier, the input_data file in this example could have been created with the vi editor.

Printing NRS report output

The normal way to run an NRS report is with output redirected to a file. Once the report has completed, the report can be printed with the lpr command.

```
lpr <filename>
```

Refer to UNIX Manual pages for lpr command options.

The sample reports and reports generated by the NRS Hierarchical Report Generator are designed to be printed on portrait format, 80 columns wide. Reports can be printed in landscape format, but some conversion may be required. Format options depend on printer type and available software.

Chapter 14
NRS toolset graphical user interface

The Network Reporting System provides a graphical user interface for the xnrsrpt, xnrsdatah and xnrsdiff tools.

This section describes each of these tools and how to use them:

- “Overview” (page 186) provides a brief description of each of the tools.
- “Accessing the Network Reporting System reports dialogs” (page 188) describes how you can launch the Configuration Report and the Configuration Differences dialogs.
- “Configuration Report (xnrsdatah)” (page 191) describes the Configuration Report dialog and how to use it to produce simple configuration hierarchy reports. This section also describes how to select the component type for the Configuration Report, using the Component Type Selection dialog.
- “Configuration Differences (xnrsdiff)” (page 208) describes the Configuration Differences dialog and how to use it to select two sets of configuration data and produce a report on the differences between the two configurations.
- “NRS Data (xnrsrpt)” (page 220) describes the NRS Data dialog and how to use it to specify the data selection criteria for custom reporting scripts.
- “Selecting NRS data files using the NRS file selection dialog” (page 228) describes the subdialog that helps you select NRS data files for the, Configuration Report, Configuration Differences and NRS Data dialogs

- “Populating the NRS database” (page 232) describes how the NRS database can be dynamically populated from the Configuration Report, Configuration Differences and NRS Data dialogs.

Overview

This section provides an overview of the following tools:

- “Configuration Report (xnrsdatah)” (page 186)
- “Configuration Differences report (xnrsdiff)” (page 187)
- “NRS Data (xnrsrpt)” (page 220)

The NRS Reports toolset

Configuration Report (xnrsdatah)

The xnrsdatah utility provides a graphical user interface, the Configuration Report dialog, that lets you produce simple configuration hierarchy reports, similar to those created by nrsdatah. It has a source data selection capability similar to that of xnrsrpt. xnrsdatah introduces several enhancement to nrsdatah.

They include the following:

- A filtering capability which, if enabled, can filter the output to only show the output that matches the specific component ID instance.
- Output that distinguishes components from their attributes when component filtering is enabled.
- The ability to select a root type for an NRS report from a sub-dialog.

For information on nrsdatah, see “NRS Data Hierarchy Report (nrsdatah)” (page 305).

The Configuration Report dialog is accessed through the Network Reporting System toolset. For information on how to access the Configuration Report, see “Accessing the Network Reporting System reports dialogs” (page 188).

The Configuration Report dialog lets you select the source data for the nrsdatah reporting program on the basis of components, dates, current, committed, and name patterns. You can also directly select the source files and populate the database. It lets you report on fields, which when combined with its filtering capability, reformats the report’s output so that it is more legible. The Select type... push button invokes the type selection dialog which makes it easier for you to select the root type for the report.

If invoked from the command line, the report output may optionally be produced directly to standard output, to a file, or to an xmsg window. If invoked from the MDM Toolsets, or Start Tool menu, the report output is produced directly to an xmsg window. If invoked in CIV through the Diagnostics Commands menu, the report output is to the CIV Diagnostics information pane.

For more information on the Configuration Report dialog, see “Configuration Report (xnrsdatah)” (page 191).

Configuration Differences report (xnrsdiff)

The xnrsdiff utility provides a graphical user interface, the Configuration Differences dialog, that lets you select two sets of configuration data for the nrsdiff reporting program. The nrsdiff utility compares the two configurations and produces a report on the differences. The output is produced directly to standard output, to a file, or to an xmsg window.

The Configuration Differences dialog is accessed through the Network Rerporting System toolset. For information on how to access the Configuration Differences dialog, see “Accessing the Network Reporting System reports dialogs” (page 188).

If invoked from the command line, the report output may optionally be produced directly to standard output, to a file, or to an xmsg window. If invoked from the MDM Toolsets or Start Tool menu, the report output is produced directly to an xmsg window. If invoked in CIV through the Diagnostics Commands menu, the report output is to the CIV Diagnostics information pane.

For more information on the Configuration Differences dialog, see “Configuration Differences dialog” (page 209).

NRS Data (xnrsrpt)

xnrsrpt is a graphical user interface equivalent to nrsrpt. Like its command line equivalent, nrsrpt, the xnrsrpt utility provides a generic means of selecting input for the NRS reporting program. For information on nrsrpt, see “NRS Reporter (nrsrpt)” (page 284).

The NRS Data dialog provides the graphical user interface for specifying the data selection criteria for custom reporting scripts. The xnrsrpt tool selects the module data files which match the parameters specified in the NRS Data dialog.

The dialog is started from your working directory, using the command xnrsrpt. The default output for the report programs is to standard output. See “Command syntax (xnrsrpt)” (page 224) for a description of the NRS Data command line syntax.

Accessing the Network Reporting System reports dialogs

You can access the Configuration Report and Configuration Differences dialogs from following locations:

- “Preside MDM window” (page 188)
- “Start Tool menus” (page 189)
- “Component Information Viewer (CIV) Diagnostics Commands menu” (page 189)

Preside MDM window

The following procedure describes how to open the NRS Reports dialogs from the Preside MDM window:

1 For DPN Devices, in the Preside MDM window, select Configuration -> DPN Devices ->Network Reporting System.

For Passport Devices, in the Preside MDM window, select Configuration -> Passport Devices -> Network Reporting System.

The NRS Reports submenu appears.

2 Select the NRS Reports tool you want to use and then release the mouse button.

The selected dialog opens.

The Configuration Report and Configuration Differences report output appears in a xmsg window. The text can be cut and pasted as well as dragged and dropped.

Start Tool menus

You can access the Configuration Report and the Configuration Differences Report from a DPN or Passport component in some of the Fault tools, specifically Component Information Viewer, Alarm Display and Network Viewer. Press the mouse menu button on the component, select Start Tool -> Configuration -> Network Reporting System, and then select the desired report.

For additional information on accessing the NRS Reports toolset in the fault management tools, see the following sections of 241-6001-011 *Preside MDM Fault Management User Guide*:

- Network Viewer
- Alarm Display
- Component Information Viewer

Component Information Viewer (CIV) Diagnostics Commands menu

The Network Reporting System reports tools are also available in the CIV Diagnostics information pane.You can access them directly with component context for surveillance diagnostic operations. To access the NRS Reports toolset, select Diagnostics from the Information Type options menu, then press the mouse menu button on Commands and point to NRS Reports in the Commands menu.

The figure “Launching NRS Reports from the Component Information Viewer” (page 191) displays how you would access the NRS Reports menu from CIV Diagnostics Command menu.

When you select the desired reporting tool from the Network Reporting System menu, the dialog that opens contains information for the specified component. For example, when you select the Configuration Report tool, the dialog contains the following default settings:

- the Component field contains the name of the target component and subcomponent.
- the Report on fields, Filtered and Current view are set. Current is set to the most recent configuration available.

When you select the Configuration Differences tool, the dialog contains the following default settings:

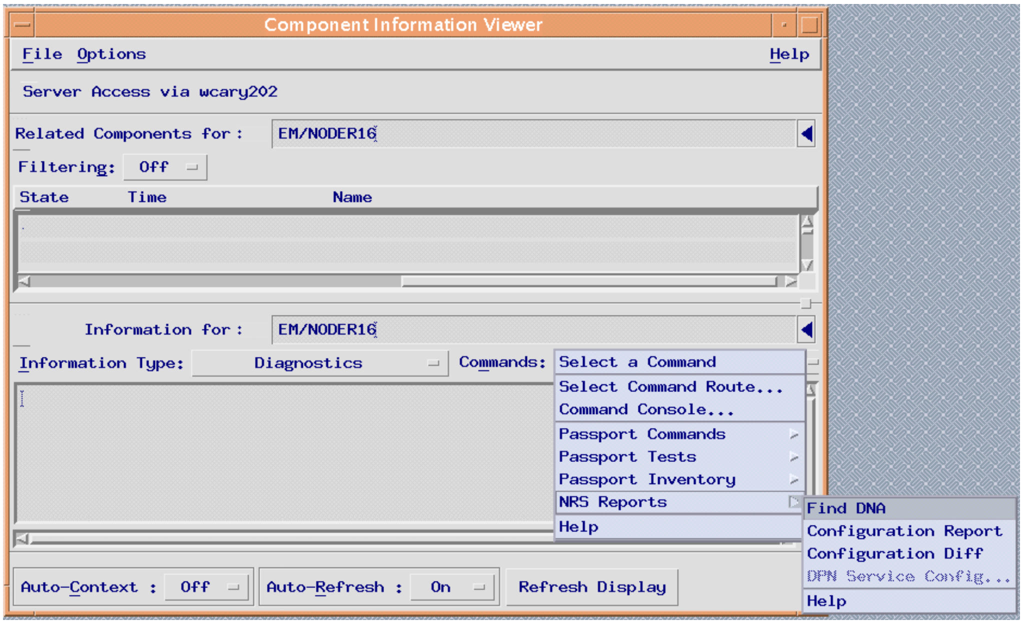
- the Module field is set to the target component,
- In Configuration 1, Dated is set to the latest dated configuration.
- In Configuration 2, Current view is set.

The report output appears in the CIV information text window.

You can generate more than one report by selecting a new component in the CIV Information pane or by pressing Enter in the pane's component entry field.

For additional information on the CIV Diagnostics, see 241-6001-011 *Preside MDM Fault Management User Guide*.

Figure 9
Launching NRS Reports from the Component Information Viewer


Component Information Viewer

Server Access via wcary202

Related Components for : EM/NODER16

Filtering: Off

Information for : EM/NODER16

Information Type: Diagnostics

Commands: Select a Command

Auto-Context : Off

Auto-Refresh : On

Configuration Report (xnrsdatah)

The xnrsdatah tool provides the Configuration Report dialog. The xnrsdatah tool is found in the directory /opt/MagellanNMS/bin.

The Configuration Report dialog lets you produce configuration reports on a specified DPN or Passport component or component type, similar to that produced by the nrsdatah tool. It also lets you produce a report on a specified NRS data file.

If component filtering is selected, xnrsdatah will re-format the output to delineate components and their attributes.

If invoked from the Preside MDM window or Start Tool menu, the report output appears in an xmsg window. It supports cut and paste as well as drag and drop. If invoked from CIV through the Diagnostics Commands menu, the report output is to the CIV Diagnostics information pane.

When you are done with a report, you can press Dismiss to dismiss it and you can generate another report if needed. Since the xmsg window is independent of the reporting GUI, you can also generate another report without dismissing the previous xmsg window, allowing you to review multiple reports at once.

When you have finished creating the reports, you can select Close in the Configuration Report dialog.

For additional information, see the following sections:

- “Accessing the Configuration Report dialog” (page 192)
- “Configuration Report dialog” (page 193)
- “Configuration Report output” (page 198)
- “Using the Configuration Report dialog” (page 199)
- “Selecting the component type for the Configuration Report” (page 200)
- “Command syntax (xnrsdatah)” (page 205)

Accessing the Configuration Report dialog

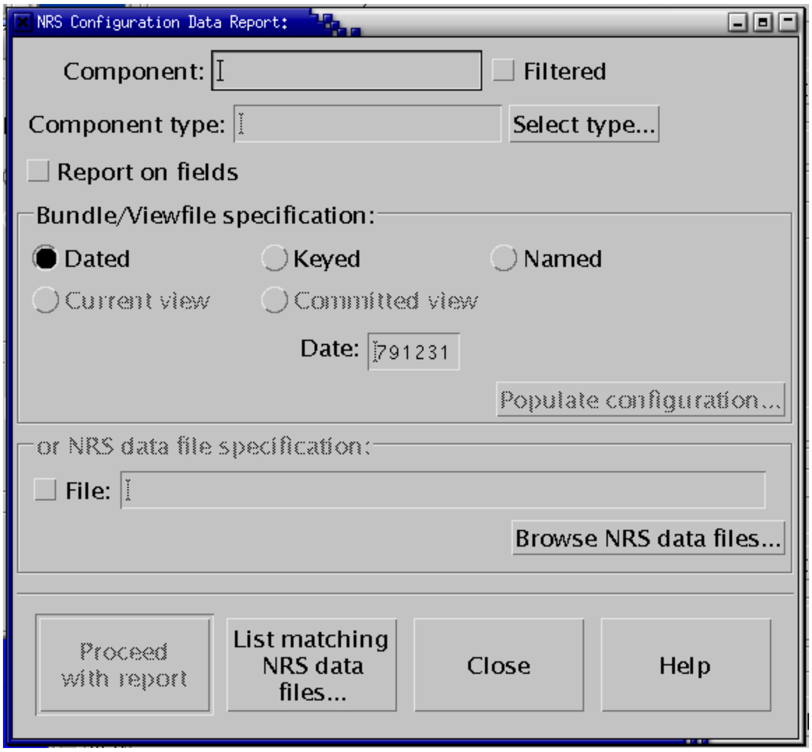
You can access the NRS Configuration Report dialog in the following ways:

- entering the xnrsdatah command line in a UNIX access window. See “Command syntax (xnrsdatah)” (page 205) for command line options for xnrsdatah.
- the Preside MDM window menu
- the Start Tool menu in some fault management tools
- the Component Information Viewer Diagnostics Commands menu

See “Accessing the Network Reporting System reports dialogs” (page 188) for the procedure to invoke the Configuration Report dialog.

The NRS Configuration Report dialog opens as shown in the figure “Configuration Report dialog” (page 193).

Figure 10
Configuration Report dialog



Configuration Report dialog

The Configuration Report dialog consists of the following areas:

- “Component fields area” (page 194)
- “Bundle/Viewfile specification” (page 195)
- “NRS data file specification” (page 197)
- “Command buttons” (page 197)

Component fields area

The Component fields consist of the following:

- The Component field lets you enter a full component ID in display or canonical format. A carriage return in this field validates and reformats the component ID to display format.

  Note: When the Configuration Report is invoked in a component context from the NRS Reports menu in the Component Information Viewer Diagnostics Command menu, or an Start Tools menu, the Component field is set to the selected component and subcomponent.

- The Filtered button enables the filtering mode. Filtered is set by default. With Filtering set, the report output only refers to the specified component or subcomponent.

  Note: For filtering to work properly for DPN devices, ensure that the NRS equivalent of the PE component name is specified in the entry field.

  For DPN devices, the component name specified in the configuration may not match the component name specified in the entry field. For example, a PE component name may be PM/TOTO PE/5, but its NRS equivalent may be PM/TOTO/ PE_NCS_Server/5.

  Therefore, if you use filtering for DPN devices, ensure that you specify the NRS component name in the entry field. Alternatively, you can restrict the reporting by using the Select type... button and select a component type from the Component Type Selection dialog.

- The Component type field lets you enter the root type for the report in the field. You can also use the Select type... button to access a list of root types.
- The Select type button invokes the Component Type Selection dialog to help you select a component type for the current device. See “Configuration Report output” (page 198).
- The Report on fields toggle button enables the field reporting mode. Report on fields is set by default. When Report on fields is enabled, and Filtered is selected, the report is output in a more legible format. The following is an example of output that has been reformatted with Filtered set.

Example

The format

```
Comp_A/A_value Comp_B/B_value Comp_C/C_value
Comp_A/A_value Comp_B/B_value Comp_C/C_value Field_1=1_value
Comp_A/A_value Comp_B/B_value Comp_C/C_value Field_2=2_value
Comp_A/A_value Comp_B/B_value Comp_D/D_value
```

takes the form:

```
Comp_A/A_value Comp_B/B_value Comp_C/C_value
          Field_1=1_value
          Field_2=2_value
Comp_A/A_value Comp_B/B_value Comp_D/D_value
```

...

Bundle/Viewfile specification

The Bundle/Viewfile specification lets you identify the desired configuration data for the specified module. Click on one of the following radio buttons to select a pattern to search for in the configuration data file name:

> ***Note:*** You can only specify one pattern at any one time. The entry field corresponds to the selected radio button.

- Dated is the date for dated bundles. See “Date Convention” (page 27).

  Note: When you access the Configuration Report dialog through the MDM Toolsets window, the Dated specification is set by default to 791231 to report on the most recent configuration available.

- Keyed is the key for keyed bundles. Enter a key in the Key entry field
- Named is the bundle name. Enter the (Grep) pattern in the Pattern entry field.

- Current view is the current active configuration on the node. The tool tries to communicate with the named node to extract the name of the current bundle or viewfile. If a group or OA is not available, a Connection Dialog opens, asking you to connect and identify such a route. The selected configuration appears in the Current viewfile entry field.

  The component type is derived from the module name:

  EM/* --> indicates Passport
  PM/* --> indicates a DPN-100
  By default, a single node name without a type prefix is considered a Passport.

  Note: When the Configuration Report dialog is invoked in the component context from the Component Information Viewer Diagnostics Commands menu or the Start Tool menu, the Current view specification is set by default, to report on the most recent configuration available.

- Committed view is the committed configuration on the node. Like the current view option, the tool tries to communicate with the named node to extract the name of the committed bundle or view file.
- The Populate configuration... button lets you request the immediate population of the configuration matching the specified bundle parameters. When you select Populate configuration... the Authentication Console opens, asking you for a valid Group/OA, UserID and Password to access the device named in the Component field. See “Populating the NRS database” (page 232) for the procedure to populate the database.

  Note: The Authentication Console dialog opens in OA or Passport Group mode, depending on the type of the specified device.

Once the authentication is acknowledged, the database is populated by the nrspop utility for DPN and the pnrspop utility for Passport. The output appears in a separate xmsg window. See the figure “Populate Configuration output window” (page 233) for an example of the output.

NRS data file specification

The NRS data file specification area of the Configuration Report dialog performs in the same way as in the NRS Data and Configuration Differences dialogs. You can specify the NRS data file or files that you want to use by either entering the NRS data file name(s) in the file text field or by selecting a file from the data file selection dialog. The NRS data file specification area contains the following:

- The File toggle button enables the NRS data file specification.
- The File text field lets you directly enter one or more NRS data files to be used. The entry can be specified in GLOB style. You are not required to specify the full path name. If you do not specify the path name, the default is /opt/MagellanNMS/data/nrs/data.
- The Browse NRS data files... push button invokes the NRS data file selection dialog to help you select a file. The file that you select appears in the File text field. For additional information on how to use the file selection dialog, see “Using the NRS Data File Selection dialog” (page 230).

Command buttons

The Configuration Report dialog has the following command buttons:

- Proceed with report invokes the nrsdatah reporting program, passing it the matching data.
- List matching NRS data files... invokes an xmsg window that provides the list of matching NRS data files, if any. This lets you validate the current dialog specifications. For explicit file specifications, the matching files are simply listed. If you have used the Bundle/Viewfile specification, nrslist provides the list of matching files.
- The Close command closes the dialog and ends the application without producing a report.
- Help invokes the help panel for the tool.

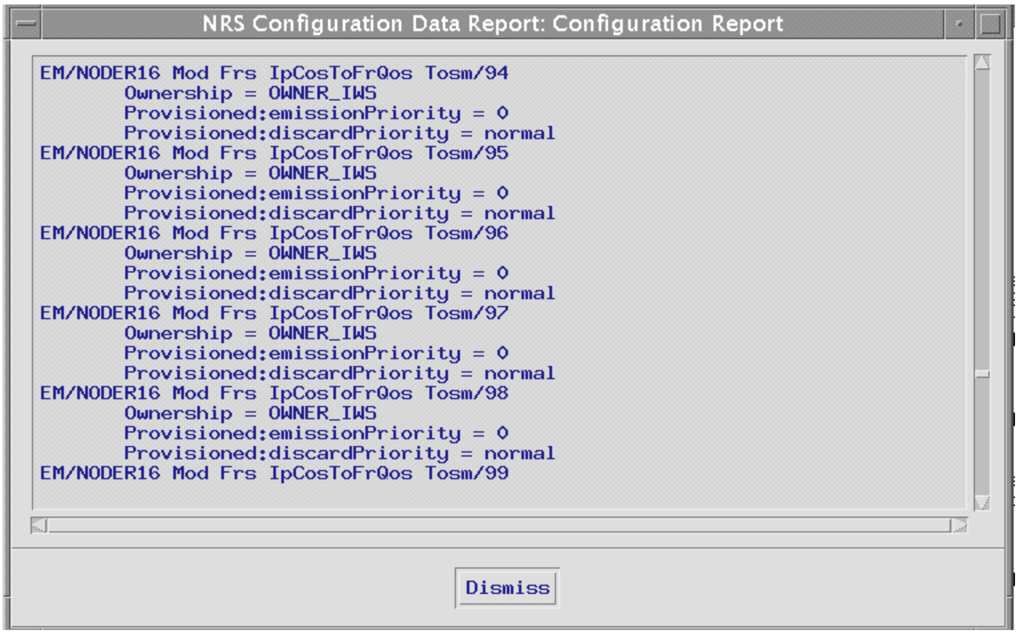
Configuration Report output

The following is an example of the output generated by the Configuration Report:

```
EM/NODER16 Shelf Card/10 Diag TrapData
       Ownership = OWNER_IWS
EM/NODER16 Shelf Card/10 Diag RecErr
       Ownership = OWNER_IWS
EM/NODER16 Shelf Card/11
       Ownership=OWNER_IWS
       Provisioned:cardType = none
       Provisioned:sparingConnection = notApplicable
       Provisioned:commentText =
(...)
```

The figure “Configuration Report output” (page 198) shows an example of the Configuration Report output.

Figure 11
Configuration Report output



NRS Configuration Data Report: Configuration Report

```
EM/NODER16 Mod Frs IpCosToFrQos Tosm/94
       Ownership = OWNER_IWS
       Provisioned:emissionPriority = 0
       Provisioned:discardPriority = normal
EM/NODER16 Mod Frs IpCosToFrQos Tosm/95
       Ownership = OWNER_IWS
       Provisioned:emissionPriority = 0
       Provisioned:discardPriority = normal
EM/NODER16 Mod Frs IpCosToFrQos Tosm/96
       Ownership = OWNER_IWS
       Provisioned:emissionPriority = 0
       Provisioned:discardPriority = normal
EM/NODER16 Mod Frs IpCosToFrQos Tosm/97
       Ownership = OWNER_IWS
       Provisioned:emissionPriority = 0
       Provisioned:discardPriority = normal
EM/NODER16 Mod Frs IpCosToFrQos Tosm/98
       Ownership = OWNER_IWS
       Provisioned:emissionPriority = 0
       Provisioned:discardPriority = normal
EM/NODER16 Mod Frs IpCosToFrQos Tosm/99
```

Using the Configuration Report dialog

The following procedure describes how to use the Configuration Report dialog.

1 Launch the Configuration Report dialog. See “Accessing the Network Reporting System reports dialogs” (page 188).

The Configuration Report dialog opens with Filtered, Report on fields, and Dated set.

Note: With Filtered and Report on fields set, the output only reports on the specified component ID instance and its subordinates, and the information is displayed in a more legible format. The output also reports on fields. If you deselect Filtered and Report on fields, xnrsdatah provides a full report on the specified component and its subcomponents. It does not contain information on the fields.

2 In the Component text field, enter a valid component name.

3 In the Component type field, enter a component type or Click Select type... to open the Component Type Selection dialog to help you select a proper type to use for reporting. See “Selecting the component type for the Configuration Report” (page 200) for a description of the dialog and how to use it.

4 Select the NRS data file by either using the Bundle/Viewfile specification area or specifying a NRS data file in the NRS data file specification field:

In the Bundle/Viewfile specification area, select one of the following options. By default Dated is set when the dialog opens.

- Dated. The default date of 791231 appears in the Date text field unless you specify another date. See “Date Convention” (page 27).
- Keyed. Enter the key in the Key text field.
- Named. Enter the name in the Pattern text field.
- Current view. The Current Viewfile field is filled with the name of the active configuration.
- Committed view. the Committed Viewfile field is filled with the name of the committed configuration.

Note: If you select Current or Committed viewfile, and no group or OA connection is available, a Connection Console dialog opens prompting you to connect and identify a route. Connect and select the appropriate group or OA.

- In the NRS data file specification pane, either type the NRS data file in the File text field or click Browse NRS data files... and select the desired NRS data file. The File toggle button is set. See “Selecting NRS data files using the NRS file selection dialog” (page 228) for a description of the data file selection dialog and how to use it.

5 Click Proceed with report.

If invoked from the NRS Toolset menu or the Start Tool menus, the Configuration report is output to an xmsg window. If invoked from the CIV Diagnostics Commands menu, the report is output to the CIV Information pane.

Note: If no matching NRS data file(s) exist for the specified module, a dialog opens saying no matching files were found to report on and asking you if you want to populate them from the device. Click OK. Clicking OK invokes the Authentication Console dialog. Perform the Authentication procedure. For the procedure to populate the configuration, see “Populating the NRS database” (page 232).

Alternately, you can populate the database before clicking Proceed with report, by using the Populate configuration... command. Clicking Populate configuration... invokes the Authentication Console dialog. Perform the procedure described in “Populating the NRS database” (page 232)

Selecting the component type for the Configuration Report

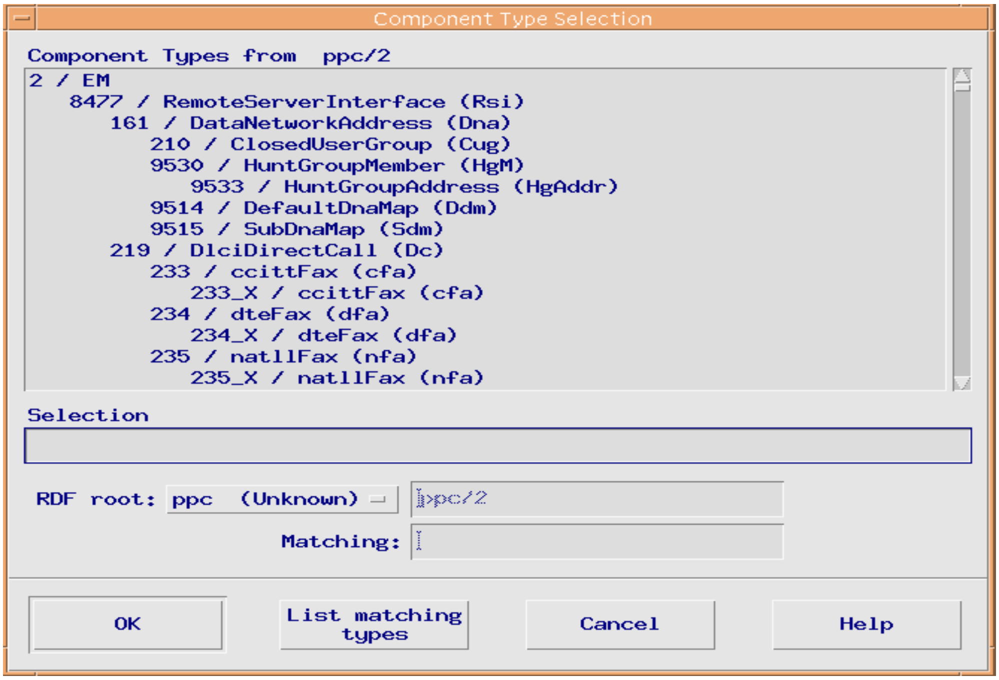
This section describes how you can select the component type for the Configuration Report by using the Component Type Selection dialog.

For additional information see the following sections:

- “Component Type Selection dialog” (page 201)
- “Using the Component Type Selection dialog” (page 204)

The Component Type Selection dialog is shown in the figure “Component Type Selection dialog” (page 201).

Figure 12
Component Type Selection dialog



```
Component Types from  ppc/2
2 / EM
   8477 / RemoteServerInterface (Rsi)
      161 / DataNetworkAddress (Dna)
         210 / ClosedUserGroup (Cug)
         9530 / HuntGroupMember (HgM)
            9533 / HuntGroupAddress (HgAddr)
         9514 / DefaultDnaMap (Ddm)
         9515 / SubDnaMap (Sdm)
      219 / DlciDirectCall (Dc)
         233 / ccittFax (cfa)
            233_X / ccittFax (cfa)
         234 / dteFax (dfa)
            234_X / dteFax (dfa)
         235 / natlIFax (nfa)
            235_X / natlIFax (nfa)
```

Component Type Selection dialog

The Component Type Selection dialog lets you select the component type that you want to use for reporting in the Configuration Report. You invoke the Component Type Selection dialog by pressing Select type... in the Configuration Report dialog. This dialog lets you search for the appropriate type by focusing on specific sub-roots and providing a filtering capability. The list that is produced displays the component types matching the root and filter specifications.

The Component Type Selection dialog consists of the following areas:

- “Component Types list area” (page 202)
- “Selection field” (page 202)
- “RDF root filter area” (page 202)

- “Matching text field” (page 203)
- “Command Buttons” (page 204)

Component Types list area

This area lists the available component types and their subcomponents. When you specify the root, using the As specified: option in the RDF root field, the dialog will list only the specified component type and all of its subcomponent types, subject to the filters that you have specified in the Matching: text field.

Selection field

The Selection field displays the selected component or subcomponent type. Clicking a component type in the Component Types area enters the selected component type in the Selection field.

RDF root filter area

The RDF root filter area consists of the following:

- RDF root

  Note: See “Record definition and module data file formats” (page 129) for additional information on Record Definition Files (RDF).

  RDF root provides access to an option menu that lists the possible RDF roots:

  — dpn: DPN-100

  — ppc: CDN versions of Passport

  — ppe: MED versions of Passport

  — As specified

The possible RDF roots for Passport are found in PPTypeConfig.cfg file. If additional Passport versions are supported, they are automatically included from the PPTypeConfig.cfg file of Preside Multiservice Data Manager (MDM).

When you select a RDF root from the menu, the root component type appears (greyed out) in the neighboring text entry field.

If you select As specified, the neighboring text entry field is enabled letting you enter the desired root component type.

Its value can be:

For a DPN-100 type, specified as dpn/<type name>.
For a Passport type, specified as ppc/<FMIP ID> or ppe/<FMIP ID>

Matching text field

The Matching text field provides a filtering capability. It lets you specify a matching type. This filters out all the types except those matching the type specified in the Matching field. If the matching field is left empty, the Component Types area will display the types hierarchically from the specified root.

When you enter a matching type in the Matching field, the list of component types from the specified root is scanned to identify all matching entries. The list that results is purged of all duplicates. nrsowner is then invoked on each resulting entry to display the possible type parentages. The concatenation of all these results is provided in the list. The following is an example of a how the list is displayed.

Specifying a root type of ppc/2 or ppe/2, and a matching type of ds1 produces the following in the Component Types area:

```
2  /EM
    2684 / SignallingChannel (SigChan)
       2521 / Ds1Cas

2  /EM
    371 / Logical Processor (Lp)
       510 / DS1

2  /EM
    371 / Logical Processor (Lp)
       528 / DS3
          510 / DS1
```

If the Matching field is left empty, the Component Types list will display all of the types hierarchically from the specified root.

Command Buttons

The Component Type Selection dialog has the following command buttons:

- OK transfers the selected type to the Component Type field in the Configuration Report dialog.
- List matching types refreshes the contents of the Component Types area of the dialog to list the matching types.
- Cancel closes the dialog without changing the Component Type field in the Configuration Report dialog.
- Help invokes the on-line help for the Component Type Selection dialog.

Using the Component Type Selection dialog

The following procedure describes how to use the Component Type Selection dialog:

1 In the Configuration Report main window, specify a Component.

2 Invoke the Component Type Selection dialog by clicking Select Type... in the Configuration Report dialog.

The Component Type Selection dialog opens.

3 From the RDF root options menu, select the desired RDF root.

4 Click List matching types.

A list of component types is displayed in the Component types area of the dialog.

5 If desired, enter a matching type in the Matching field to restrict the listing to the specified type. Otherwise proceed to step 7.

6 Click List matching types to refresh the contents of the list.

The Component Types area displays a list of the types and their parents that match the type specified.

Note: If a matching type is not entered, the Component Types area displays a hierarchical list of all component types for the specified root.

7 In the Component Types area, click on the desired type.

The selected type appears in the Selection text field.

8 Click OK to transfer the selected type to the Component type field in the Configuration Report dialog.

The Component Type Selection dialog closes, and the selected type appears in the Component type field of the main dialog, with a dpn, ppe, or ppc prefix, depending on the selected component type/root.

Note: Alternatively, you can double click on the desired type to transfer it to the Component type field in the Configuration Report dialog.

Command syntax (xnrsdatah)

Enter the following command syntax as one continuous command:

```
xnrsdatah
  [-ask|-noask]
  [-stay]
  [-nofile]
  [-o <output file> | -x]
  [-title <report title>]
  [-pm|-spm <target component>]
  [-filtered]
  [   -file <file>
    | -dated <date>
    | -keyed <key>
    | -named <name> [-unique]
    | -current
    | -committed]
  [-fields]
  [<component type>]
  [-h]
```

with supported abbreviations:

```
current:  cu
dated:    da
file:     fi
filtered: filt
keyed:    ke
named:    na
report:   re
unique:   un
```

where:

`-ask | noask`

Option for specifying if the Configuration Report dialog is shown. If -noask is used, the GUI will not be shown and the data selection is made exclusively from these command line options.

`-stay`

Option that indicates that the selection dialog should stay once the report is invoked. By default it drops and xnrsdatah terminates. Specifying -stay allows the launching of multiple reports.

`-nofile`

Specifying -nofile prevents the explicit selection of NRS data files by hiding the corresponding dialog area. The source data can only be selected by bundle/viewfile specification.

`-o <output file>|-x`

if -o is specified with the output file path, the report's output is sent to the named file. If -x is used, the output is sent to an independent xmsg window. The default is to send the report output to the standard output stream if neither -o or -x are used.

`-title <text>`

Option used to override the default window title to the one specified.

`-pm|-spm <component ID>`

An option specifying a default value of the component ID. A full component ID in either Display or API canonical format can be specified. If -spm is specified instead of -pm, you will not be able to change the component ID in the dialog.

`-filtered`

If the filtered option is set, the default value for the Filtered toggle button is set.

`-file <file name>`

Option for specifying a default NRS data file name/pattern to the dialog. The corresponding mode is adopted.

`-dated <date>`

An option specifying a default dated pattern to the dialog, which then adopts the bundle/viewfile mode.

`-keyed <key>`

An option specifying a default keyed pattern to the dialog which then adopts the bundle/viewfile mode.

`-named <name>`

An option specifying a default NRS data file name/pattern to the dialog, which then adopts the corresponding mode.

`-unique`

Option that if specified, is passed down to the xnrsrpt utility to provide the unique file mappings from the specified bundle/viewfile patterns.

-current|-committed

If a module is also specified (-pm or -spm), the dialog automatically fetches its current or committed bundle/viewfile name. This is the same as selecting the Current or Committed view radio button.

`-fields`

An option that if set, the default value for the Fields toggle button is set.

`<component type>`

An option specifying a default value for the root component type for the report. If specified, the component type must be the last option on the command line.

`-h`

Invokes a help panel for the tool (standard error stream).

Example

```
xnrsdatah -dated 791231 -pm PM/R78
```

This command invokes the NRS Configuration Report tool to produce a report on node PM/R78 for its most recent dated configuration. The report output is sent to standard output.

```
xnrsdatah -x -stay
```

This command invokes the NRS Configuration Report tool with no previous selections. The tool stays up once reports are launched and the reports themselves are produced in a separate xmsg window.

```
xnrsdatah -stay -x -pm EM/TOTO -current -fields
-filtered ppc/586
```

This command invokes the tool to produce a report on the named module for its current configuration for its Framer components. The output is produced in a separate window. The report output contains field information and is filtered for legibility.

Configuration Differences (xnrsdiff)

The xnrsdiff tool compares two device configurations and reports what has changed between them. This tool is found in the /opt/MagellanNMS/bin directory. xnrsdiff builds on the xnrsrpt configuration selection form and duplicates it to allow the selection of two sets of configuration data.

For additional information, see the following sections:

- “Accessing the Configuration Differences dialog” (page 209)
- “Configuration Differences dialog” (page 209)
- “Configuration Differences report output” (page 213)
- “Using the Configuration Differences report dialog” (page 215)
- “Command syntax (xnrsdiff)” (page 216)

Accessing the Configuration Differences dialog

You can access the NRS Configuration Differences dialog in the following ways:

- entering the xnrsdiff command line in a UNIX access window. See "Command syntax (xnrsdiff)" (page 216) for command line options for xnrsdiff.
- the MDM Toolsets menu
- the Start Tool menu in some fault management tools
- the Component Information Viewer Diagnostics Commands menu

See "Accessing the Network Reporting System reports dialogs" (page 188) for the procedure to invoke the Configuration Differences dialog.

The Configuration Differences dialog opens as shown in the figure "Configuration Differences dialog" (page 210).

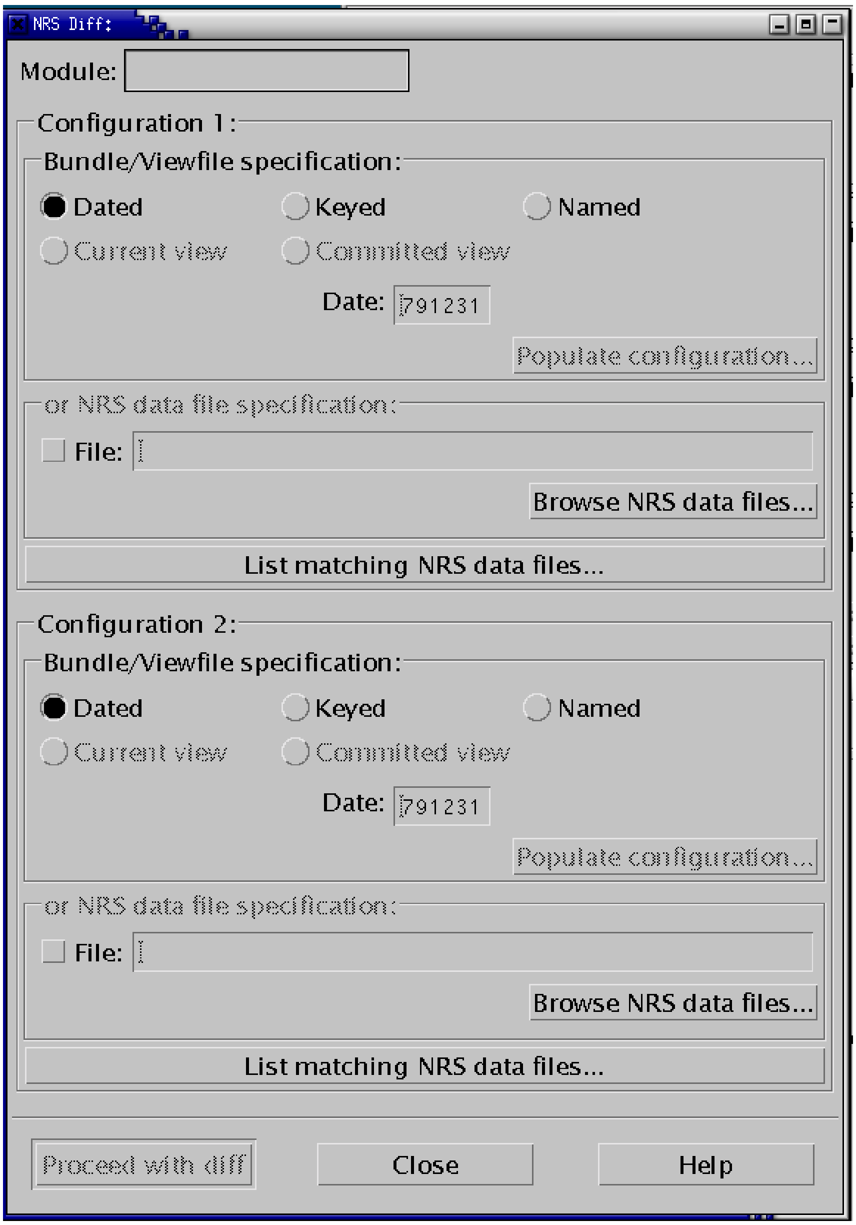
Configuration Differences dialog

The Configuration Differences dialog provides a graphical user interface to select two sets of configuration data and output the differences between the two configurations.

The Configuration differences dialog consists of the following areas:

- "Module text field" (page 211)
- "Configuration 1 and Configuration 2 panes" (page 211)
- "Command buttons" (page 213)

Figure 13
Configuration Differences dialog



Module text field

The Module text field lets you enter the node name. This field is common to both Configuration 1 and Configuration 2. Therefore, the bundle/viewfile configuration specifications can only refer to this one node.

> ***Note:*** When the dialog is invoked in the component context from the Component Information Viewer Diagnostics Commands menu or the Start Tool menus, the Module field is set to the selected component.

Configuration 1 and Configuration 2 panes

The Configuration Differences dialog lets you select two sets of configuration data for the xnrsdiff reporting program. The Configuration 1 pane lets you select the first set of configuration data. The Configuration 2 pane lets you select the second set of configuration data. Both Configuration 1 and Configuration 2 contain the following areas:

- Bundle/Viewfile specification lets you identify the desired configuration data for the specified module. You have the following options:

  Note: The module type is derived from the module name:
  EM/* --> indicates Passport
  PM/* --> indicates a DPN-100
  By default, a single node name without a type prefix is considered a Passport.

  — Dated is the date for dated bundles. See “Date Convention” (page 27).

  — Keyed is the key for keyed bundles. Enter the key in the Key entry field

  — Named is the bundle name. Enter the (Grep) pattern in the Pattern entry field.

  — Current view is the current active configuration on the node. The tool tries to communicate with the named node to extract the name of the current bundle or viewfile. If a group or OA connection is not available, a Connection Dialog opens, asking you to connect and identify such a route. The selected configuration appears in the Current viewfile entry field.

    - Committed view is the committed configuration on the node. As for current view, the tool tries to communicated with the named noe to extract the name of the committed bundle or viewfile.

- The Populate configuration... push button lets you request the immediate population of the configuration matching the specified bundle parameters. When you select Populate configuration... the Authentication Console opens, asking you for a valid Group/OA, UserID and Password to access the device named in the Module field. See “Populating the NRS database” (page 232) for the procedure to populate the database.

  The Authentication Console dialog opens in OA or Passport Group mode, depending on the type of the specified device.

  Once the authentication is acknowledged, the database is populated by the nrspop utility for DPN and the pnrspop utility for Passport. The output appears in a separate xmsg window. See the figure “Populate Configuration output window” (page 233) for an example of the output.

When the dialog is invoked from the MDM Toolsets window, the Dated specification is set by default in both the Configuration 1 and Configuration 2 panes, to 791231 to report on the most recent configuration available.

When the dialog is invoked in the component context from the Component Information Viewer Diagnostics Commands menu or the Start Tool menu, the Dated specification is set by default to the latest dated configuration in the Configuration 1 pane. In the Configuration 2 pane, the Current view specification is set by default, to report on the current active configuration.

NRS data file specification (Configuration 1 and Configuration 2)

An NRS data file specification pane is found in both the Configuration 1 and Configuration 2 panes. The NRS data file specification areas of the Configuration Differences dialog perform in the same way as in the NRS Data dialog and the Configuration Report dialog. You can specify the NRS data file or files that you want to use by either entering the NRS data file name(s) in the file text field or by selecting a file from the data file selection dialog.

The NRS data file specification area contains the following:

- The File toggle button enables the NRS data file specification.
- The File text field lets you directly enter one or more NRS data files to be used. The entry can be specified in GLOB style. You are not required to specify the full path name. If you do not specify the path name, the default is /opt/MagellanNMS/data/nrs/data.
- The Browse NRS data files... push button invokes the NRS data file selection dialog to help you select a file. The file that you select appears in the File text field. For additional information on how to use the file selection dialog, see “Using the NRS Data File Selection dialog” (page 230).
- The List matching NRS data files... lets you validate the specifications for Configuration 1 or Configuration 2, depending on which one is selected. Clicking List matching NRS data files results in a xmsg window listing the matching files for the configuration.

Command buttons

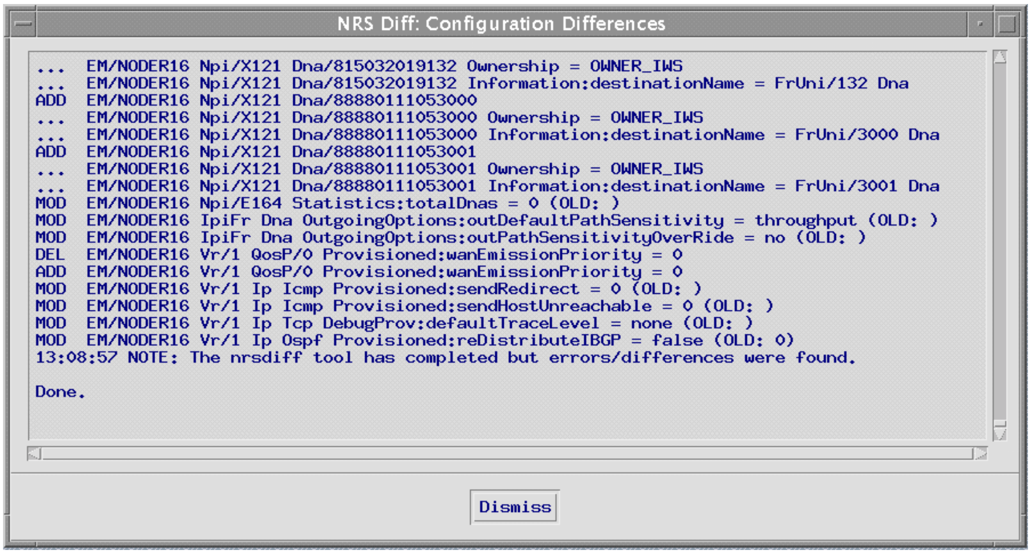
The Configuration Differences dialog has the following command buttons:

- The Proceed with diff command button invokes the nrsdiff reporting utility.
- Close exits the dialog without generating a report.
- Help invokes the on-line help for the Configuration Differences dialog.

Configuration Differences report output

The figure “Configuration Differences report output” (page 214) shows an example of the Configuration Differences report output.

Figure 14
Configuration Differences report output



```
...  EM/NODER16 Npi/X121 Dna/815032019132 Ownership = OWNER_IWS
...  EM/NODER16 Npi/X121 Dna/815032019132 Information:destinationName = FrUni/132 Dna
ADD  EM/NODER16 Npi/X121 Dna/88880111053000
...  EM/NODER16 Npi/X121 Dna/88880111053000 Ownership = OWNER_IWS
...  EM/NODER16 Npi/X121 Dna/88880111053000 Information:destinationName = FrUni/3000 Dna
ADD  EM/NODER16 Npi/X121 Dna/88880111053001
...  EM/NODER16 Npi/X121 Dna/88880111053001 Ownership = OWNER_IWS
...  EM/NODER16 Npi/X121 Dna/88880111053001 Information:destinationName = FrUni/3001 Dna
MOD  EM/NODER16 Npi/E164 Statistics:totalDnas = 0 (OLD: )
MOD  EM/NODER16 IpiFr Dna OutgoingOptions:outDefaultPathSensitivity = throughput (OLD: )
MOD  EM/NODER16 IpiFr Dna OutgoingOptions:outPathSensitivityOverRide = no (OLD: )
DEL  EM/NODER16 Vr/1 QosP/0 Provisioned:wanEmissionPriority = 0
ADD  EM/NODER16 Vr/1 QosP/0 Provisioned:wanEmissionPriority = 0
MOD  EM/NODER16 Vr/1 Ip Icmp Provisioned:sendRedirect = 0 (OLD: )
MOD  EM/NODER16 Vr/1 Ip Icmp Provisioned:sendHostUnreachable = 0 (OLD: )
MOD  EM/NODER16 Vr/1 Ip Tcp DebugProv:defaultTraceLevel = none (OLD: )
MOD  EM/NODER16 Vr/1 Ip Ospf Provisioned:reDistributeIBGP = false (OLD: 0)
13:08:57 NOTE: The nrsdiff tool has completed but errors/differences were found.

Done.
```

Each line in the list of differences starts with one of the following:

- MOD

  A field has a different value in the two NRS data files. The field is displayed with the new value, followed by (OLD:<old value>).

- ADD

  A component is present in the Configuration 2 NRS data file but not in the Configuration 1 file

- DEL

  A component is present in the Configuration 1 NRS data file but not in the Configuration 2 file.

- ...

  Indicates a component or field was added or deleted as part of a higher level add or delete.

Using the Configuration Differences report dialog

1 Launch the Configuration Differences report.

The Configuration Differences dialog opens.

For information on accessing the Configuration Differences report, see “Accessing the Network Reporting System reports dialogs” (page 188).

2 In the Module text field, enter a Passport or DPN name.

3 For both Configuration 1 and Configuration 2, select the NRS data file by either using the Bundle/Viewfile specification area or specifying a NRS data file in the NRS data file specification field:

In the Bundle/Viewfile specification area, select one of the following:

- Dated. The default date of 791231 appears in the Date text field unless you specify another date. See. “Date Convention” (page 27).
- Keyed. Enter the key in the Key text field.
- Named. Enter the name in the Pattern text field.
- Current view. The Current Viewfile field is filled with the name of the active configuration.
- Committed view. the Committied Viewfile field is filled with the name of the committed configuration.

Note: If you select Current viewfile, and no group or OA connection is available, a Connection Console dialog opens prompting you to connect and identify a route. Connect and select the appropriate group or OA.

In the NRS data file specification pane, click the File toggle button and either type the NRS data file in the File text field or click Browse NRS data files... and select the desired NRS data file. See “Selecting NRS data files using the NRS file selection dialog” (page 228) for a description of the data file selection dialog and how to use it.

4 Click Proceed with diff.

If invoked from the MDM Toolsets menu or from the Start Tool menus, the Configuration Differences report, displaying the differences between the two configurations, is output to an xmsg window. If invoked from the CIV Diagnostics Commands menu, the report is output to the CIV Information pane.

If no matching NRS data file(s) exist for the specified module, a dialog opens saying no matching files were found to report on and asking you if you want to populate them from the device. Click OK. Clicking OK invokes the Authentication Console dialog. Perform the Authentication procedure. For the procedure to populate the configuration, see “Populating the NRS database” (page 232).

Alternately, you can populate the database before clicking Proceed with diff, by using the Populate configuration... command. Clicking Populate configuration... invokes the Authentication Console dialog. Perform the procedure described in “Populating the NRS database” (page 232).

Command syntax (xnrsdiff)

Enter the following command line syntax as one continuous command:

```
xnrsdiff
[ask|noask]
[-stay]
[-nofile]
[-o <output file> | -x]
[-title <report title>]
[-pm <target module> | -spm <target module>]
[ -file1 <file>
 |-dated1 <date>
 |-keyed1 <key>
 |-named1 <name>
 |-current1
 |-committed1]
[ -file2 <file>
 |-dated2 <date>
 |-keyed2 <key>
 |-named2 <name>
 |-current2
 |-committed2]
[-log [<log file name>]]
[-component <component name>]
[-keepfiles <directory name>]
[-quiet]
[-h]
```

Minimum abbreviations:

current1: cu1
current2:cu2
dated1: da1
dated2: da2

keyed1:	ke1
keyed2:	ke2
named1	na1
named2:	na2
file1:	fi1
file2:	fi2
component:	co
keepfiles:	ke
quiet:	qu

where:

`-ask | noask`

Option for specifying if the Configuration Report dialog is shown. If -noask is used the data selection is made exclusively from these command line arguments. the default is -ask.

`-stay`

Option that indicates that the selection dialog should stay once the report is invoked. By default it drops and xnrsdiff terminates. Specifying -stay allows the launching of multiple reports.

`-nofile`

Specifying -nofile prevents the explicit selection of NRS data files by hiding the corresponding dialog areas. The Configuration 1 and Configuration 2 data can only be selected by bundle/viewfile specification.

`-o <output file> | -x`

if -o is specified with the output file path, the report's output is sent to the named file. If -x is used, the output is sent to an independent xmsg window. The default is to send the report's output to the standard output stream if neither -o or -x are used

`-title <text>`

Option used to override the default window title to the one specified.

`-pm|-spm <target module>`

An option specifying a default value of the target module. A full component ID in either Display or API canonical format can be specified. If -spm is specified instead of -pm, you will not be able to change the module in the dialog.

`-file1|-file2 <file name>`

Option for specifying a default NRS data file name/pattern to the Configuration 1 and Configuration 2 pane respectively. The corresponding mode is adopted by the pane.

`-dated1|-dated2 <date>`

An option specifying a default dated string to the Configuration 1 and Configuration 2 pane respectively, which then adopts the bundle/viewfile mode.

`-keyed1|keyed2 <key>`

An option specifying a default keyed string to the Configuration 1 and Configuration 2 pane respectively, which then adopts the bundle/viewfile mode.

`-named1|named2 <name>`

An option spefiying a default named string to the Configuration 1 and Configuration 2 pane respectively, which then adopts the corresponding bundle/viewfile mode.

`-current1|current2`

If a module is also specified (-pm or -spm), the dialog automatically fetches its current bundle/viewfile name for Configuration 1 and Configuration 2 pane respectively. This is the same as selecting the Current view radio button.

`-committed1|committed2`

If a module is also specified (-pm or -spm), the dialog automatically fetches its committed bundle/viewfile name for the Configuration 1 and Configuration 2 pane respectively. This is the same as selecting the Committed view radio button.

`-log <log file name>`

An option passed on directly to nrsdiff to log its operations to file.

`-component <component name>`

An option passed on directly to nrsdiff to specify the root component type from which to compare the configurations.

`-keepfiles <directory name>`

An option passed on directly to nrsdiff to keep the working files for further use.

`-quiet`

An option passed on directly to nrsdiff to make it operate with minimal activity tracking output.

`-h`

Invokes a help panel for the tool (standard error stream).

Example

```
xnrsdiff -pm PM/R34
```

The command line in this example invokes a dialog to select two sets of configuration data for PM R34 for which to determine the differences. The output is presented on the standard output stream.

```
xnrsdiff -pm PM/R34 -x -stay
```

The command line in this example invokes a dialog to select two sets of configuration data for PM R34 for which to determine the differences. The output is presented in an xmsg window. Similarly, you could use -o to send the output to a file. With -stay, the dialog remains after launching the report so another selection can be made.

```
xnrsdiff -fi1 /opt/MagellanNMS/data/nrs/data/
dpn.R34.4034.94011217.940112data -fi2 /opt/Magellan/
data/nrs/data/dpn.R34.4034.94011903.940119.data
-component ITI -quiet -log
```

Note: The command must be entered as one continuous command.

This command line invokes the same dialog but prefills the Configuration 1 and Configuration 2 file areas with the specified arguments. The -component, -quiet and -log(as well as -keepfiles if specified) are passed on to nrsdiff as is.

NRS Data (xnrsrpt)

The xnrsrpttool provides the NRS Data dialog. The xnrsrpt tool is found in the directory /opt/MagellanNMS/bin.

The NRS Data dialog lets you select the source data on the basis of components, dates, current, committed, and name patterns for the NRS reporting program/script. It also gives you the option to directly select the source files and to populate the database as you go.

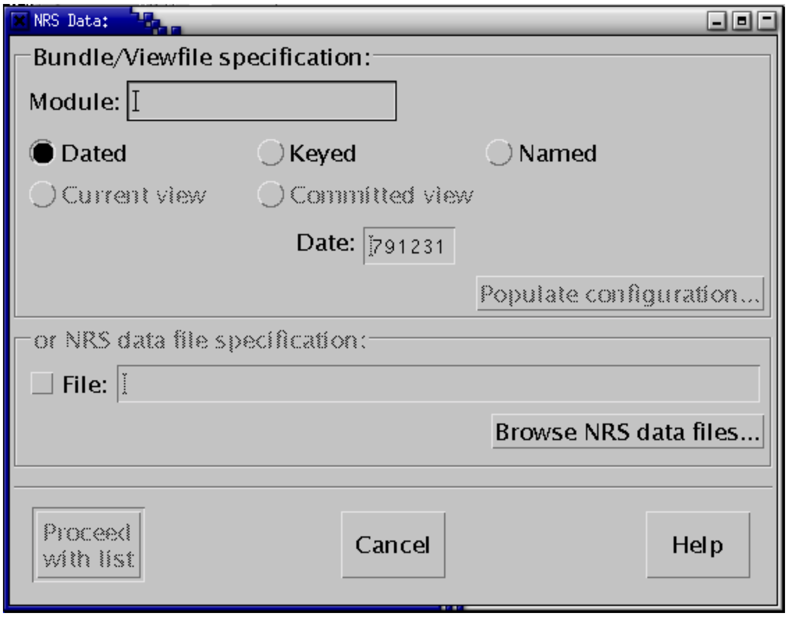
The NRS Data dialog is launched by entering the xnrsrpt command in a UNIX access window. For a description of the command line, see “Command syntax (xnrsrpt)” (page 224).

For additional information on NRS Data, see the following sections:

- “NRS Data dialog” (page 221)
- “Using the NRS Data dialog” (page 223)
- “Command syntax (xnrsrpt)” (page 224)

The figure “NRS Data dialog” (page 221) shows the NRS Data dialog.

Figure 15
NRS Data dialog



NRS Data dialog

The NRS Data dialog consists of three areas. For additional information on each area of the NRS Data dialog, see the following sections:

- “Bundle/Viewfile specification” (page 221)
- “NRS data file specification” (page 222)
- “Command buttons” (page 223)

  Note: The two file selection areas, Bundle/Viewfile specification and NRS data file specification, are mutually exclusive.

Bundle/Viewfile specification

The Bundle/Viewfile specification lets you specify the module and the patterns for the specified module. In the Bundle/Viewfile specification you enter the module name in Module field. You then click on one of the following radio buttons to select the pattern:

You can only specify one pattern at any one time. This reduces the number of files selected. The entry field corresponds to the selected radio button.

The pattern options are the following:

- Dated is the date for dated bundles. The file with the activation date closest to (without exceeding) the specified date is selected. The date must be valid and in the format yymmdd.
- Keyed is the key for keyed bundles. You must enter the key in the Key entry field.
- Named is the bundle name. Enter the pattern in the Pattern text field.
- Current view is the current active configuration on the node. The tool tries to communicate with the named node to extract the name of the current bundle or viewfile. If a group or OA is not available, a Connection Dialog opens, asking you to connect and identify such a route. The selected viewfile name appears in the Current viewfile entry field.

  The component type is derived from the module name:

  EM/* --> indicates Passport
  PM/* --> indicates a DPN-100
  By default, a single node name without a type prefix is considered a Passport.

- Committed view is the committed configuration on the node. As for Current view, the tool tries to communicate with the named node to extract the name of the current bundle or viewfile.
- The Populate configuration... button lets you ask for the immediate population of the configuration matching the specified bundle parameters. For the procedure to populate the database, see “Populating the NRS database” (page 232).

NRS data file specification

You can specify the NRS data file or files that you want to use by either entering the NRS data file name(s) in the file text field or by selecting a file from the data file selection dialog. The NRS data file specification area contains the following:

- The File toggle button enables the NRS data file specification.

- The File text field lets you directly enter one or more NRS data files to be used. The entry can be specified in GLOB style. You are not required to specify the full path name. If you do not specify the path name, the default is /opt/MagellanNMS/data/nrs/data.
- The Browse NRS data files... button invokes the NRS data file selection dialog to help you select a file. The file that you select appears in the File text field. For and description of the file selection dialog, and how to use it, see “Using the NRS Data File Selection dialog” (page 230).

Command buttons

The command buttons area provides the following commands:

- Proceed with list invokes the reporting program, passing to it the matching data file specification. If a direct file specification, using GLOB style, matches more than one file, the reports for the matching file specifications are concatenated. The report output appears on standard output, a file, or in a separate xmsg window as specified in the command line.

  If no matching file can be found, the tool will present a dialog asking you to populate the NRS database with the matching file. Selecting OK opens the Authentication Console dialog in the same manner as if you had selected the Populate configuration data... command.

- Close closes the dialog and ends the application without producing a report.
- Help invokes the on-line help for the NRS Data dialog

Using the NRS Data dialog

The following procedure describes how to use the NRS Data dialog.

1 From a UNIX access window, invoke the NRS Data dialog. See “NRS Data (xnrsrpt)” (page 188).

2 In the Module text field, enter a valid module name (Passport or DPN).

3 Select the NRS data file by either using the Bundle/Viewfile specification area or specifying a NRS data file in the NRS data file specification field:

In the Bundle/Viewfile specification area, select one of the following:

- Dated. The default date of 791231 appears in the Date text field unless you specify another date. See, “Date Convention” (page 27).

- Keyed. Enter the key in the Key text field.
- Named. Enter the pattern in the (Grep) Pattern text field.
- Current view. The Current Viewfile field is filled with the name of the active configuration.
- Committed view. The Committed Viewfile field is filled wih the name of the committed configuration.

Note: If you select Current or Committed view, and no group or OA connection is available, a Connection Console dialog opens prompting you to connect and identify a route. Connect and select the appropriate OA.

In the NRS data file specification pane, click the File toggle button and either type the NRS data file in the File text field or click Browse NRS data files... and select the desired NRS data file. See “Selecting NRS data files using the NRS file selection dialog” (page 228) for a description of the data file selection dialog and how to use it.

4 Click Proceed with list.

The report is output to standard output by default, or alternatively to a specified file or an xmsg window as specified in the command line.

If no matching NRS data file(s) exist for the specified module, a dialog opens saying no matching files were found to report on and asking you if you want to populate them from the device. Click OK. Clicking OK invokes the Authentication Console dialog. Perform the Authentication procedure. For the procedure to populate the configuration, see “Populating the NRS database” (page 232).

Alternately, you can populate the database before clicking Proceed with list, by using the Populate configuration... command. Clicking the Populate configuration button invokes the Authentication Console dialog. Perform the Authentication procedure.

Command syntax (xnrsrpt)

The xnrsrpt tool’s command line has the following options:

```
xnrsrpt
  [-ask|noask]
  [-stay]
  [-nofile]
  [-o <output file> | -x]
  [-title <report title>]
  [-pm|-spm <target module>]
  [  -file <file>
   | -dated <date>
   | -keyed <key>
   | -named <name> [-unique]
   | -current
   | -committed]
  [-report <reporter program> <additional arguments...>]
  [-h]
```

with supported abbreviations:

```
current:cu
committed:com
dated:da
file:fi
keyed:ke
named:na
report:re
unique:un
```

where:

`-ask|noask`

Option to specify if the dialog will show. If you specify -noask, the dialog will not show and the data selection is made exclusively from these command line arguments. The default is -ask.

`-stay`

Option to specify that the selection dialog stays once the report is invoked. If not specified, the selection dialog drops and xnrsrpt terminates. Specifying -stay lets you launch multiple reports.

`-nofile`

Option to prevent the explicit selection of NRS data file by hiding the corresponding dialog area. If this option is selected, you can only select the source data by bundle/viewfile specification.

```
-o <output file>|-x
```

Option to specify where the report output is sent. If -o is specified with an output file path, the report's output is sent to the named file. If -x is used, the report's output is sent to an independent xmsg window. If neither -o or -x are specified, the report's output is sent to the standard output stream.

```
-title <report title>
```

Option to override the default window title to the one specified. The default title indicates the name of the specified reporter program
(-report).

```
-pm|-spm <component ID>
```

Option used to specify a default value of the module name. A full component ID in either Display or API canonical format can be specified. If -spm is specified instead of -pm, the user will not be permitted to change the module name in the dialog.

```
-file <file name>
```

Option to specify a default NRS data file name or pattern to the dialog, which then adopts the corresponding mode.

```
-dated <date>
```

Option to specify a default NRS data file name/pattern to the dialog, which then adopts the bundle/viewfile mode.

```
-keyed <key>
```

Option to specify a default keyed pattern to the dialog which then adopts the bundle/viewfile mode.

```
-named <name>
```

Option to specify a default named pattern to the dialog which then adopts the bundle/viewfile mode.

`-current|committed`

If a module is specified (-pm or -spm) the dialog automatically fetches its current/committed bundle/viewfile name. This option acts in the same manner as though the corresponding radio button had been selected.

`-unique`

If specified, it is passed to the nrsrpt utility to provide unique file mappings from the specified bundle/viewfile patterns.

-report <reporter path> <arguments...>

This should be the last argument which identifies the reporting program / script to invoke with the selected data as input and its arguments. See the description of nrsrpt in “NRS Reporter (nrsrpt)” (page 284)

-*h*

Invokes a help panel for the tool (standard error stream).

> ***Note:*** The first two examples assume that the report script has been previously created by the following command sequence:

Example

```
cd /opt/MagellanNMS/data/nrs/rpt
nrscomp /opt/MagellanNMS/lib/nrs/rpt/nrsx25sh.rpt
x25short
```

This script is designed to produce an x25 report for a DPN-100.

```
xnrsrpt -report /opt/MagellanNMS/data/nrs/rpt/
x25short
```

This command line invokes the NRS selection dialog to execute the x25 short report against the selected configuration(s). The output is sent to standard output. If a report (-report and following) is not specified, the dialog will invoke nrslist to list the matching NRS data files.

```
xnrsrpt -x -report /opt/MagellanNMS/data/nrs/rpt/
x25short
```

This command line invokes the NRS selection dialog to execute the x25 short report against the selected configuration(s). The output is sent to xmsg window. If a report (-report and following) is not specified, the dialog will invoke nrslist to list the matching NRS data files.

Similarly, if you use -o option, the output is sent to the specified output file.

```
xnrsrpt -stay -x -pm EM/TOTO -keyed CORE -report \
/opt/MagellanNMS/bin/nrsdatah -fields
```

This command line invokes the NRS selection dialog to execute a configuration hierarchy report (with fields). The dialog is pre-filled with a target module name and viewfile key. In addition, the dialog will stay up allowing further selection and reporting cycles.

Selecting NRS data files using the NRS file selection dialog

This section describes the data file selection dialog that you can use to locate and identify an explicit NRS database file.

The NRS Data, Configuration Report and Configuration Differences dialogs all have an NRS data file specification panel. This panel gives you the option to directly enter the NRS data files that you want to report on, or to select the NRS data files from a NRS data file selection dialog. The file selection dialog lists the available NRS data files for a device.

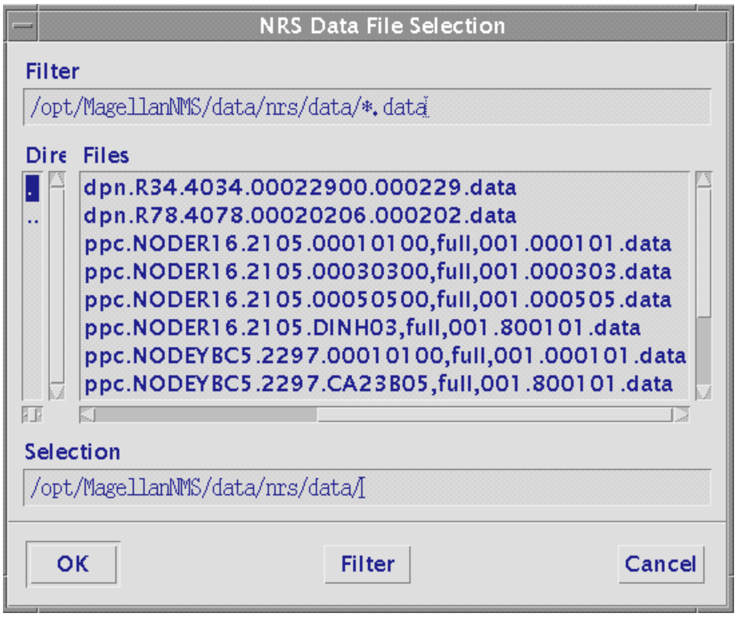
The file selection dialog can be invoked by clicking Browse NRS data files... in the NRS Data report, the Configuration Report and the Configuration Differences report.

See the following sections for additional information:

- “NRS Data File Selection dialog” (page 229)
- “Using the NRS Data File Selection dialog” (page 230)

For an example of the data file selection dialog, see the figure “NRS Data File Selection dialog” (page 229).

Figure 16
NRS Data File Selection dialog


NRS Data File Selection
Filter
/opt/MagellanNMS/data/nrs/data/*.data
Dire
Files
.
..
dpn.R34.4034.00022900.000229.data
dpn.R78.4078.00020206.000202.data
ppc.NODER16.2105.00010100,full,001.000101.data
ppc.NODER16.2105.00030300,full,001.000303.data
ppc.NODER16.2105.00050500,full,001.000505.data
ppc.NODER16.2105.DINH03,full,001.800101.data
ppc.NODEYBC5.2297.00010100,full,001.000101.data
ppc.NODEYBC5.2297.CA23B05,full,001.800101.data
Selection
/opt/MagellanNMS/data/nrs/data/
OK
Filter
Cancel

NRS Data File Selection dialog

The file selection dialog contains the following areas:

- “Filter text field” (page 230)
- “Directory scroll window” (page 230)
- “Files scroll window” (page 230)
- “Selection field” (page 230)
- “Command buttons” (page 230)

Filter text field

The Filter text field lets you specify filters to limit the number of NRS data files listed in the Files area. For example, to view only the NRS data files for a specific module, you can specify the module name as a pattern in the Filter text field and click the Filter command.

Example

If your /opt/MagellanNMS/data/nrs/data directory contains ppc, dpn and ppe modules and you only want to see all the NRS data files for ppc modules, you can enter the following filter in the Filter text field:

```
/opt/MagelanNMS/data/nrs/data/ppc*.data
```

The Files area of the file selection dialog only lists ppc modules.

Directory scroll window

The Directory scroll area lists the available data directories.

Files scroll window

The Files scroll window lists the available files for the selected directory.

Selection field

The Selection field specifies the full pathname for a selected file.

Command buttons

The file selection dialog has the following command buttons:

- OK closes the file selection dialog. The selected file appears in the file text field of the NRS data file specification area of the invoking dialog window.
- Clicking Filter activates the filter that you have specified in the Filter text field
- Cancel closes the file selection dialog.

Using the NRS Data File Selection dialog

See the following procedures for using the NRS data file selection dialog:

- “Locating and specifying a NRS database file” (page 231)
- “Creating filters” (page 231)

Locating and specifying a NRS database file

The following describes the procedure to locate and select an NRS database file using the Browse NRS data files... command in the Configuration Report, Configuration Differences, or NRS Data dialog:

1 Invoke one of the above NRS report dialogs.

2 Click Browse NRS data files...

The file selection dialog opens. The default path name is specified in NRS_DATA_DIR in /opt MagellanNMS/cfg/NRS.cfg, and points to /opt/ MagellanNMS/data/nrs/data. A list of files contained in the directory appears in the Files area of the dialog.

3 From the Files list, click on the NRS data file that you want to use.

The name of the selected file is transferred to the Selection field of the dialog.

4 Click OK.

The selected file appears in the File field in the NRS data file specification area of the main dialog and the file selection dialog disappears.

Note: You can create a filter to view only the NRS data files for a specific module. See the procedure “Creating filters” (page 231)

Creating filters

The data file selection dialog gives you the ability to reduce the number of files that are displayed in the Files window making it easier to find and select the appropriate file.

To use the filter option, follow this procedure:

1 In the Filter text field of the file selection dialog, specify the pattern for the filter.

2 Click Filter

The Files window of the dialog displays a list of files that correspond to the filter pattern.

Example

If your /opt/MagellanNMS/data/nrs/data directory contains ppc, dpn and ppe modules and you only want to see all the NRS data files for ppc modules, you can enter the following filter in the Filter text field:

```
/opt/MagelanNMS/data/nrs/data/ppc*.data
```

The Files area of the file selection dialog only lists ppc modules.

Populating the NRS database

The NRS database can be dynamically populated from the NRS Data, Configuration Report and Configuration Differences dialogs.

The following describes the procedure for populating the database using the Populate configuration command:

1 Launch the desired dialog and enter the desired Module or Component name.

2 Click Populate configuration...

The Authentication Console opens, asking you for a valid Group or OA, UserID and Password to access the device named in the Module field.

Note: The Authentication Console dialog opens in OA or Passport Group mode, depending on the type of the specified device.

3 Complete the Authentication Console dialog and click Authenticate.

The database is populated by the nrspop utility for DPN and the pnrspop utility for Passport. The output of the population process appears in a separate xmsg window.

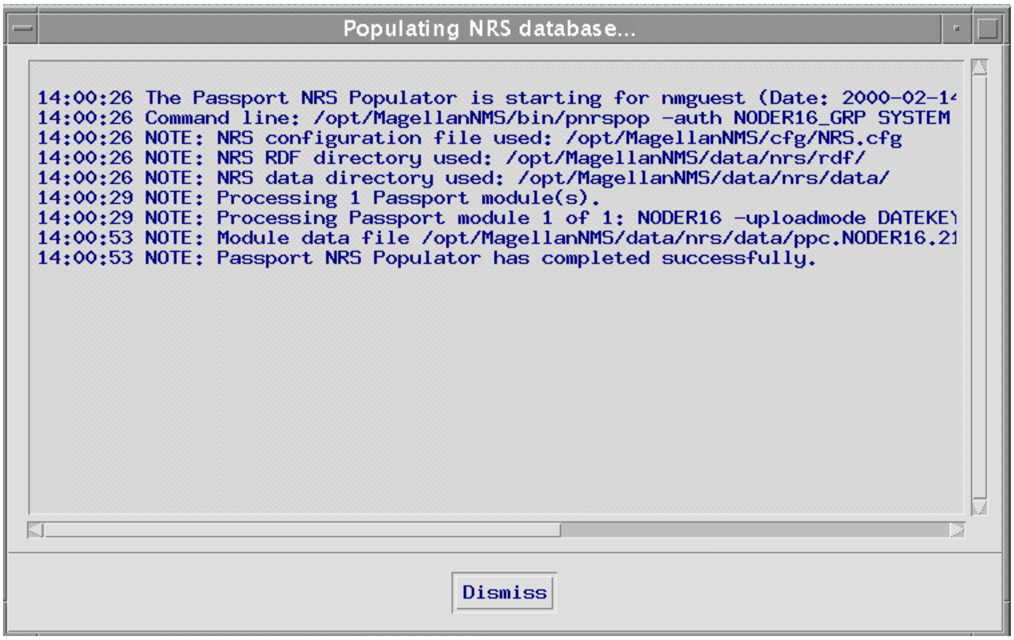
4 Once the population is complete, Click Dismiss.

The xmsg window closes.

For additional information on the NRS database population, see “DPN NRS Populator tool” (page 39) and “Passport NRS Populator” (page 42).

See the figure “Populate Configuration output window” (page 233) for an example of the output of the Populate configuration command.

Figure 17
Populate Configuration output window



```
14:00:26 The Passport NRS Populator is starting for nmguest (Date: 2000-02-1
14:00:26 Command line: /opt/MagellanNMS/bin/pnrspop -auth NODER16_GRP SYSTEM
14:00:26 NOTE: NRS configuration file used: /opt/MagellanNMS/cfg/NRS.cfg
14:00:26 NOTE: NRS RDF directory used: /opt/MagellanNMS/data/nrs/rdf/
14:00:26 NOTE: NRS data directory used: /opt/MagellanNMS/data/nrs/data/
14:00:29 NOTE: Processing 1 Passport module(s).
14:00:29 NOTE: Processing Passport module 1 of 1: NODER16 -uploadmode DATEKE
14:00:53 NOTE: Module data file /opt/MagellanNMS/data/nrs/data/ppc.NODER16.2
14:00:53 NOTE: Passport NRS Populator has completed successfully.
```

Chapter 15
NRS toolset reference

This chapter explains how to use each of the tools in the Network Reporting System toolset.

For more information on using the Population tools, see:

- “DPN NRS Populator (nrspop)” (page 236)
- “Passport NRS Populator (pnrspop)” (page 240)
- “Service Integrity Audit” (page 246)
- “DPN NRS Automatic Populator (nsrauto)” (page 252)
- “DPN NRS/NCD Population Manager (popmgr)” (page 256)
- “DPN NRS PM Lister (nrspmlst)” (page 258)

For more information on using the Service Integrity tools, see

- “Service Integrity Audit” (page 246)
- “NRS-based Service Integrity Checks (NSIC)” (page 262)

For more information on using the report creation tools, see:

- “NRS Hierarchical Report Generator (nrshier)” (page 269)
- “NRS RDF Hierarchy Generator (nrsrdfh)” (page 272)
- “NRS RDF Owners Hierarchy Generator (nrsowner)” (page 276)
- “NRS Preprocessor (nrscomp)” (page 280)
- “NRS Component Joiner (nrsjoin)” (page 281)

- “NRS Reporter (nrsrpt)” (page 284)

For more information on using the maintenance tools, see:

- “NRS Module File Deletor (nrsrm)” (page 287)
- “NRS Sanity Checker (nrsanity)” (page 290)

For more information on using the utility tools, see:

- “NRS Module Filename Lister (nrslist)” (page 291)
- “NRS Validator (nrsvalid)” (page 298)

For more information on using the report programs, see:

- “NRS Differences Report (nrsdiff)” (page 299)
- “NRS Data Hierarchy Report (nrsdatah)” (page 305)
- “NRS Find DNA (nrsfinddna)” (page 308)
- “NRS Display Service (nrsdserv)” (page 310)

DPN NRS Populator (nrspop)

The DPN NRS Populator (nrspop) tool uploads module service data from either the DPN-100 network, the backup disk, or the NMS disk, then reformats the data and stores it in the NRS database. The DPN NRS Populator is called nrspop and is found in the directory /opt/MagellanNMS/bin. The DPN NRS Populator uses the NRS_DATA_DIR and the NRS_RDF_DIR symbols from the NRS configuration file NRS.cfg.

This section outlines both the DPN NRS Populator command line syntax and the syntax of the command file which lists the modules to be uploaded. Error messages are written to stderr and information messages are written to stdout.

For return codes, see the table “DPN NRS Populator return code description” (page 238).

Command syntax

Enter the following command syntax as one continuous command.

```
nrspop -f <command file name>
  -ncs <destination mnemonic> <capability id>
  <password> | -manual [-log [<log file name>]]
  [-quiet] [-h]
```

where:

`-f <command file name>`

Option to use command file. Command file containing a list of modules for which the populator is to get data. Comment and blank lines are permitted, but comment lines must start with a # character.

`-ncs <destination mnemonic> <capability id> <password>`

Option for the NCS authentication

`destination mnemonic` NCS mnemonic of the OA interface.
`capability id` NCS capability id.
`password` NCS capability id password.

`-manual`

Indicates manual NCS access mode. Authentication with NCS is bypassed if this option is specified.

`-log [<log file name>]`

Option to have populator messages recorded in a logfile. The default file name is nrspop.log unless otherwise specified.

`-quiet`

Option to suppress the generation of information messages.

`-h`

Option to display command line usage information.

Example

```
nrspop -f /localdisk/user/nrspop.requests -ncs mdi id
pw
nrspop -f myfile -ncs mdi id pw -log
/localdisk/user/my.log
nrspop -ncs mdi id pw -f myfile -log -quiet
```

Table 6
DPN NRS Populator return code description

Return code	Description
0	Normal success: all modules successfully uploaded or already in NRS database.
1	Validation failure: command line or module list file validation error.
2	Provisioning failure: Provisioning error or communication problem. Some modules may have been successfully uploaded prior to the error condition.
3	Processing failure: NRS Populator encountered an error condition during the reformatting of the data for a module and exited. Some modules may have been successfully uploaded prior to the error condition.
4	User requested termination. A termination signal (Control-C) was received by the tool.
11, 12, . . .	Partial success: The NRS Populator completed, but some module(s) were not uploaded. The number of module uploads which failed are <return code> -10. For example, If the return code was 12, then 2 module upload failures occurred.

Command file syntax

The following is the syntax for lines in the command file, specifying modules to be uploaded. Comment and blank lines are allowed but a comment line must start with a #

character.

```
<pm> -uploadmode <upload_mode>
  [<upload_bundle_or_key_or_date> [upload_namsid]]
  [-uploadlocation <location_mode>
  [<location_value>]]
```

where:

`pm`

The PM the service data belongs to.

`-uploadmode <upload_mode> [<upload_bundle_or_key_or_date> [upload_namsid]`

Option to indicate the mode to be used for uploading.

`<upload_mode>` Indicates the mode of retrieval for the service data. Must be one of: COMMITTED, ACTIVE, USER_SPECIFIED, KEYED, DATED. Modes are not case sensitive and only the first character is required.
<upload_bundle_or_key_or_date> Indicates bundle/key/date for upload. Only valid if upload_mode is USER_SPECIFIED, KEYED or DATED. A date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.
<upload_namsid> Indicates NAMS ID for upload. Not valid if upload_mode is ACTIVE or COMMITTED.

`-uploadlocation <location_mode> [<location_value>]`

Option to indicate the location of the MCF to upload.

<location_mode> Indicates from where the service data is to be retrieved. Must be one of: USER_SPECIFIED, NMS_DISK, BACKUP_DISK. This field is not case sensitive and only the first character is required.
<location_value> Represents the PM where the MCF is stored. Used only with location_mode USER_SPECIFIED.

The following are conditions that must be met when certain command file parameters are chosen:

- If upload_mode is COMMITTED or ACTIVE then location_mode cannot be BACKUP_DISK or NMS_DISK.
- The upload_bundle_or_key_or_date is treated as a bundle, key or date depending on the upload_mode. If upload_mode is USER_SPECIFIED then upload_bundle_or_key_or_date will be treated as a bundle. It will be treated as a key when the upload_mode is KEYED and as a date when the upload_mode is DATED. If the upload_mode is COMMITTED or ACTIVE then this optional parameter cannot be present.
- For upload mode USER_SPECIFIED, KEYED or DATED, the NAMS ID must be specified when the upload location is not USER_SPECIFIED.
- When the location_value is not used the <pm> value is used. The default upload location is USER_SPECIFIED <pm>.

Example

The following is an example of a command file.

```
# This is a sample nrspop command file
R34 -uploadmode dated 990317 4034
R34 -uploadmode K 99WK25 4034 -uploadlocation NMS_DISK
R35 -uploadmode committed
R36 -uploadmode ACT -uploadlocation User_specified R40
R37 -uploadmode US ABCD -uploadlocation BACKUP_DISK
```

Passport NRS Populator (pnrspop)

The Passport NRS Populator (pnrspop) tool uploads module service data from the Passport modules, then reformats the data and stores it in the NRS database. The Passport NRS Populator is called pnrspop and is found in the directory /opt/MagellanNMS/bin. The Passport NRS Populator uses the NRS_DATA_DIR and the NRS_RDF_DIR symbols from the NRS configuration file NRS.cfg.

> ***Note:*** Ensure that the capability/scope for the userid used to log onto the Passports is the same across all populations. Different capability privileges opens access to the information stored in the NRS database to everyone. It can also result in inconsistencies in the NRS database. For instance, when you run the Configuration Differences report tool, the results for files which have been populated with configuration privileges are very different from files which have been populated with administration privileges.

This section outlines both the Passport NRS Populator command line syntax and the syntax of the command file which lists the modules to be uploaded. Error messages are written to stderr and information messages are written to stdout.

For return codes, see the table “Passport NRS Populator return code description” (page 244).

Command syntax

Enter the following command syntax as one continuous command.

```
pnrspop -auth <group name> <capability id> <password>
  -uploadmode <mode> |<view filename>]<key>|<date>|
  [-log [<log file name>]] [-activation_date <date>]
  [-pm <pm> ...<pm>] [-f <command file name>] [-quiet]
  [-h]
```

where:

`-auth <group name> <capability id> <password>`

Option to use Passport group authentication.

`<group name>` The mnemonic of the Passport group to log on to.
`<capability id>` The capability id to log on with.
`<password>` Capability id password.

`-uploadmode <mode> |<view filename>]<key>|<date>|`

Option to indicate the upload mode to be used. If *-pm* is not specified, the upload mode is applied for all Passport modules belonging to the authenticated group.

`<mode>` Indicates the mode of retrieval for the provisioning data. Must be one of:
USER_SPECIFIED (U) - The <view filename> is treated as a complete provisioning file name.
KEYED (K) - The <key> can be composed of letters, digits, and underscores. The key cannot begin with an underscore and must be no more than six characters in length.
DATED (D) - Valid <date> in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.
COMMITTED (COM)/CURRENT (CUR) - There are no parameters associated with these modes.

> ***Note:*** If CURRENT is specified, *pnrspop* populates the current view data without opening a Passport provisioning session. This allows other applications to update the Passport data while this Passport is currently being populated by *pnrspop*. For all other update modes, *pnrspop* starts a Passport provisioning session to load the specified provisioning file.

<key>|<date>|<view filename> Indicates the view file/key/date for upload. Only valid if upload_mode is USER_SPECIFIED, KEYED, or DATED.

-log [<log file name>]

Option to have messages recorded in a logfile. The default file name is *pnrspop.log* unless otherwise specified.

-activation_date <date>

Option to use the date for naming the module data file. The date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format. If the NMS dated option is used, this option is ignored. If this option is not specified and dated mode is not used, the default date is 800101 (January 1, 1980).

-pm <pm>...<pm>

Option to indicate Passport module(s) to be accessed. More than one module can be specified with this option.

-f <command file name>

Option to use command file. Command file containing a list of modules for which the populator is to get data. Comment and blank lines are permitted, but comment lines must start with a # character.

-quiet

Option to suppress the generation of information messages.

-h

Option to display command line usage information.

Example

The following are examples of command lines used to obtain Passport data.

Upload data for all Passport modules belonging to the authenticated group:

```
pnrspop -auth <group name> <capability id> <password>
   -uploadmode <mode> |<view filename>]<key>|<date>|
   [-log [<log file name>]]
   [-activation_date <date>]
```

Upload data for Passport module(s) specified in command line:

```
pnrspop -auth <group name> <capability id> <password>
   -pm <pm> ...<pm>
   -uploadmode <mode> |<view filename>]<key>|<date>|
   [-log [<log file name>]]
   [-activation_date <date>]
```

Upload data for Passport modules specified in the command file:

```
pnrspop -auth <group name> <capability id> <password>
   -f <command file name>
   [-log [<log file name>]]
   [-activation_date <date>]
```

Upload data for Passport modules specified in both the command line and command file:

```
pnrspop -auth <group name> <capability id> <password>
   -pm <pm> ...<pm>
   -uploadmode <mode> |<view filename>]<key>|<date>|
   -f <command file name>
   [-log [<log file name>]]
   [-activation_date <date>]
```

Table 7
Passport NRS Populator return code description

Return code	Description
0	Normal success: all modules successfully uploaded or already in NRS database.
1	Validation failure: command line or module list file validation error.
2	Provisioning failure: Provisioning error or communication problem. Some modules may have been successfully uploaded prior to the error condition.
3	Processing failure: NRS Populator encountered an error condition during the reformatting of the data for a module and exited. Some modules may have been successfully uploaded prior to the error condition.
4	User requested termination. A termination signal (Control-C) was received by the tool.
11, 12, . . .	Partial success: The NRS Populator completed, but some module(s) were not uploaded. The number of module uploads which failed are <return code> -10. For example, If the return code was 12, then 2 module upload failures occurred.

Command file syntax

The following is the syntax for lines in the command file, specifying modules to be uploaded. Comment and blank lines are allowed but a comment line must start with a # character.

```
<pm> -uploadmode <mode> [<view_filename>|<key>|<date>]
```

where:

```
<pm>
```

The PM the service data belongs to.

```
-uploadmode <mode> [<view_filename>|<key>|<date>]
```

Mandatory option to indicate the mode to be used for uploading.

`<mode>` Indicates the mode of retrieval for the provisioning data. Must be one of:
USER_SPECIFIED (U) - The <view filename> is treated as a complete provisioning file name.
KEYED (K) - The <key> can be composed of letters, digits, and underscores. The key cannot begin with an underscore and must be no more than six characters in length.
DATED (D) - Valid <date> in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.
COMMITTED (COM)/CURRENT (CUR) - There are no parameters associated with these modes.
<view_filename>|<key>|<date> Indicates the view file/key/date for upload. Only valid if upload_mode is USER_SPECIFIED, KEYED, or DATED.

Example

```
#This is a sample pnrspop command file
NODER1 -uploadmode CURRENT
NODER2 -uploadmode K MYKEY
NODER3 -uploadmode COMMITTED
```

Service Integrity Audit

The Service Integrity Audit tool is used to populate the NRS database for DPN and/or Passport modules, and, optionally, to execute the NRS-based Service Integrity checks (NSICs), and to populate the Network Configuration Database (NCD). With this tool, the MCFs/View Files are retrieved directly from the modules.

When the Service Integrity Audit tool is executed, the tool will determine for DPN modules the valid MCF for each PM and populate the MCF into the NRS database. Similarly for Passport modules, the tool will determine the valid View File for each module and populate it into the NRS database. It then performs the NSIC checks and populates the NCD. When the Service Integrity Audit tool is executed, an NRS database population is always performed, but NSIC checks and NCD population are optional and can be omitted.

For return codes, see the table “Sisauto return codes” (page 251).

Command syntax

The following is the command syntax for the sisauto tool. This command can be executed from a UNIX command line or from a cron job.

Enter the following command syntax as one continuous command.

For DPN and Passport:

```
sisauto -ncs <destination mnemonic>
<capability id> <password>
   -auth <group name> <user id> <password>
   [-f_pm <file of PMs> | -pm <pm> | -restart_pm]
   [-nmsdated_pm <NMS MCF date>]
   [-f_em <file of EMs> | -em <em> | -restart_em]
   [-nmsdated_em <NMS view file date>]
   [-nrsdate <NRS selection date>] [-nonsic]
   [-noncdpop] [-quiet] [-h]
```

For DPN only:

```
sisauto -ncs <destination mnemonic> <capability id>
<password>
   [-f_pm <file of PMs> | -pm <pm> | -restart_pm]
```

```
      [-nmsdated_pm <NMS MCF date>]
      [-nrsdate <NRS selection date>] [-nonsic]
      [-noncdpop] [-quiet] [-h]
```

For Passport only:

```
   sisauto -auth <group name> <user id> <password>
      [-f_em <file of EMs> | -em <em> | -restart_em]
      [-nmsdated_em <NMS view file date>]
      [-nrsdate <NRS selection date>] [-nonsic]
      [-noncdpop] [-quiet] [-h]
```

where:

`-ncs <destination mnemonic> <capability id> <password>`

Mandatory option for NCS authentication when listing the MCFs from the various PMs or performing NRS populations. This option is also used when the -f_pm, -pm, and -restart_pm options are not specified. In this case, all the PMs reporting to the NCS OA represented by the <destination mnemonic> parameter or reporting to any OA that exists in the hierarchy of the NCS OA are considered. This is referred to as all PMs mode.

Only OAs of type OA, OABU, CF and CFBU are supported with all PMs mode.

`<destination mnemonic>` NCS mnemonic of the MDI interface
<capability id> NCS capability id
<password> Password for the NCS capability id

> ***Note:*** The above parameters are not case sensitive.

`-auth <group name> <user id> <password>`

Option used for the authentication to Passport modules (EMs) when listing the view files and performing the NRS populations.

This option is also used to determine all the EMs under the specified group when the -f_em, -em and -restart_em options are not specified. This is referred to as the all EMs mode.

`<group name>` Pasport group name
`<user id>` User ID
`<password>` Password for User ID.
The <password> parameter is case sensitive; the <group name> and <user id> parameters are not.

`-f_pm <file of PMs>`

Option to specify a file containing a list of PMs. This is referred to as the PM file mode.The file must contain the PM mnemonics, one per line. These mnemonics are not case sensitive but the file name is.
File name example: /localdisk/admin/pm.list.

`-pm <PM>`

Option to specify a particular PM (for example, r34). This is referred to as the PM mode. The <PM> is a specific PM mnemonic.The <PM> parameter is not case sensitive.

-restart_pm

Option to specify that all the PMs where a problem occurred while determining the valid MCFs or populating NRS in the previous execution of the tool, are considered. This list also includes all the PMs that were not processed in the previous execution of the tool. This is referred to as PM restart mode. NSIC and/or NCD Population are run independently of this option.This list of PMs is kept internally.

`-nmsdated_pm <NMS MCF date>`

Option used to select the NMS MCF. The NMS dated MCF with a date closest to, but not exceeding, the specified date is selected. This option is mandatory only when the configuration variable SIS_MCF_DETERMINATION_MODE is set to 2, 3 or 4 and variable SIS_NMS_MCF_MODE is set to 3. This option cannot be specified if these variables are set to different variables.
The <NMS MCF date> is a valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-f_em <file of EMs>`

Option to specify a file containing a list of EMs. This is referred to as the EM file mode.The file must contain the EM mnemonics, one per line. These mnemonics are not case sensitive but the file name is.
File name example: /localdisk/admin/passport.list.

`-em <EM>`

Option to specify a specific EM (for example, node43). This is referred to as the EM mode. The <EM> parameter is not case sensitive.

`-restart_em`

Option to specify that all the EMs where a problem occurred while determining the valid View File, or populating NRS in the previous execution of the tool, are considered. This list also includes all the EMs that were not processed in the previous execution of the tool. This is referred to as EM restart mode. NSIC and/or NCD Population are run independently of this option. This list of EMs is kept internally.

`-nmsdated_em <NMS view file date>`

Option used to select the NMS view file. The NMS dated view file with a date closest to, but not exceeding, the specified date is selected. This option must be specified when the configuration variable SIS_VIEWFILE_DETERMINATION_MODE is set to 2 and variable SIS_NMS_VIEWFILE_MODE is set to 3. This option cannot be specified if these variables are set to different values.
The <NMS view file date> is a valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-nrsdate <NRS selection date>`

Option to specify a date to be used when selecting the NRS files when running NSIC and when populating NCD. The <NSR selection date> is a valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-nonsic`

Option to specify that the NSIC checks should not be executed.

`-noncdpop`

Option to specify that the NCD population should not be performed.

`-quiet`

Option to suppress the generation of information messages. Information messages will not be sent to standard error or a log file.

`-h`

Option to display command line usage information.

> ***Note:*** If the *-nonsic* and *-nrsdate* options are not specified, activation date 791231 (December 31, 2079), meaning "the latest NRS file available for each module", is used by default. If the *-noncdpop* and *-nrsdate* options are not specified, the date 791231 is also used to select the NRS files to be populated into NCD.

Example

```
/opt/MagellanNMS/bin/sisauto -ncs mdi id pw -auth group id
pwd

/opt/MagellanNMS/bin/sisautui
```

Table 8
Sisauto return codes

Return code	Description
0	Successful completion. All the MCF Lists, View File Lists, and NRS Populations were successful and NSIC did not find any errors or warnings. The NRS files were populated successfully to NCD.
1	Error. Some of the MCF Lists, View File Lists, or NRS Populations were not successful, or errors were produced by NSIC, or the NCD Population found some duplication, or some major error(s) occurred with the tool.
2	User requested termination. A termination signal was received by the tool, for example, Control-C. sisauto terminated properly and the -restart_pm or restart_em options can be used to restart the tool.
3	The tool is currently running elsewhere. This tool does not allow concurrent sessions to be running unless the SIS_WORK_DIR parameter (as defined in the Service Integrity Audit configuration file) is different for each session.
4	Warnings. All the MCF Lists, View File Lists, and NRS Populations were successful but warnings were produced by NSIC.

Table 9
sisautui return codes

Return code	Description
0	Successful completion. The sisauto tool was called. The success or failure of the sisauto tool is not reflected here.
1	Error. A validation error occurred while retrieving some of the parameters.
2	User requested termination. A termination signal, Control-C, was received by the tool. The tool has terminated.

DPN NRS Automatic Populator (nsrauto)

The DPN NRS Automatic Populator (nrsauto) can be used to automate the population of the NRS database with MCFs that fall into the following categories:

- Active - retrieve the active MCF.
- Committed - retrieve the committed MCF.
- Dated - retrieve the latest or all the NMS DATED MCFs with a date less than or equal to the specified date.
- File system dated - retrieve the latest or all the MCFs with a file system date less than or equal to the specified date.

The -ncs parameters are used for NCS authentication but they are also used to request a list of all the PMs in the NCS OA hierarchy of the specified destination mnemonic. The only OA types that are supported are OA, OABU, CF and CFBU.

The -pm option can be used to specify a particular PM. The -f option can be used to specify a list of PMs.

The nrsauto utility calls the nrspmlst tool to generate a command file for the MCFs that are selected. It then calls the *nrspop* tool with that command file to perform the actual population. These tools can be found in the directory /opt/MagellanNMS/bin.

If this tool is run from a cron job, it must be executed using the cmcwrap program in order to create the necessary session servers.

Example

The following is an example of the command used to execute *nrsauto* from a cron job.

```
/opt/MagellanNMS/bin/cmcwrap
/opt/MagellanNMS/bin/nrsauto <nrsauto options>
```

Note: The nrsauto utility requires NAMS, APPLICATION, PASSIVE and SWITCHING, DEVICE, PASSIVE NCS capabilities.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsauto -ncs <destination mnemonic> <capability id>
<password> [-pm <pm> | -f <file of PMs>]
-active | -committed | -dated <date> [-latest_dated] |
-file_sys_dated <date> [-latest_file_sys_dated]
[-log [<log file name>]] [-cmdfile <nrspop command file
name>] [-quiet] [-h]
```

where:

`-ncs <destination mnemonic> <capability id> <password>`

Option for the NCS authentication. Also used to select all the PMs in the NCS OA hierarchy of the specified destination mnemonic when the *-pm* and *-f* options are not specified.

`<destination mnemonic>` NCS mnemonic of the OA interface.
`<capability id>` NCS capability id.
`<password>` NCS capability id password.

`-pm <pm>`

Option for specifying a particular PM.

`-f <file of PMs>`

Option to specify a list of PMs. File name of a file containing a list of PMs. Comment and blank lines are permitted, but comment lines must start with a # character.

`-active`

Option to populate the active MCF.

`-committed`

Option to populate the committed MCF.

`-dated <date>`

Option to select the NMS DATED MCFs which do not exceed the specified date. The field is a valid date is in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-latest_dated`

Option to select only the latest MCF from the NMS DATED MCFs selected by the -*dated* option.

`-file_sys_dated <date>`

Option to select the MCFs that have a file system date which do not exceed the specified date. The field is a valid date is in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-latest_file_sys_dated`

Option to select only the latest MCF from the MCFs selected by the -*file_sys_dated* option.

`-log [<log file name>]`

Option to have the messages recorded in a log file. The default file name if *nrsauto.log* unless otherwise specified.

`-cmdfile <nrspop command file name>`

Option to save the NRS populator command file.

`iet`

Option to suppress the generation of information messages.

`-h`

Option to display command line usage.

Example

The following updates the NRS database with the active MCF of every PM in the NCS OA hierarchy of destination mnemonic MDI and records all the log messages in *nrsauto.log* (default).

```
nrsauto -ncs mdi id pw -active -log
```

The following updates the NRS database with the committed MCF of all the PMs specified in the *pm.list* file and suppress all informational messages.

```
nrsauto -ncs mdi id pw -committed -f pm.list -quiet
```

The following updates the NRS database with today’s NMS DATED MCF of every PM in the NCS OA hierarchy of destination mnemonic MDI:

```
nrsauto -ncs mdi id pw -dated `date ‘+%y%m%d’`
-latest_dated
```

The following updates the NRS database with all the MCFs of PM A20 and save the NRS populator command file in *PMA20.file*.

Note: A date of 791231 represents December 31, 2079.

```
nrsauto -ncs mdi id pw -file_sys_dated 791231
-pm A20 -cmdfile PMA20.file
```

For return codes, see the table “DPN NRS Automatic Populator return code description” (page 255).

Table 10
DPN NRS Automatic Populator return code description

Return code	Description
0	The tool completed successfully. The PMs that are down are not considered as errors.
1	Errors were found.
2	A termination signal (Control-C) was received.
11, 12, ...	Some MCFs were not uploaded. The number of MCFs that were not uploaded is <return code> minus 10. For example, if the return code is 12, then 2 MCFs were not uploaded.

DPN NRS/NCD Population Manager (popmgr)

The NRS/NCD Population Manager tool (popmgr) is used to simplify the population of the NRS and NCD databases. With this tool, the MCFs are retrieved from BACKUP_DISK or from NMS_DISK.

The popmgr tool periodically checks the BACKUP_DISK or NMS_DISK directory for the arrival of new MCFs. The NRS population tool then populates the MCF data into NRS database and automatically executes the NCD population tool to populate NRS data into the NCD database.

Provisioning changes will be populated automatically into the NRS and NCD databases. This reduces the latency between a download and the availability of data in NRS and NCD and allows NSIC to be run sooner after provisioning activities.

This tool does not support Passport modules. It will only populate the DPN NRS data into the given NCD database. If MCFs are manually populated into the NRS database or populated using the Service Integrity Audit tool, the population management tool will automatically detect the new NRS data and populate it into the NCD database if the tool is configured correctly.

For return codes, see the table “popmgr return codes” (page 258).

Command syntax

The following command syntax is used to start the popmgr tool. Enter the following command syntax as one continuous command. To display online help for this command use the -h option.

If the -dated or -keyed options are not specified on the command line, the latest MCF/NRS data per PM will be selected.

```
/opt/MagellanNMS/bin/popmgr -ncs <destination
mnemonic>
  <capability id> <password> | -manual
  [-dated <date> | -keyed <key>] [-log [<log file>]]
  [-restart] [-noncdpop] [-quiet] [-h]
```

where:

`-ncs <destination mnemonic> <capability id> <password>`

Mandatory option for NCS authentication when performing NRS populations.

`<destination mnemonic>` NCS mnemonic of the MDI interface
`<capability id>` NCS capability id
`<password>` NCS capability id password

`-manual`

Option to use without NCS authentication when performing NRS population. This option cannot be used with the *-ncs* option.

`-dated <date>`

Option used to select the MCF/NRS data. The MCF/NRS data with a bundle matching the specified date is selected. This option cannot be specified with the *-keyed* option.

`<date>` A valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

`-keyed <key>`

Option used to select the MCF/NRS data. The MCF/NRS data with the highest version of the specified key is selected. This option cannot be specified with the *-dated* option.

`-log [<log file>]`

Option to have all messages recorded in a log file. The log file will be updated under the POPMGR_WORK_DIR directory.

`<log file>` The log file is named *popmgr.log* unless otherwise specified.

`-restart`

Option to restart the *popmgr* tool from the last stop point. All requests will be reprocessed if a problem occurred while selecting MCF/NRS data, populating NRS or NCD in the previous execution of the tool.

`-noncdpop`

Option to specify that NCD population should not be executed. NCD population is executed by default.

`-quiet`

Option to suppress the generation of information messages.

`-h`

Option to display command line usage information.

Table 11
popmgr return codes

Return code	Description
0	The tool has finished normally without error.
1	The tool has finished normally but errors were found.
2	The user terminates the tool.
Note: *popmgr* usually runs all the time. The tool only exists in the following situations: 1. command line syntax error 2. terminated by user 3. computer reboot.	

DPN NRS PM Lister (nrspmlst)

The NRS PM Lister (nrspmlst) tool is used to generate a list of NRS population requests for the MCFs that fall in the following categories:

- Active - the active MCF.
- Committed - the committed MCF.
- Dated - the latest or all the NMS DATED MCFs with a date less than or equal to the specified date.
- File system dated - the latest or all the MCFs with a file system date less than or equal to the specified date.

The -ncs parameters are used for NCS authentication but they are also used to request a list of all the PMs in the NCS OA hierarchy of the specified destination mnemonic. The only OA types that are supported are OA, OABU, CF and CFBU.

The -pm option can be used to specify a particular PM. The *-f* option can be used to specify a list of PMs. This tool is located in the /opt/MagellanNMS/bin directory.

If this tool is run from a cron job, it must be executed using the cmcwrap program in order to create the necessary session servers.

Example

The following is an example of the command used to execute *nrspmlst* from a cron job.

```
/opt/MagellanNMS/bin/cmcwrap /opt/MagellanNMS/bin/
nrspmlst <nrspmlst options>
```

Note: The *nrspmlst* utility requires NAMS, APPLICATION, PASSIVE, SWITCHING, DEVICE, and PASSIVE NCS capabilities.

Command syntax

Enter the following command syntax as one continuous command.

```
nrspmlst -ncs <destination mnemonic> <capability id>
  <password> [-pm <PM> | -f <file of PMs>]
-active | -committed | -dated <date> [-latest_dated] |
-file_sys_dated <date> [-latest_file_sys_dated]
[-log [<log file name>]] [-quiet] [-h]
```

where:

`-ncs <destination mnemonic> <capability id> <password>`

Option for the NCS authentication. Also used to select all the PMs in the NCS OA hierarchy of the specified destination mnemonic when the *-pm* and *-f* options are not specified.

`<destination mnemonic>` NCS mnemonic of the OA interface.
`<capability id>` NCS capability id.
`<password>` NCS capability id password.

```
-pm <pm>
```

Option for specifying a specific PM.

```
-f <file of PMs>
```

Option to specify a list of PMs. File name of a file containing a list of PMs. Comment and blank lines are permitted, but comment lines must start with a # character.

```
-active
```

Option to select the active MCF.

```
-committed
```

Option to select the committed MCF.

```
-dated <date>
```

Option to select the NMS DATED MCFs that do not exceed the specified date. The date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

```
-latest_dated
```

Option to select only the latest MCF from the NMS DATED MCFs selected by the -*dated* option.

```
-file_sys_dated <date>
```

Option to select the MCFs that have a file system date which does not exceed the specified date. The date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

```
-latest_file_sys_dated
```

Option to select only the latest MCF from the MCFs selected by the -file_sys_dated option.

```
-log [<log file name>]
```

Option to have the messages recorded in a log file. The default file name is *nrspmlst.log* unless otherwise specified.

```
-quiet
```

Option to suppress the generation of information messages.

```
-h
```

Option to display command line usage information.

Example

```
nrspmlst -ncs mdi id pw -active
nrspmlst -ncs mdi id pw -f pm.file -dat 940510
   -latest_d
nrspmlst -ncs mdi id pw -pmR34 -file_s 940510 -quiet
   -log
nrspmlst -comm -log /localdisk/user/my.log
```

For return codes, see the table “DPN NRS PM Lister return code description” (page 261)

Table 12
DPN NRS PM Lister return code description

Return code	Description
0	The tool completed successfully. The PMs that are down are not considered as errors.
1	Errors were found.
2	A termination signal (Control-C) was received.

NRS-based Service Integrity Checks (NSIC)

NRS-based Service Integrity Check (NSIC) is an application used to detect inconsistencies in the service data which could cause service problems during network operation. These inconsistencies are not detected during a normal provisioning session because data from several or even all modules in the network may be required to determine the nature of the problem.

NSIC is a diagnostic tool which does not change or effect service data in any way. It is a passive tool which is run after the service data has been generated. It provides a means of identifying inconsistencies in service data prior to activation and reduce the possibility of service disruption.

If no feature options are specified then all checks will be performed. This is the recommended way to run NSIC. The time spent by NSIC to perform all checks in a single run, is less than the total time spent to perform different checks in multiple runs.

For return codes, see the table “NSIC return code description” (page 268).

Command syntax

Enter the following command syntax as one continuous command.

```
/opt/MagellanNMS/bin/nsic [<nrs_selector> |
<nrs_list>]
   [<check_options>]
   [<other_options>]
   [-h]
```

where:

```
<nrs_selector>  = [-type <module_type>]
           [-pm <pm>]
           [-namsid <namsid>]
           [-bundle <bundle> | -viewfile <viewfile>]
           [-activation_date <activation_date>]
           [-dated <date> | -keyed <key>]

<nrs_list>      = -nrs_list <file>

<check_options> = [-dna_uniqueness] [-gid_uniqueness]
```

```
           [-mid_uniqueness] [-module_name_uniqueness]
           [-nodeid_uniqueness] [-Namsid_uniqueness]
           [-oa_uniqueness] [-vxid_uniqueness]
           [-ipaddr_uniqueness] [-redirection]
           [-hunt_group] [-pvc] [-direct_call] [-rid_map]
           [-call_forward] [-accounting] [-voice]
           [-external_dna_file <dna_file>]
           [-external_rid_file <rid_file>]
           [-external_ipaddr_file <ipaddr_file>] [-atm]

<other_options> = [-noconfirm]
           [-sorted <sorted output filename>]
           [-debug]
```

-type <module_type>

Option for identifying the module type parameter.

<module_type> Character string expression for matching with the module type in the module data filenames.

-pm <pm>

Option for identifying the PM parameter.

<pm> Character string expression for matching with the PM mnemonic in the module data filenames.

-namsid <namsid>

Option for specifying the NAMS ID parameter.

<namsid> Character string expression for matching with the NAMS ID in the module data filenames.

-viewfile <viewfile>

Option to specify the provisioning file name.

<viewfile> Character string expression used to match the name of the Passport view in the module data files.

-bundle <bundle>

Option for identifying the bundle parameter.

<bundle> Character string expression for matching with the bundle in the module data filename.

-activation_date <activation_date>

Option for specifying the activation date.

<activation_date> Character string expression for matching with the activation_date in the module data filenames. The date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-keyed <key>

Option for selecting only one file for every NAMS ID. The file with the highest version of the specified key is selected.

-dated <date>

Option for selecting only one file for every NAMS ID. The file with the activation date closest to, but not exceeding, the specified date is selected.

<date> Date is valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-debug

Generate information used for debugging.

-nrs_list <file>

This options allows NRS files to be specified manually. This option can be used instead of the <nrs_selector> options to specify NRS files.

> ***Note:*** The output of the *nrslist* command is acceptable input as an NRS list file.

<file> The NRS list file contains one NRS data file per line. The complete file path must be given. Leading and trailing blanks are acceptable as are blank lines. Comment lines are permitted, but they must start with a # character.

-accounting

Accounting parameters are checked for network-wide consistency.

-atm

Semantic checks on ATM PVCs to detect inconsistencies or problems in the PVCs across Passport nodes.

-call_forward

Semantic checks on all call forward DNAs and RIDs will be executed.

-direct_call

Semantic checks on all direct calls will be executed.

-dna_uniqueness

Sematic checks on all DNAs will be executed.

This check is always performed when the call redirection, hunt group, direct call or PVC checks are requested

-external_dna_file <dna_file>

The external DNA file contains a list of DNAs which are not defined in the network data view but are still considered to be valid. This file usually contains a list of DNAs that are provisioned on network modules (NMs) and are referenced in an AM/RM. For example, as the remote end of a PVC, a direct call or a redirected DNA.

<dna_file> External DNAs are entered in the file one per line, in X121 or E164 format. Leading and trailing blanks are acceptable as are blank lines. Comment lines are allowed but the first non-blank character must be a #.

-external_ipaddr_file <ipaddr_file>

The external IP address file contains a list of IP addressess which are not defined in the network data view but are still considered to be valid.

<ipaddr_file> External IP addresses are entered in the file one per line. Leading and trailing blanks are acceptable as are blank lines. Comment lines are allowed but the first non-blank character must be a #.

-external_rid_file <rid_file>

The external RID file contains a list of RIDs which are not defined in the network data view but are still considered to be valid.

<rid_file> Rids are entered in the file one per line. They can range from 1 to 126. Leading and trailing blanks are acceptable as are blank lines. Comment lines are allowed but the first non-blank character must be a #.

-gid_uniqueness

Semantic checks on gateway IDs will be executed.

-h

Option to display command line usage information.

-hunt_group

Semantic checks on hunt groups will be executed.

-ipaddr_uniqueness

Uniqueness checks on all IP addresses will be executed.

-mid_uniqueness

Semantic checks on MIDs defined in the Module_Data component are executed.

-module_name_uniqueness

Semantic checks on module mnemonics defined in the DPN Module_Data component and on node names defined in the Passport ModuleData component, will be executed.

-Namsid_uniqueness

Semantic checks on NAMS IDs defined in the Module_data component will be executed

-noconfirm

By default NSIC will prompt for a confirmation before executing the checks. If this option is specified, you will not be prompted for confirmation.

-nodeid_uniqueness

Uniqueness checks on nodeIds defined in the Passport ModuleData component will be executed.

-oa_uniqueness

Semantic checks on OA names and types defined in the OA Link component will be executed.

-pvc

Semantic checks on PVCs will be executed.

-atm

Semantic checks on ATM PVCs across Passport nodes will be executed.

-redirection

Semantic checks on call redirection will be executed.

-rid_map

Semantic checks on RIDs will be executed.

-sorted <sorted output filename>

By default, NSIC generates unsorted messages as errors are found. With this option the messages are sorted by node name and saved in the specified file. A summary is also provided.

-voice

Semantic checks on voice networking will be executed.

-vxid_uniqueness

Semantic checks on virtual XIDs will be executed.

Example

Dated access, all checks, an external DNA file is supplied.

```
/opt/MagellanNMS/bin/nsic -dated 920612
-external_dna_file ext.dnas
```

Keyed access with DNA uniqueness and PVC checks only.

```
/opt/MagellanNMS/bin/nsic -keyed R7BASE
-dna_uniqueness -pvc
```

Table 13
NSIC return code description

Return code	Description
0	The checks were successfully executed and if selected, the sorted output file was generated.
1	Command line validation error.
2	A termination signal (Control-C) was received.
3	Unable to generate the sorted output file.

NRS Hierarchical Report Generator (nrshier)

The NRS Hierarchical Report Generator (nrshier) is used to help create reports on DPN and Passport service data that is organized in a hierarchy.

By default, this tool creates a pre-nawk report template corresponding to the specifications (an NRS component name, marking the top of the sub-tree within the service data tree being considered for the report, and a list of fields to be included in the report). After it is created, the pre-nawk report template must be compiled using the NRS Preprocessor.

The -create_profile option is provided to create a profile instead of creating the pre-nawk report template. This can be used to create a list of all the fields that could be included in the report without having to search the RDFs. This profile could then be edited to remove unwanted fields and used as the profile file for generating the pre-nawk report template.

The NRS Hierarchical Report Generator is found in the /opt/MagellanNMS/bin directory uses the NRS_RDF_DIR symbol from the NRS configuration file NRS.cfg. The output of this tool (pre-nawk report template or profile) is written to stdout while error messages are written to stderr.

This section outlines both the command line syntax and the syntax of the profile file.

For return codes, see the table “NRS Hierarchical Report Generator return code description” (page 272).

Command syntax

Enter the following command syntax as one continuous command.

```
nrshier <top_of_hierarchy_component>
  [-profile <profile_file>] [-create_profile] [-h]
```

where:

<top_of_hierarchy_component>

An NRS component name. This component marks the top of the sub-tree within the service data tree where fields can be selected from. The value should be of format <RDF type>/<component name>.

-profile <profile_file>

Option to specify a profile file. If not specified, all the fields of all the components part of the sub-tree are selected.

-create_profile

Option to create the profile instead of the pre-*nawk* report template.

-h

Option to display command line usage information.

Example

```
nrshier dpn/SPOOLING_SDA -profile my_spooling_fields >
spool_rpt

nrshier dpn/SpOoLiNg_SdA -create_profile >
all_spooling_fields

nrshier dpn/x25

nrshier dpn/x25 -profile a_profile -create_profile

nrshier dpn/CUG > cug_rpt

/opt/MagellanNMS/bin/nrshier dpn/PO -profile
a_few_service_fields > short_po_rpt
```

Profile file syntax

The following is the syntax for the profile file. This profile is used to specify the fields to be included in the report. Comment and blank lines are allowed but comment lines must start with a # character.

```
<component_name> [<field_name>]
```

where:

<component_name>

NRS component name. Each NRS component must have a corresponding RDF.

<field_name>

NRS field name. The field must exist in the RDF of the specified <component_name>.

Example

The following is an example of a profile for a report on DPN X.25.

```
# This is a sample profile file for a report
# on the X.25 service. These first 3 lines
# are comments and are ignored by the tool.
PM NAMSID
PM BUNDLE
X25
x25link LINESPEED
x25dnacug _X25MDNA
X25DNACUG x2receivewindowsize
X25DNACUG X2TRANSMITWINDOWSIZE
X25DNACUG CUSTOMER_DATA
x25directcall _X25DIRECTCALL
X25DIRECTCALL REMOTEDNA
X25PVC _X25pvc
X25PVC remotedna
```

Profile rules

When the -profile option is not specified, all the fields of all the components part of the sub-tree are selected (except for fields _COMPONENT and _HIERARCHY_LEVEL that are not real service data fields). In DPN profiles, the following four fields from the PM component are included: _PM, NAMSID, BUNDLE, and ACTIVATION_DATE. In Passport profiles, the _2, NAMSID, VIEW_FILENAME, and ACTIVATION_DATE from the EM component are also included. These fields are included since they are really part of the key of each record. For example, a PE cannot be identified just by the PE number. The PM, NAMSID, BUNDLE, ACTIVATION_DATE and PE number uniquely identify a PE.

If a profile file is specified, only the fields of the components part of the sub-tree can be selected as well as fields from the PM record. If on a given line a component name without a field name is entered, all the fields in that component are included (except for fields _COMPONENT and _HIERARCHY_LEVEL).

A special field called CUSTOMER_DATA can be specified on every component. When this field is requested, the customer data database is queried for the component. Since customer data is only supported on specific components (DNA, PO, NCS OA, NCS application and PM), the component is checked to determine the type of customer data component that should be used.

Table 14
NRS Hierarchical Report Generator return code description

Return code	Description
0	Successful completion, no errors
1	Error condition

NRS RDF Hierarchy Generator (nrsrdfh)

The NRS Hierarchy Generator (nrsrdfh) is a tool used to help determine the relationships between the various NRS components as well as the mapping between category name and UI long and short names for all components and fields. This tool is called *nrsrdfh* and can be found in the /opt/MagellanNMS/bin directory. It uses the NRS_RDF_DIR symbol from the NRS configuration file NRS.cfg.

The nrsrdfh tool can be called with an optional <top_of_hierarchy_component> parameter that must correspond to any of the NRS components. The format should be <RDF_type>/<component_name>.

If this tool is invoked without the <top_of_hierarchy_component> option, the entire hierarchy will be printed for RDF type, dpn, ppc, and ppe, if they exist.

The output format of this tool is as follows:

```
<category name of the component> / <UI long name of the component>
[(<UI short name of the component>)]
   <category name of child 1> / <UI long name of child 1>
   [(<UI short name of child 1>)]
      <category name of grandchild 1.1> / <UI long name of grandchild 1.1>
      [(<UI short name of grandchild 1.1>)]
     <category name of grandchild 1.2> / <UI long name of grandchild 1.2>
     [(<UI short name of grandchild 1.2>)]
 ...
   <category name of child 2> / <UI long name of child 2>
   [(<UI short name of child 2>)]
     <category name of grandchild 2.1> / <UI long name of grandchild 2.1>
     [(<UI short name of grandchild 2.1>)]
 ...
```

Note: The UI short name is not printed if it is the same as the UI long name.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsrdfh [<top_of_hierarchy_component>] [-fields] [-h]
```

where:

<top_of_hierarchy_component>

An NRS component name. This component marks the top of the sub-tree within the service data tree. The value should be of format <RDF_type>/<component_name>.

-fields

Mapping of all fields in the service data hierarchy.

-h

Option to display command line usage information

Example

```
nrsrdfh -fields > all_hierarchy
nrsrdfh dpn/iti
/opt/MagellanNMS/bin/nrsrdfh ppc/1218
nrsrdfh ppc/1218 -fields
```

The following is an example of the output format for the DPN ITI component when the */opt/MagellanNMS/bin/nrsrdfh dpn/iti* command is entered.

```
13:42:32 NOTE:Extracting the hierarchy of RDF type dpn,
component ITI.
ITI / ITI
   ITILINK / Link
   ITILDNA / DNA
      ITIDNACUG / DNA_CUG
         CUGS / NCUGs
            CUG / NCUG_Index
         ICUGS / ICUGs
            ICUG / ICUG_Index
      ITIPVC / PVC
      ITIHUNTDNA / Hunt_DNA
   ITIUSERPROFILE / User_Profile_0
   ITIINTERFACE / Interface
   ITIAUTODIRECT / Auto_Direct
   ITIVIDEOTEXAPPLICATION / Videotex_Application
   ITINUIANDTRANSLATIONOPTIONS /
   NUI_And_Translation_Options
   MI8_CUS_MODEM_PROF_ENV / Custom_Modem_Profile
```

The following is an example of the output format for the Passport (ppc) 1218 component when the */opt/MagellanNMS/bin/nrsrdfh ppc/1218* command is entered.

```
14:28:12 NOTE:Extracting the hierarchy of RDF type ppc,
component 1218.
1218 / AccessControl (Ac)
   1226 / IpAccess
   1219 / Userid
```

The following is an example of the output format for the Passport (ppc) 1218 component when the */opt/MagellanNMS/bin/nrsrdfh ppc/1218 -fields* command is entered.

```
16:18:10 NOTE:Extracting the hierarchy of RDF type ppc,
component 1218.
1218 / AccessControl (Ac)
   _COMPONENT / Component
   _HIERARCHY_LEVEL / Hierarchy_level
   _2 / EM
   _1218 / AccessControl (Ac)
   OAM / Ownership
   2296 / publicKeyAuth (keyAuth)
   1226 / IpAccess
      _COMPONENT / Component
      _HIERARCHY_LEVEL / Hierarchy_level
      _2 / EM
      _1218 / AccessControl (Ac)
      _1226 / IpAccess
      OAM / Ownership
      1228 / ipAddressMask (mask)
   1219 / Userid
      _COMPONENT / Component
      _HIERARCHY_LEVEL / Hierarchy_level
      _2 / EM
      _1218 / AccessControl (Ac)
      _1219 / Userid
      OAM / Ownership
      1222 / customerIdentifier (cid)
      1223 / commandScope (scope)
      1224 / commandImpact (impact)
      1225 / allowedAccess (nmifs)
      1265 / loginDirectory (dir)
      2293 / allowedOutAccess (outAccess)
```

For return codes, see the table “NRS RDF Hierarchy Generator return code description” (page 276).

Table 15
NRS RDF Hierarchy Generator return code description

Return code	Description
0	Successful completion, no errors
1	Error condition

NRS RDF Owners Hierarchy Generator (nrsowner)

The NRS RDF Owners Hierarchy Generator (nrsowner) tool is used to determine the owners hierarchy of NRS components. This tool uses the NRS_RDF_DIR symbol from the NRS configuration file NRS.cfg and is found in the /opt/MagellanNMS/bin directory.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsowner <component_name>
       [-tree | -list | -path]
       [-h]
```

or

```
nrsowner <file_name> -file <RDF type>
       [-tree | -list | -path]
       [-h]
```

or

```
nrsowner <field_name> -field
       [-h]
```

where:

<component_name>

An NRS component name whose owners hierarchy within the service data tree is to be determined. The value should be of format <RDF_type>/<component>.

-tree

Option to select output in *tree* format. This is the default output format.

-list

Option to select output in *list* format where each component is listed exactly once.

-path

Option to select output in *path* format.

-file <RDF_type>

Option to indicate that the first parameter is to be treated as a file name. The RDF type must also be specified to indicate under which RDF hierarchy the components specified in the file belong.

<file_name>

Name of file containing a list of NRS component names, one per line, whose owners hierarchy within the service data tree are to be determined.

<field_name>

The name of the NRS component field or attribute. The value should be of format <RDF_type>/<field>.

-field

Option to list all fields with <field_name> in path format.

-h

Option to display command line usage information.

Example

The following are examples of command lines and their output.

```
/opt/MagellanNMS/bin/nrsowner dpn/ITILDNA
```

```
PM / PM
  PE / PE
    PI / PI
      PO / PO
        ITI / ITI
          ITILDNA / DNA
PM / PM
  PE / PE
    PI / PI
      PO / PO
        DS0B / DS_0B_Application
          DS0BITI / ITI
            ITILDNA / DNA
```

```
/opt/MagellanNMS/bin/nrsowner dpn/ITILDNA -list
```

```
ITILDNA
ITI
PO
PI
PE
PM
DS0BITI
DS0B
```

```
/opt/MagellanNMS/bin/nrsowner dpn/ITILDNA -path
```

```
PM / PE / PI / PO / ITI / ITILDNA
PM / PE / PI / PO / DS0B / DS0BITI / ITILDNA
```

```
/opt/MagellanNMS/bin/nrsowner infile -tree -file dpn
```

The infile contains:

```
ITILDNA
X25MDNA
```

The output is:

```
PM / PM
  PE / PE
    PI / PI
      PO / PO
```

```
        ITI / ITI
          ITILDNA / DNA
PM / PM
  PE / PE
    PI / PI
      PO / PO
        DS0B / DS_0B_Application
          DS0BITI / ITI
            ITILDNA / DNA
PM / PM
  PE / PE
    PI / PI
      PO / PO
        X25 / X25
          X25MDNA / DNA
PM / PM
  PE / PE
    PAM_SDA / PE_Servers
      X25_ML_AGENT / X25_Multilink_Agent
          X25MDNA / DNA
```

`/opt/MagellanNMS/bin/nrsowner dpn/DNAVXID -field`

```
PM / PE / PI / PO / SNA / SNAPU / SNAXPAD / SNADNACUG
DNAVXID
PM / PE / PI / PO / SNA / SNAPU / SNAMULTIHOST /
SNADNACUG DNAVXID
PM / PE / PI / PO / SNA / SNAPU / SNAMULTIHOST / SNALU
/ SNADNACUG DNAVXID
PM / PE / PI / PO / SNA / SNAPU / SNALUMULTPAD / SNALU
/ SNADNACUG DNAVXID
PM / PE / PI / PO / SNA / SNAPU / SNAT21 / SNADNACUG
DNAVXID
PM / PE / PI / PO / SNA / SNAPU / SNALUMULTPAD /
SNADNACUG DNAVXID
```

For return codes, see the table “NRS RDF Owners Hierarchy return code description” (page 280).

Table 16
NRS RDF Owners Hierarchy return code description

Return code	Description
0	Successful completion, no errors
1	Error condition

NRS Preprocessor (nrscomp)

The NRS Preprocessor (nrscomp) translates the NRS nawk language extensions into executable nawk code. The NRS Preprocessor is invoked with the nrscomp shell which is found in the /opt/MagellanNMS/bin directory. The preprocessor uses the NRS_INCLUDE_DIR and NRS_RDF_DIR symbols from the NRS configuration file NRS.cfg.

The report output can be directed to an output file. If the compile is successful the file protection on the output file specified will be changed to executable. If an output file is not specified, the output is directed to stdout.

The NRS language extensions recognized are:

- field data value reference
- POS
- TYPE
- WIDTH
- TITLE
- ABBREVIATION
- GROUP
- INCLUDE

For more information on these particular language extensions, see “Programming with NRS” (page 157).

Command syntax

Enter the following command syntax as one continuous command.

```
nrscomp <report_file> [<output_file>] [-h]
```

where:

<report_file>

Name of the report program written in *nawk* with the NRS language extensions.

<output_file>

Name of the file to write the compiled output to.

-h

Option to display the command line usage information

Example
`nrscomp myreport`

```
/opt/MagellanNMS/bin/nrscomp nrsx25sh.rpt nrsx25sh
```

For return codes, see the table “NRS Preprocessor return code description” (page 281).

Table 17
NRS Preprocessor return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition

NRS Component Joiner (nrsjoin)

The NRS Component Joiner (nrsjoin) takes two NRS components as input and joins them when common fields have identical values. The Component Joiner command line parameters are the components to be joined and the fields to be used. If the NRS nawk-based tools are to used on the results of a join the resulting component should not have more than 199 fields.

When two components are joined, the resulting new component contains all the fields of both the original components. The new component first contains all of the fields of component 1, then the field delimiter, followed by all the fields of component 2. Duplicate fields are not eliminated. Field separators are copied to the new component structure.

The -b option allows a new component to be created when there is no match for component 1 with component 2. In this case, the new component contains the fields from component 1, then the field delimiter, followed by a number of empty fields corresponding to the number of fields in component 2. Note that the unmatched components are mixed with the matched components.

The number of fields in the resulting record can be restricted by using the -p1 or -p2 option followed by the position of the fields that are required. In this case, the new component contains the selected fields of component 1, then the field delimiter, followed by the selected fields from component 2.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsjoin -f <filename1> <filename2> -j <field
positions> |   -j1 <field positions> -j2 <field
positions>
  [-t <field separator>] [-c <component name>]
  [-p1 <field positions>] [-p2 <field positions>] [-b]
  [-h]
```

where:

-f <filename1> <filename2>

Mandatory option to use input files.

<filename1> Relation 1, individual record in filename1 (must be sorted on join fields).
<filename2> Relation 2, individual record in filename2 (must be sorted on join fields).

-j <field positions> -j1 <field positions -j2 <field positions>

Mandatory option for field positions to be joined. *j1* to specify field positions for filename1, *j2* to specify field positions for filename2, and *j* if field positions are the same for both files.
Note: Field positions start at 1 not 0.

-t <field separator>

Option to use the field separator. The default is the <DEL> character.

<field_separator> Specifies the character which is used as a field separator for the input files. The same separator character is used in the output file.

-c <component name>

Option to introduce a component name in front of each resulting record. The default is no component name. Specifies the value to be used as the first field of each record.

-p1 <field positions>

Option to specify the field positions from component 1 to be kept in the resulting record. If not specified, all fields from component 1 will be kept.

-p2 <field positions>

Option to specify the field positions from component 2 to be kept in the resulting record. If not specified, all fields from component 2 will be kept.

-b

Option to output unmatched records from component 1 as well as the matched records.

-h

Option to display command line usage information.

Example

```
nrsjoin -f 1st.file 2nd.file -j1 6 4 -j2 1 2
nrsjoin -f 1st.file 2nd.file -j 3 1 4 -c A_RECORD

nrsjoin -f 1st.file 2nd.file -j1 6 4 -j2 1 2 -p1 1 2
   -p2 4 5 -b
```

For return codes, see the table “NRS Component Joiner return code description” (page 284).

Table 18
NRS Component Joiner return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition
2	Command line or input file validation failure.

NRS Reporter (nrsrpt)

The Reporter (nrsrpt) is a tool that provides a convenient command line interface for running NRS reports. This tool is called nrsrpt and is found in the /opt/MagellanNMS/bin directory. It uses the NRS_DATA_DIR symbol from the NRS configuration file NRS.cfg.The command line interface accepts most of the nsrlist parameters (type, pm, namsid, bundle, activation_date, unique, dated and keyed).

See “NRS Module Filename Lister (nrslist)” (page 291) for a complete description of these parameters.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsrpt <report_program>
  [-type <module_type>] [-pm <pm>] [-namsid <namsid>]
  [-bundle <bundle>|-viewfile <view_filename>]
  [-activation_date <activation_date>]
  [-dated <date> | -keyed <key> | -unique]
  [<parameters or options to be passed to the report>]
  [-h]
```

where:

<report_program>

Name of report program to be run.

-type <module_type>

Option for identifying the module type parameter.

<module_type> Character string expression for matching with the module type in the module data filenames.

-pm <pm>

Option for identifying the PM parameter.

<pm> Character string expression for matching with the PM mnemonic in the module data filenames.

-namsid <namsid>

Option for identifying the NAMS ID parameter.

<namsid> Character string expression for matching with the NAMSID in the module data filenames.

-bundle <bundle>

Option for identifying the bundle parameter.

<bundle> Character string expression for matching with the bundle in the module data filename.

-viewfile <view_filename>

Option used to specify the provisioning file name.

<view_filename> Character string expression used to match the name of the Passport view in the module data files. This expression is not converted to uppercase before performing the match.

-activation_date <activation_date>

Option for specifying the activation date parameter.

<activation_date> Character string expression for matching with the activation_date in the module data filenames. Date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-dated <date>

Option for selecting only one file for every NAMS ID. The file with the activation date closest to, but not exceeding, the specified date is selected. Date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-keyed <key>

Option for selecting only one file for every NAMS ID. The file with the highest version of the specified key is selected.

-unique

Option for selecting only one file for every NAMS ID. The file with the highest bundle (in ASCII) is selected.

<parameters or options to be passed to the report>

Values to be passed to report (for reports that accept command line parameters).

-h

Option to display command line usage information.

Example

```
nrsrpt X25DC -bundle ‘^MYBUNDLE$’
nrsrpt myreport -na ‘^4034$’ -pm ‘^R34$’
nrsrpt PVC -namsid ‘^40..$’ -unique > PVC.txt
nrsrpt net_env -type ‘^dpn$’
/opt/MagellanNMS/bin/nrsrpt X25DC -act 9304\.\. -
unique

setenv REGION_1 ’^4034$|^4035$|^5026$’
/opt/MagellanNMS/bin/nrsrpt -namsid $REGION_1

nrsrpt nrsppmod -viewfile ‘^MyView,’
```

For return codes, see the table “NRS Reporter return code description” (page 287).

Table 19
NRS Reporter return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition

NRS Module File Deletor (nrsrm)

The Module File Deletor (nrsrm) is a convenient way to delete some of the DPN and Passport module data files from the NRS database. This tool is called nrsrm to be consistent with the function of the UNIX rm command. It is found in the /opt/MagellanNMS/bin directory and uses the NRS_DATA_DIR symbol from the NRS configuration file NRS.cfg.

The tool accepts many parameters, of which five parameters correspond to the five parts of the module data filenames. They are module type, PM, namsid, bundle and activation_date. For each of these parameters a full regular expression can be specified and the tool removes all files which match the command line parameters. Regular expressions must be in quotation marks or each special character of the regular expression must be preceded with a backslash character.

The -purgedated and -purgekeyed options are provided to further reduce the number of files deleted. See “NRS Module Filename Lister (nrslist)” (page 291) for a complete description of these parameters. By default, this tool prompts you for a confirmation on every file before it is deleted. This can be bypassed by specifying the -force option. This option should be used very carefully.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsrm [-type <module_type>] [-pm <pm>] [-namsid
<namsid>]
  [-bundle <bundle> | -viewfile <view_filename>]
  [-activation_date <activation_date>]
  [-purgedated <date> | -purgekeyed <key>] [-force]
  [-h]
```

where:

-type <module_type>

Option for identifying the module type parameter.

<module_type> Character string expression for matching with the module type in the module data filenames.

-pm <pm>

Option for identifying the PM parameter.

<pm> Character string expression for matching with the PM mnemonic in the module data filenames.

-namsid <namsid>

Option for identifying the NAMS ID parameter.

<namsid> Character string expression for matching with the NAMS ID in the module data filenames.

-bundle <bundle>

Option for identifying the bundle parameter.

<bundle> Character string expression for matching with the bundle in the module data filename.

-viewfile <view_filename>

Option used to specify the provisioning file name.

<view_filename> Character string expression used to match the name of the Passport view in the module data files.

-activation_date <activation_date>

Option for specifying the activation date parameter.

<activation_date> Character string expression for matching with the activation_date in the module data filenames. Date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-purgedated <date>

Option for selecting the module files that have an activation date less than or equal to the specified date except the highest one for each NAMS ID.

<date> Date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-purgekeyed <key>

Option for selecting the module data files where the bundle matches the specified key except the highest one of each NAMS ID.

-force

Option to delete the files without confirmation.

-h

Option to display command line usage information.

Example

```
nrsrm -purged 930401
nrsrm -bun ‘^9304.*’
nrsrm -pm ‘^R34$’
nrsrm -nams ‘^4034$’ -act 9304‘..’ -purgek WK13
nrsrm -purged 930401 -force
```

For return codes, see the table “NRS Module File Deletor return code description” (page 290).

Table 20
NRS Module File Deletor return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition

NRS Sanity Checker (nrsanity)

The NRS Sanity Checker (nrsanity) is a tool used to ensure that the DPN and Passport NRS data files are compatible with the current RDFs. This tool is called nrsanity and can be found in the /opt/MagellanNMS/bin directory. It uses the NRS_DATA_DIR and NRS_RDF_DIR symbols from the NRS configuration file NRS.cfg.

The RDF version is stored in each NRS data file by the NRS Populator. The PM.rdf file, in the dpn subdirectory, contains the DPN RDF version. The 2.rdf file, in the ppc and ppe subdirectories, contains the Passport RDF version for the corresponding Passport family. This tool ensures that the RDF version in the NRS data file is at least a sub-string of the RDF version found in the corresponding RDF file.

Command syntax

nrsanity [-h]

where:

-h

Option to display command line usage information.

Example

```
nrsanity > result

/opt/MagellanNMS/bin/nrsanity
```

For return codes, see the table “NRS Sanity Checker return code description” (page 291).

Table 21
NRS Sanity Checker return code description

Return code	Description
0	Successful completion, no errors
1	Error condition

NRS Module Filename Lister (nrslist)

The Module Filename Lister (nrslist) tool is a convenient way of selecting DPN and Passport module data files. This tool is called nrslist and is found in the /opt/MagellanNMS/bin directory. It uses the NRS_DATA_DIR symbol from the NRS configuration file NRS.cfg. The tool accepts many parameters, of which five parameters correspond to the five parts of module data filenames. They are: module type, PM, namsid, bundle and activation date. For each of these parameters a full expression can be specified and the tool outputs all filenames which match the command line parameters. Regular expressions must be in quotation marks or each special character of the regular expression must be preceded with a backslash character. See “Regular expressions” (page 325) for more information.

When a parameter is not explicitly specified in the command line, that parameter defaults to match everything. For example, nrslist invoked with no parameters would be equivalent to:

nrslist -type ‘.*’ -pm ’.*’ -na ’.*’ -bu ’.*’ -act ’.*’

Special options are provided to further reduce the number of files selected. They are: -unique, -dated, -keyed, -purgedated and -purgekeyed.

-unique

The -unique option is used to select only one file for every NAMS ID. The file with the highest bundle (in ASCII) is selected.

Example

```
dpn.R34.4034.99121912.991219.data
dpn.R34.4034.99122151.991221.data
dpn.R34.4034.99123003.991230.data--> is selected for
4034

dpn.A20.2020.99111107.991111.data
dpn.A20.2020.99112301.991123.data--> is selected for
2020

ppc.A41.3041.524,full,001.991011.data
ppc.A41.3041.526,full,003.991027.data
ppc.A41.3041.530,full,002.991123.data --> is selected
for 3041

ppe.R51.5051.99122501,full,001.990225.data
ppe.R51.5051.MYBUN10,full,001.990225.data
ppe.R51.5051.MYBUN12,full,001.990225.data --> is
selected for 5051
```

-dated <date>

The -dated option is used to select only one file for every NAMS ID. The file with the activation date closest to (without exceeding) the specified date is selected. If more than one file has the same activation date, the following rules on the bundle are used.

There are two types of bundles:

- NMS DATED bundles
- Other NMS bundles (KEYED,USER_SPECIFIED)

If the bundle types are different, an NMS DATED bundle is considered higher than another NMS bundle. If the bundle types are equal, the file with the highest bundle (in ASCII) is selected.

Example

The following is displayed if the specified date is 991027 (October 27, 1999).

```
dpn.R34.4034.99101912.991019.data
dpn.R34.4034.99102151.991021.data
dpn.R34.4034.99102157.991021.data --> is selected for
4034
```

```
dpn.R34.4034.99103003.991030.data
dpn.R34.4034.987.991021.data
dpn.R34.4034.MYBUN99.991021.data

dpn.A20.2020.99111107.991111.data
dpn.A20.2020.99112301.991123.data No module selected
for 2020

ppc.A41.3041.524,full,001.991011.data
ppc.A41.3041.526,full,001.991027.data --> is selected
for 3041
ppc.A41.3041.530,full,001.991123.data

ppc.R51.5051.165,full,001.990225.data
ppc.R51.5051.99022501,full,001.990225.data--> is
selected for 5051
ppc.R51.5051.MYBUN12,full,001.990225.data
```

-keyed <key>

The -keyed option is used to select only one file for every NAMS ID. The file with the highest version of the specified key is selected.

Example

The following is displayed if the specified key is WK13.

```
dpn.R34.4034.99101912.991019.data
dpn.R34.4034.WK1201.990403.data
dpn.R34.4034.WK1202.990403.data
dpn.R34.4034.WK1304.990410.data
dpn.R34.4034.WK1305.990410.data--> is selected for
4034
dpn.R34.4034.WK1410.990417.data

dpn.A20.2020.WK1207.990403.data
dpn.A20.2020.WK1401.990417.data No module selected for
2020
```

-purgedated <date>

The -purgedated option is used to select the module files that have an activation date less than or equal to the specified date except the highest one for each NAMS ID. If more than one file has the same highest activation date, the rules described for the -dated option apply.

Example

The following is displayed if the specified date is 991027 (October 27, 1999).

```
dpn.R34.4034.99101912.991019.data--> is selected for
4034
dpn.R34.4034.99102151.991021.data--> is selected for
4034
dpn.R34.4034.99102157.991021.data
dpn.R34.4034.99103003.991030.data
dpn.R34.4034.987.991021.data --> is selected for 4034
dpn.R34.4034.MYBUN97.991021.data --> is selected for
4034

dpn.A20.2020.99111107.991111.data
dpn.A20.2020.99112301.991123.dataNo module selected
for 2020

ppc.A41.3041.524,full,001.991011.data   --> is
selected for 3041
ppc.A41.3041.526,full,001.991027.data
ppc.A41.3041.530,full,001.991123.data

ppe.R51.5051.165,full,001.981225.data --> is selected
for 5051
ppe.R51.5051.98122501,full,001.991225.data
ppe.R51.5051.MYBUN12,full,001.981225.data --> is
selected for 5051
```

The -purgedated option is normally used when deleting module data files.

-purgekeyed <key>

The -purgekeyed option is used for each NAMS ID, to select the files with the bundle following the specified key except for the highest version.

Example

The following is displayed if the specified key is WK13.

```
dpn.R34.4034.99102157.991021.data
dpn.R34.4034.99103003.991030.data
dpn.R34.4034.987.991021.data
dpn.R34.4034.MYBUN97.991021.data
dpn.R34.4034.WK1313.991019.data--> is selected for
4034
dpn.R34.4034.WK1314.991019.data

dpn.A20.2020.WK1224.991111.data
dpn.A20.2020.WK1310.991123.dataNo module selected for
2020

ppc.A41.3041.WK1300,full,001.991011.data--> is
selected for 3041
ppc.A41.3041.WK1301,full,001.991027.data--> is
selected for 3041
ppc.A41.3041.WK1302,full,001.991123.data

ppe.R51.5051.165,full,001.991225.data
ppe.R51.5051.99122501,full,001.991225.data
ppe.R51.5051.MYBUN12,full,001.991225.dataNo module
selected for 5051
```

The -purgekeyed option is normally used when deleting module files.

These five options are mutually exclusive. The -dated and -purgedated options are handled by the nrsdated tool which is found in the /opt/MagellanNMS/bin directory.

Command syntax

Enter the following command syntax as one continuous command.

```
nrslist[-type <module_type>] [-pm <pm>]
  [-namsid <namsid>]
  [-bundle <bundle> |-viewfile <view_filename>]
  [-activation_date <activation_date>]
  [-dated <date> | -keyed <key> | -unique |
  -purgedated <date> | -purgekeyed <key>] [-h]
```

where:

-type <module_type>

Option for identifying the module type parameter.

<module_type> Character string expression for matching with the module type in the module data filenames.

-pm <pm>

Option for identifying the PM parameter.

<pm> Character string expression for matching with the PM mnemonic in the module data filenames.

-namsid <namsid>

Option for identifying the NAMS ID parameter.

<namsid> Character string expression for matching with the NAMS ID in the module data filenames.

-bundle <bundle>

Option for identifying the bundle parameter.

<bundle> Character string expression for matching with the bundle in the module data filename

-viewfile <view_filename>

Option used to specify the provisioning file name.

<view_filename> Character string expression used to match the name of the Passport view in the module data files.

-activation_date <activation_date>

Option for specifying the activation date parameter.

<activation_date> Character string expression for matching with the activation_date in the module data filenames. Date must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-dated <date>

Option for selecting only one file for every NAMS ID. The file with the activation date closest to (without exceeding) the specified date is selected.

<date> Date must be valid and in the format *yymmdd*. See "Date Convention" (page 27) for more information on the date format.

-keyed <key>

Option for selecting only one file for every NAMS ID. The file with the highest version of the specified key is selected.

-unique

Option for selecting only one file per NAMS ID. The file with the highest bundle (in ASCII) is selected.

-purgedated <date>

Option for selecting the module files that have an activation date less than or equal to the specified date except the highest one for each NAMS ID.

<date> Date must be valid and in the format *yymmdd*. See "Date Convention" (page 27) for more information on the date format.

-purgekeyed <key>

Option for selecting the module data files where the bundle matches the specified key except the highest one of each NAMS ID.

-h

Option to display command line usage information.

Example

```
nrslist -pm '^R34$' -bundle '^930401..'
nrslist -pm '^R4.*' -na '^4040$' -act 9304'..'
nrslist -pm 'R4.*' -namsid 4040
nrslist -pm R4\.\* -nams 404'.+'
nrslist -pm R34 -nams 4034 -bundle '^WK..07$'
nrslist -pm R34 -namsid 4034 -bundle WK34\.\.
nrslist -unique
nrslist
nrslist -dated 930401
nrslist -keyed WK13
nrslist -pm ‘^R34$’ -purgedated 930401
nrslist -type ‘^dpn$’ -purgekeyed WK34
nrslist -pm ‘^PP146$’ -viewfile ‘^MyView,’
```

For return codes, see the table “NRS Module Filename Lister return code description” (page 298).

Table 22
NRS Module Filename Lister return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition.

NRS Validator (nrsvalid)

The NRS Validator (nrsvalid) is a tool used by the other NRS UNIX tools like nrslist and nrscomp to retrieve the value of NRS variables (NRS_DATA_DIR, NRS_RDF_DIR, and so on) from the NRS configuration file NRS.cfg. The value of the variable is returned.

This tool looks for the NRS.cfg file in the MagellanNMS directory under the user’s home directory first. If not found, it looks in the /opt/MagellanNMS/bin directory.

This tool can be called from customer developed tools to retrieve the symbol(s) needed.

Command syntax

```
nrsvalid <nrs_variable> [-h]
```

where:

<nrs_variable>

One of the NRS variables found in the *NRS.cfg* file.

-h

Option to display command line usage information.

Example

```
nrsvalid NRS_DATA_DIR
nrsvalid NRS_RDF_DIR
nrsvalid NRS_DEFAULT_PP_TYPE
nrsvalid NRS_INCLUDE_DIR
nrsvalid NRS_CUST_DATA_DB_NAME
nrsvalid NRS_CUST_DATA_DB_DIR
nrsvalid NRS_CUST_DATA_DB_HOST
```

For return codes, see the table “NRS Validator return code description” (page 299).

Table 23
NRS Validator return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition

NRS Differences Report (nrsdiff)

The NRS Differences Report (nrsdiff) is used to compare two NRS data files and output the differences. This tool is called nrsdiff and is found in the /opt/MagellanNMS/bin directory.

Before performing the comparison, the NRS data files are first reformatted using the nrsdatah tool in order to make the output comparable to how the data is viewed from the Component Provisioning tool.

Each line in the list of differences starts with one of the following

- MOD

  A field has a different value in the two NRS data files. The field is displayed with the new value followed by (OLD: <old value>).

- ADD

  A component is present in the second NRS data file but not in the first file.

- DEL

  A component is present in the first NRS data file but not in the second file.

- ...

  Indicates a component or field that was added or deleted as part of a higher level add or delete. For example, if a port is added, all the components and fields below this port (the service, the DNA, the link, the line speed field, etc.) are also added. The port will be displayed with the word ADD at the beginning of the line and everything else below the port will have ... at the beginning of the line.

An example of the output is provided in the examples section.

The differences are printed to standard output and all the other messages are printed to standard error.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsdiff [-pm <PM>]
  -dated1 <date> | -keyed1 <key> | -bundle1 <bundle> |
  -viewfile1 <view file name> | -f1 <NRS file name>
  -dated2 <date> | -keyed2 <key> | -bundle2 <bundle> |
  -viewfile2 <view file name> | -f2 <NRS file name
  [-log [<log file name>]] [-component <component
name>]
  [-keepfiles <directory name>] [-quiet] [-h]
```

where:

-pm <PM>

Option to indicate which module to perform the differences on.

-dated1 <date> -dated2 <date>

Option to select the NRS data file with the activation date closest to (without exceeding) the specified date.

<date> Date used by the *-dated1* and *-dated2* options. It must be valid and in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format.

-keyed1 <key> -keyed2 <key>

Option to select the NRS data file with the highest version of the specified key.

<key> Key used by the *-keyed1* and *-keyed2* options.

-bundle1 <bundle> -bundle2 <bundle>

Option to select the NRS data file with the specified DPN bundle.

<bundle> Character string expression for matching with the bundle in the module data file names.

-viewfile1 <view file name> -viewfile2 <view file name>

Option to select the NRS data file with the specified provisioning file name.

<view file name> Character string expression used to match the name of the Passport view in the NRS data files.

-f1 <NRS file name> -f2 <NRS file name>

Option used to specify an NRS data file.

<NRS file name> The name of the NRS data file.

-log [<log file name>]

Option to have messages recorded in a logfile. The default file name is *nrsdiff.log* unless otherwise specified.

-component <component name>

Option used to specify a particular component to report differences on. If this option is not specified, the differences are performed on all the components found in the NRS data files.

<component name> An NRS component name for which to report differences on. The value should be of format <RDF type>/<component>.

-keepfiles <directory name>

Option to save the intermediate files used before performing the differences. The files are called *nrsdiff.nrs_1* and *nrsdiff.nrs_2* and are saved in the specified directory. If this option is not specified, the files are created in the */tmp* directory and are removed before exiting.

<directory name> Directory name where to save the intermediate files.

-quiet

Option to suppress the generation of information messages.

-h

Option to display the command line usage information.

Examples

The following example determines the differences between the NRS files dated 931201 (December 1, 1993) and 931225 (December 25, 1993) for PM R34.

```
nrsdiff -pm r34 -dated1 931201 -dated2 931225
```

The following example determines the differences between the ITI ports found in the NRS files
/opt/MagellanNMS/data/nrs/data/dpn.R34.4034.94011217.940112.data
and

/opt/MagellanNMS/data/nrs/data/dpn.R34.4034.94011903.940119.data. No informational messages are generated and all the output is saved in a file called *nrsdiff.log* (default).

```
nrsdiff -component dpn/ITI -quiet -log
-f1
/opt/MagellanNMS/data/nrs/data/
dpn.R34.4034.94011217.940112.data
-f2
/opt/MagellanNMS/data/nrs/data/
dpn.R34.4034.94011903.940119.data
```

The following example determines the differences between the NRS file */opt/MagellanNMS/data/nrs/data/dpn.R34.4034.94011217.940112.data* and the NRS file with bundle 563 for PM R56. All the output is saved in a file called *mylog*. The intermediate data files used to perform the differences are *nrsdiff.nrs_1* and *nrsdiff.nrs_2*. These files are located in the */localdisk/fred* directory.

```
nrsdiff -log mylog -keepfiles /localdisk/fred
   -f1
/opt/MagellanNMS/data/nrs/data/
dpn.R34.4034.94011217.940112.data -pm r56 -bu2 563
```

Following is an example of the output produced when the following command is entered.

```
nrsdiff -pm ac12 -dated1 940701 -dated2 940801
```

```
13:59:56 NOTE: The nrsdiff tool is starting.
Date: 94-08-24.
13:59:59 NOTE: NRS file 1 is
/opt/MagellanNMS/data/nrs/data/
dpn.AC12.1012.94062500.940625.data.
14:00:01 NOTE: NRS file 2 is /opt/MagellanNMS/data/
nrs/data/dpn.AC12.1012.94071604.940716.data.
14:00:01 NOTE: Reformatting NRS file 1.
14:00:27 NOTE: Reformatting NRS file 2.
14:00:49 NOTE: Determining the differences between the
2 NRS files.
MOD  PM/AC12 PE/1 PE_Loader
   PE_software_load:Load_file_name = ACCENT.PIMG3302
   (OLD: ACCENT.PIMG3205)
ADD  PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range
```

```
...  PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range Ownership =
   OWNER_IWS
... PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range Order = Ascend
... PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range LCN_base = 1
... PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range Total_LCNs =
   100
...  PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range Total_PVCs =
   0
... PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range #ovcs = 0
...  PM/AC12 PE/1 PI/1 PO/2 X25 LCN_Range #ivcs = 0
DEL  PM/AC12 Data_Spooling Direct_Call/2
...  PM/AC12 Data_Spooling Direct_Call/2 Ownership =
   OWNER_IWS
...  PM/AC12 Data_Spooling Direct_Call/2
   DNA_of_data_collection_system = X302103034142
...  PM/AC12 Data_Spooling Direct_Call/2 User_data = C4
ADD  PM/AC12 Data_Spooling Direct_Call/1
...  PM/AC12 Data_Spooling Direct_Call/1 Ownership =
   OWNER_IWS
...  PM/AC12 Data_Spooling Direct_Call/1
   DNA_of_data_collection_system = X302103034142
...  PM/AC12 Data_Spooling Direct_Call/1 User_data = C4
14:00:54 NOTE: The nrsdiff tool has completed but
errors/differences were found.
```

The first NRS data file can be determined by specifying the module name (using the *-pm* option) and an NRS data file selection option (one of *-dated1*, *-keyed1*, *-bundle1* or *-viewfile1*) or by specifying the NRS data file directly using the *-f1* option.

Similarly, the second NRS data file can be determined using the options with suffix 2 instead of 1.

For return codes, see the table “NRS Differences Report return code description” (page 305).

Table 24
NRS Differences Report return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition.
2	A termination signal (Control-C) was received.

NRS Data Hierarchy Report (nrsdatah)

The NRS Data Hierarchy Report tool is used to print the components or the components and their fields for DPN and Passport NRS data files.This tool is called nrsdatah and is found in the directory /opt/MagellanNMS/bin.

For each component, it prints the keys of the component and for each field, the keys of the component followed by the group name, field name and field value. Spaces in the group name and field name are replaced by underscores.

The nrsdatah tool can be called with an optional <component name> parameter that must correspond to any of the NRS components. When specified, only the sub-trees delimited by that component will be displayed.

This tool uses the content of the NRS data file(s) on standard input.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsdatah [<component name>] [-fields] [-nolistindex]
[-h]
```

where:

<component name>

An NRS component name. This marks the top of the sub-tree that is to be displayed in the service data tree. If not specified, all the components are displayed. The value should be of format <RDF type>/<component>.

-fields

Option to display the fields. If not specified, only the components are displayed.

`-nolistindex`

For key fields of list components, only print <title>, not <title>/<index>. This option is used by the nrsdiff tool.

-h

Option to display command line usage information.

Example

The following is an example of equivalent ways to use the *nrsdatah* tool and the output generated.

```
cat
/opt/MagellanNMS/data/nrs/data/
dpn.A34.4034.99032704.990327.data | /opt/MagellanNMS/
bin/nrsdatah dpn/x25
```

or

```
/opt/MagellanNMS/bin/nrsrpt
/opt/MagellanNMS/bin/nrsdatah -pm a34 -dated 990327
dpn/x25
```

```
PM/A34 NAMSID/4034 Bundle/99032704 Activation_Date/
   990327 Data_Version/NMS110Dai Model_Version/
   NMS110Dai RDF_Version/NMS110Dai
   Ownership/OWNER_IWS
PM/A34 PE/3 PI/3 PO/1 X25
PM/A34 PE/3 PI/3 PO/1 X25 Link
PM/A34 PE/3 PI/3 PO/1 X25 Interface
PM/A34 PE/3 PI/3 PO/1 X25 LCN_Range
PM/A34 PE/3 PI/3 PO/1 X25 DNA/X88884034403100
PM/A34 PE/3 PI/3 PO/1 X25 DNA/X88884034403100 DNA_CUG
PM/A34 PE/3 PI/3 PO/1 X25 DNA/X88884034403100 DNA_CUG
NCUGs
PM/A34 PE/3 PI/3 PO/1 X25 DNA/X88884034403100 DNA_CUG
ICUGs
```

```
PM/A34 PE/7 PI/7 PO/1 X25
PM/A34 PE/7 PI/7 PO/1 X25 Link
PM/A34 PE/7 PI/7 PO/1 X25 Interface
PM/A34 PE/7 PI/7 PO/1 X25 LCN_Range
PM/A34 PE/7 PI/7 PO/1 X25 DNA/X88884034407100
PM/A34 PE/7 PI/7 PO/1 X25 DNA/X88884034407100 DNA_CUG
PM/A34 PE/7 PI/7 PO/1 X25 DNA/X88884034407100 DNA_CUG
NCUGs
PM/A34 PE/7 PI/7 PO/1 X25 DNA/X88884034407100 DNA_CUG
ICUGs
(...)
```

Following is an example of the output generated when the following command is entered.

```
/opt/MagellanNMS/bin/nrsrpt
/opt/MagellanNMS/bin/nrsdatah -pm noder16 -dated
990901 ppc/427 -fields
```

```
EM/NODER16 NAMSID/2105 View_Filename/
   99082500,full,002 Activation_Date/990825
   Data_Version/BE0117A Model_Version/BE0117A
   RDF_Version/BE0117A Ownership/OWNER_IWS
EM/NODER16 Rtg
EM/NODER16 Rtg Ownership = OWNER_IWS
EM/NODER16 Rtg DebugProv:defaultTraceLevel =
EM/NODER16 Rtg Provisioned:tandemTraffic = allowed
EM/NODER16 Rtg splittingRegionIds
EM/NODER16 Rtg splittingRegionIds Ownership =
OWNER_IWS
EM/NODER16 Rtg Top
EM/NODER16 Rtg Top Ownership = OWNER_IWS
EM/NODER16 Rtg Dpn
EM/NODER16 Rtg Dpn Ownership = OWNER_IWS
EM/NODER16 Rtg Dpn Provisioned:logicalNetworkNumber =
1
EM/NODER16 Rtg Dpn Provisioned:routingId = 1
EM/NODER16 Rtg Dpn Provisioned:moduleId = 1105
(...)
```

For return codes, see the table “NRS Data Hierarchy Report return code description” (page 308).

Table 25
NRS Data Hierarchy Report return code description

Return code	Description
0	Successful completion, no errors.
1	Error condition.
2	A termination signal (Control-C) was received.

NRS Find DNA (nrsfinddna)

The nrsfinddna tool is used to locate a specified DNA or range of DNAs, and to generate the required DNA dictionary. It can be invoked from the Command Console or from a UNIX window.

DNA dictionary

The DNA dictionary contains the list of DNAs in the system, which the nrsfinddna tool requires to conduct its find operation. The dictionary is built using the nrsdnadict tool, or the -c option on the nrsfinddna tool. It is saved as *.dna_dictionary* in the directory referenced by the NRS_DATA_DIR variable in the NRS Configuration file. This allows the user to rebuild the dictionary periodically by creating a *cron* job (see 241-6001-304 *Preside MDM Configuration Management Administrator Guide* for details on running a cron job).

Command options

The nrsfinddna tool can be used simply to locate a particular DNA, to build a new or specified DNA dictionary, or to find a range of DNAs. Each of these options is detailed in the examples following.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsfinddna  <dna> [<NRS selection date>]
            [-create_dna_dictionary]
            [-h]
```

where:

```
<dna>
```

The DNA in the format X or E followed by digits. A regular expression can also be specified to match one or more DNAs.

The DNA is typed without any spaces.

```
<NRS selection date>
```

Specifies the NRS data file with the activation date closest to and not exceeding the specified date. It is a valid date in the format *yymmdd*. See “Date Convention” (page 27) for more information on the date format. The default date is 791231 (signifying the end of the final year in the range, 2079); it represents the latest view dictionary.

```
    -create_dna_dictionary
```

Option used to create the DNA dictionary. For each module, the NRS data file with the activation date closest to, but not exceeding, the <NRS selection date> is used to build the DNA dictionary.

```
    -h
```

The Help option displays command line usage information.

The following examples illustrate the different ways the *nrsfinddna* and *nrsdnadict* tools may be used.

Example 1: Finding a DNA

To find a specified DNA in the existing DNA dictionary, type:

```
nrsfinddna X302144556677
```

Example 2: Create the DNA dictionary and find a DNA

To find a specified DNA in the DNA dictionary, but first create that dictionary, type:

```
nrsfinddna X302144556677 -c 960810
```

For each module, the NRS data file with the activation date closest to but not exceeding 960810 (August 10, 1996) is selected when creating the dictionary.

Example 3: Finding a range of DNAs

To find all the DNAs that match the specified DNA pattern in the DNA dictionary, type:

```
nrsfinddna ‘X3021.45.*’
```

Single quotes are used here so that the shell does not interpret the special characters in its own way.

Example 4: Create a DNA dictionary

To create a DNA dictionary for the latest view of the network, type:

```
nrsdnadict
```

For each module, the NRS data file with the activation date closest to but not exceeding 791231 (December 31, 2079) is selected when creating the dictionary.

NRS Display Service (nrsdserv)

The NRS Display Service (nrsdserv) tool is used to display an image of all the ports configured on a specified PM. The tool searches the NRS database and displays the PE number, Image name, PI number, PI Board type, PO number, and Service type for all ports on that PM.

The nrsdserv tool can be invoked from the Command Console or from a UNIX window.

Command syntax

Enter the following command syntax as one continuous command.

```
nrsdserv<pm> [<NRS selection date>]
[-h]
```

where:

<pm>

The packet module desired.

<NRS selection date>

Specifies the NRS data file with the activation date closest to and not exceeding the specified date. It is a valid date in the format *yymmdd*.
See “Date Convention” (page 27) for more information on the date format.
The default date is 791231 (signifying the end of the final year in the range, 2079); it represents the latest view for the specified PM.

-h

The Help option displays command line usage information.

The following examples illustrate the different ways you can use the nrsdserv tool.

Example 1: Displaying the services of a selected PM

To display all the PEs, PIs, ports and the service types configured for Accent AC0301 using the NRS data file with the highest activation date, type:

```
nrsdserv AC0301
```

The following is produced:

```
PM: AC0301       NAMSID: 301          Bundle: WK1202


------ ------
PE    :      1
Image:
PI    :      1
Board: ACC_T1
------ ------
PO  1:     UTP
    2:     X25
    3:     X25
    4:     X25
    5:     X25
    6:     X25
    7:     X25
    8:     X25
    9:     X25
   10:     X25
```

Example 2: Displaying the services of a selected PM for a specific activation date

To display all the PEs, PIs, ports and service types configured for PM R34, using the NRS data file with the activation date closest to but not exceeding 990510 (May 10, 1999), type:

```
nrsdserv R34 990510
```

The following is produced:

```
PM: R34         NAMSID: 4034        Bundle: 99022903


------ ------ ------ ------ ------ ------ ------ ------
PE   :      1      2      3      4      5      6      7
Image: RMOFF3  RAUTP HGSERV RMSERV NCS386 NCSLOA    X32
PI   :      1      2      -      -      -      6      7
Board: V35_SC  DM_A1      -      -      - V24_4P V24_4P
------ ------ ------ ------ ------ ------ ------ ------
PO  1:      -    UTP      -      -      -    X25    X32
    2:      -    UTP      -      -      -    X25    X32
    3:      -      -      -      -      -    X25    X32
    4:      -      -      -      -      -    X25    X32
```

Appendix A Configuring NRS

This section provides a typical scenario on how to configure NRS when the NRS database has to be accessible by multiple workstations.

In this example, the NRS database reside on NMS1 in the directory /opt/MagellanNMS/data/nrs/data and is accessible in read/write mode from NMS2 and NMS3.

On NMS1, the directories /opt/MagellanNMS/data/nrs/data and /opt/MagellanNMS/data/nrs/rdf must be exported so that they can be mounted by NMS2 and NMS3. This action makes the NRS database visible by all three workstations and ensures that the same set of RDFs are used for all users and for all workstations looking at the NRS database.

From the root user, use vi to modify or create the file /etc/dfs/dfstab. Add the following lines:

```
share -F nfs -o rw=NMS2:NMS3 -d “NRS data directory”
/opt/MagellanNMS/data/nrs/data

share -F nfs -o rw=NMS2:NMS3 -d “NRS RDF directory”
/opt/MagellanNMS/data/nrs/rdf
```

These commands export both directories in read/write mode for NMS2 and NMS3. Execute the following command to make the change effective:

```
shareall
```

Note: If file /etc/dfs/dfstab did not exist before, you need to reboot the workstation. Or, execute the following commands as the root user: /usr/lib/nfs/nfsd -a 16 and /usr/lib/nfs/mountd. These commands are only executed at reboot time when the file /etc/dfs/dfstab exists.

On NMS2 and NMS3, the directories /opt/MagellanNMS/data/nrs/data and /opt/MagellanNMS/data/nrs/rdf must be mounted from NMS1 and called by the same name.

From the root user, use vi to modify the file /etc/vfstab. Add the following lines:

```
NMS1:/opt/MagellanNMS/data/nrs/data - /opt
/MagellanNMS/data/nrs/data nfs 0 yes rw
```

```
NMS1:/opt/MagellanNMS/data/nrs/rdf - /opt
/MagellanNMS/data/nrs/rdf nfs 0 yes rw
```

Execute the following commands to make the change effective:

```
mount /opt/MagellanNMS/data/nrs/data
mount /opt/MagellanNMS/data/nrs/rdf
```

Appendix B
Configuring the Service Integrity Audit

This section provides a typical scenario on how to configure the Service Integrity Audit.

In this example, the network is operating with 200 PMs (10 RMs, 70 AMs, 120 MASs) and 20 EMs. Changes are being made during the week and the new MCFs/view files are being activated on the following Sunday. The Dated mode is being used with next Sunday's date. In the case where a quick patch requires the use of the Configuration toolset, the MCF is named with a key.

The network is not referencing DNAs, RIDs or IP addresses outside the network. The NCD database NCDDB is already configured and is to be updated by the Service Integrity Audit tool.

There are three workstations (NMS1, NMS2 and NMS3) that can be used to increase the speed of the tool. These workstations are connected on the same LAN. Users *user1*, *user2* and *user3* will be used on their respective workstations. The tool will be started as a cron job from the NMS1 workstation at 10:00 p.m. every night. If required, the tool may be started from the toolset interactively. NRS is not configured yet.

Refer to the following sections that describe the steps you need to follow:

- "Configure NRS" (page 316)
- "Configure the Service Integrity Audit directories" (page 316)
- "Modify the configuration file" (page 316)
- "Create the instances file" (page 319)

- “Verification” (page 320)
- “Create a cron job” (page 320)
- “Monitoring” (page 321)

Configure NRS

The first thing to do is to configure NRS on the three workstations. Follow the procedure described in “Configuring NRS” (page 313). This procedure will make the NRS data (*/opt/MagellanNMS/data/nrs/data)* and rdf (*/opt/MagellanNMS/data/nrs/rdf*) directories available from NMS2 and NMS3.

Configure the Service Integrity Audit directories

Next, the Service Integrity Audit data (/opt/MagellanNMS/data/sis/data) and Service Integrity Audit work (/opt/MagellanNMS/data/sis/work) directories need to be accessible from NMS2 and NMS3. To do this, follow the steps described in “Configuring NRS” (page 313) but perform them for the Service Integrity Audit directories.

Modify the configuration file

Next, modify the */opt/MagellanNMS/cfg/SIS.cfg* file using *vi* to set the following variables:

- SIS_DATA_DIR

  The value is /opt/MagellanNMS/data/sis/data.

- SIS_WORK_DIR

  The value is /opt/MagellanNMS/data/sis/work.

- SIS_MCF_DETERMINATION_MODE

  The value is 2. This mode is used because the Configuration toolset is used and all changes are saved using the DATED naming convention.

- SIS_NAS_MCF_MODE

  This variable is no longer used and should be set to 0.

- SIS_NMS_MCF_MODE

  The value is 2. Provisioning using weekly mode.

- SIS_MERGED_MCF_MODE

  This variable is no longer used and should be set to 0.

- SIS_NMS_MCF_DAY

  The value is -7. It corresponds to Sunday.

- SIS_VIEWFILE_DETERMINATION_MODE

  The value is 2. This mode is used because the Configuration toolset is used and all changes are saved using the DATED naming convention.

- SIS_NMS_VIEWFILE_MODE

  The value is 2. Since the customer is provisioning using a weekly mode, mode 2 can be used.

- SIS_NMS_VIEWFILE_DAY

  The value is -7. It corresponds to Sunday.

- SIS_NSIC_EXTERNAL_DNA_FILE

  This variable is left blank. DNAs outside the network are not being referenced, therefore, NSIC and NCDPOP do not need a list of external DNAs.

- SIS_NSIC_EXTERNAL_RID_FILE

  This variable is left blank. RIDs outside the network are not being referenced, therefore, NSIC and NCDPOP do not need a list of external RIDs.

- SIS_NSIC_EXTERNAL_IPADDR_FILE

  This variable is left blank. IP addresses outside the network are not being referenced, therefore NSIC and NCDPOP do not need a list of external IP addresses.

- SIS_NCD_TARGET

  This variable is set to NCDDB.

- SIS_OUTPUT_FILES_KEEP_COUNT

  The value is 7. *sisauto* is being executed daily, therefore, 7 days worth of output files are kept in the SIS_DATA_DIR directory.

- SIS_NCS_DESTMNEM_DEFAULT

  This variable is left blank. Since it is only used by the sisautui tool, there is no need to specify a value here unless we plan to use sisautui. Otherwise, this variable is set to the NCS destination mnemonic used to connect to when starting SIS interactively.

- SIS_NCS_CAPID_DEFAULT

  This variable is left blank. Since it is only used by the sisautui tool, there is no need to specify a value here unless we plan to use sisautui. Otherwise, this variable is set to the capability id used to connect to when starting SIS interactively.

- SIS_PM_FILE_DEFAULT

  The variable is left blank. Since it is only used by the sisautui tool, there is no need to specify a value here unless you plan to use sisautui.

- SIS_AUTH_GROUP_DEFAULT

  The variable is left blank. Since it is only used by the sisautui tool, there is no need to specify a value here unless you plan to use sisautui.

- SIS_AUTH_USERID_DEFAULT

  The variable is left blank. Since it is only used by the sisautui tool, there is no need to specify a value here unless we plan to use sisautui.

- SIS_EM_FILE_DEFAULT

  The variable is left blank. Since it is only used by the sisautui tool, there is no need to specify a value here unless we plan to use sisautui.

Create the instances file

In the */opt/MagellanNMS/data/sis/work* directory, using *vi* define the files, *.mcflist.instances*, *.mcfmerge.instances*, *.nrspop.instances*, .viewfilelist.instances and .pnrspop.instances.

In file */opt/MagellanNMS/data/sis/work/.mcflist.instances*, there will be 12 parallel processes performing the MCF List; four on each workstation. The file contains the following:

```
localhost
localhost
localhost
localhost
NMS2 user2
NMS2 user2
NMS2 user2
NMS2 user2
NMS3 user3
NMS3 user3
NMS3 user3
NMS3 user3
```

In file */opt/MagellanNMS/data/sis/work/.mcfmerge.instances*, there will be one process even though MCF merges are no longer performed. The file contains the following:

```
localhost
```

In file */opt/MagellanNMS/data/sis/work/.nrspop.instances*, there will be three parallel processes performing the DPN NRS Populations: one on each workstation. The file contains the following:

```
localhost
NMS2 user2
NMS3 user3
```

In file */opt/MagellanNMS/data/sis/*work/.viewfilelist.instances, there will be three parallel processes performing the View File List; one on each workstation. The file contains the following:

```
localhost
NMS2 user2
NMS3 user3
```

In file */opt/MagellanNMS/data/sis/*work/.pnrspop.instances, there will be three parallel processes performing the Passport NRS Populations; one on each workstation. The file contains the following:

```
localhost
NMS2 user2
NMS3 user3
```

Having multiple MCF List, DPN NRS Population, View File List, and Passport NRS Population processes will increase the speed of the tool. The proper number depends on multiple factors, for example, number of PMs/ EMs in the network, size of these PMs/EMs, frequency of service data changes, number of workstations, workstation model, memory, and other processes running at the same time.

Verification

Next, verify that *user1* on NMS1 can execute commands on NMS2 as *user2* and NMS3 as *user3* without requiring a password. One way to find out if this is already possible is to try to remote login to NMS2 from NMS1 (rlogin to NMS2). If this works without having to specify a password, there is no need to configure it. If a password is required perform the following.

On NMS2, in the home directory of *user2* and on NMS3 in the home directory of *user3*, create a file called *.rhosts*. This file should contain the following line:

```
NMS1 user1
```

This means that *user1* on NMS1 can login on NMS2 as *user2* and NMS3 as *user3* without entering a password.

To verify that the configuration is correct on NMS1, the tool should be started interactively from the toolset with a few PMs and a few EMs.

Create a cron job

Finally, when the configuration is validated, call the tool from a cron job using the *all PMs and all EMs* mode. The *siscron* tool can be used to run the cron job. It supports all the options supported by the *sisauto* tool. The output generated by *sisauto* is saved in a file called *sisauto.<date>.messages*. This file is generated in the SIS_DATA_DIR directory as specified in the Service Integrity Audit configuration file.

Using the *crontab -e* command, add the following line in order to call the *siscron* script every night at 10:00 p.m. Replace <mdi> <id> <pw> with your specific NCS values and **<grp> <uid> <pwd>** with actual Passport authentication values.

```
0 22 * * * /opt/MagellanNMS/bin/siscron -ncs <mdi>
<id> <pw> -auth <grp> <uid> <pwd>
```

The configuration is now complete.

Monitoring

Once the tool is running daily from cron, the files *sisauto.<date>.messages* and, if necessary, *sisauto.<date>.log*, sisauto.<date>.ncdpop and *sisauto.<date>.nsic_sorted* should be looked at daily to monitor the integrity of the service data in the network.

Appendix C
Service data time management

Time is an important factor when working with service data. Components and their associations exist for specific time periods. The time dimensions of service data cannot be ignored, otherwise incorrect component associations or linkages can result, causing reporting errors. Basic time stream management is accomplished using the Configuration facilities for managing MCFs. This is achieved through the User Preferences options of Upload and Download which allows the user to select the mode for uploading and downloading MCFs. The Network Reporting System (NRS) does not incorporate any additional time management mechanisms.

The Configuration User Preferences mode indicates the method used to find MCF names for uploading and downloading. It is recommended that the dated mode be used. The NRS tools all make use of the flexibility of this mode by allowing selections like closest to a certain date. This mode allows easy network-wide reporting by finding the relevant view of each module at a given date. The tools do not preclude the use of other User Preferences modes but will not provide the same flexibility.

Appendix D Regular expressions

The table “Metacharacters” (page 325) lists the metacharacters that can be used when using regular expressions in NRS. The main use of these regular expressions is when using the NRS Module Filename Lister (*nrslist*) or the NRS Reporter (*nrsrpt*) tools.

Table 26
Metacharacters

Character	Description
^	Beginning of a field.
	nrsrpt -bundle ‘^WK’
	This matches all bundles that being with WK. It matches WK1301, WKABCD and so on.
$	Indicates the end of a field
	nrsrpt -bundle ‘4$’
	This matches all bundles that end with a 4. It matches WK1304, 4034, but not MYBUN41.
.	Matches any single character (except the new-line).
	nrsrpt -bundle ‘^....$’
	This matches all bundles that have four characters. It matches ABCD, 4034, and so on.
(Sheet 1 of 3)	

Table 26 (continued)
Metacharacters

Character	Description
|	Implies “or”. nrsrpt -bundle ‘^MYBUNDLE$\|^513$’ This matches bundles MYBUNDLE or 513.
*	Indicates zero or more repetitions of a character. For example, /ab*c matches abc, abbc, abbbc, and so on. It also matches ac (zero repetitions of b). The asterisk is more frequently used in conjunction with the period (.*). It is used to match an arbitrary string of zero or more characters nrsrtp -bundle ‘^A.*Z$‘ This matches all bundles that begin with A and end with Z. It matches AZ, AXZ, AX9Z, etc.
+	Similar to the asterisk, but stands for one or more repetitions of a string. For example, /ab+c/ matches abc, abbc, and so on; but it does not match ac. nrsrpt -bundle ‘^A.+Z’ This matches all bundles that begin with A, end with Z and are at least three characters in length. It will match AXZ, AX9Z, etc.
?	Similar to the asterisk, but stands for zero or one repetitions of a string. For example, /ab?c/ matches ac and abc, but not abbc and so on. nrsrpt -bundle ‘^A.?Z$‘ This matches all bundles that begin with A, end with Z and are two or three characters in length. It will match AZ, AXZ, but not AX9Z.
(Sheet 2 of 3)	

Table 26 (continued)
Metacharacters

Character	Description
[X]	Matches any one of the set of characters X given inside the brackets. nrsrpt -bundle ‘^WK[0-9][0-9][0-9][0-9]$‘ This matches all bundles that start with WK, followed by four digits. It will match WK4032 but not WK40AB or WK410.
[^X]	Matches any one character that is not in the set X that follows the circumflex (^). nrsrpt -bundle ‘^[^W][^K]‘ This matches all bundles that do not start with WK.
(X)	Matches anything that the regular expression X does. Parentheses are used to control the way in which other special characters behave. For example, the asterisk (*) normally applies to the single character that immediately precedes it. For example, /abc*d/ matches abd, abcd, abccd, and so on. However, /a(bc)*d/ matches ad, abcd, abcbcd, and so on. nrsrpt -bundle ‘(01)+‘ This matches all bundles with 01 in it. It will match 01, WK0119, 92010107, etc.
(Sheet 3 of 3)	

Index

Preside Multiservice Data Manager

Network Reporting System

User Guide

Release R14.3

Copyright © 2003 Nortel Networks.
All Rights Reserved.

NORTEL, NORTEL NETWORKS, the globemark design, the NORTEL NETWORKS corporate logo, PRESIDE, and PASSPORT are trademarks of Nortel Networks. OPENVIEW is a trademark of Hewlett-Packard Company. UNIX is a trademark licensed exclusively through X/Open Company Ltd.

Publication: 241-6001-022
Document status: Standard
Document version: 14.3RSUP
Document date: December 2003
Printed in Canada