



Preside Multiservice Data Manager

# Network Model

Administrator Guide

241-6001-015



---

Preside Multiservice Data Manager

# **Network Model**

## Administrator Guide

---

Publication: 241-6001-015

Document status: Standard

Document version: 15.1RSUP

Document date: August 2004

---

Copyright © 2004 Nortel Networks.

All Rights Reserved.

Printed in Canada

NORTEL, NORTEL NETWORKS, the globemark design, the NORTEL NETWORKS corporate logo, PRESIDE, DPN, and PASSPORT are trademarks of Nortel Networks. SUN and SUNLINK are trademarks of Sun Microsystems Inc. SPARCSTATION is a trademark of Sparc International Inc. UNIX is a trademark licensed exclusively through X/Open Company Ltd.

---



## Publication history

---

### August 2004

15.1RSUP Standard

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



---

# Contents

---

<b>About this document</b>	<b>13</b>
Who should read this document and why	13
What you need to know	13
How this document is organized	13
What's new in this document	14
Text conventions	14
Related documents	15
<hr/>	
<b>Chapter 1</b>	
<b>Introducing the Network Model</b>	<b>17</b>
Purpose of the Network Model	17
Network Model elements	18
Network Model schema	18
Components and attributes	20
Component naming	20
Component classes	21
Organization structure types	24
DEFAULT/ALL organization	24
New Components organizational nodes	25
Excluded organizational nodes	25
Dial Backup, Dialup Network and BWOD links	25
Network Model states	26
Criticality	26
Raw or external state	27
Propagation	28
Unknown state	29

- Constant state components 30
- State propagation rules 31
- Acknowledgment level 32
- Acknowledgment versus maintenance 33
- Network Model storage and life cycle 34
  - Active model 35
  - Saved model 36
  - Startup (committed) model 37
  - Configuration Data files 37
- Network Model tools and servers 38
  - Network Model coordinator (NMC) 39
  - Surveillance Network Updater (SURNUP) 40
  - Network Model Server (NMServer) 43
  - Network Model Editing Server (EdServer) 44
  - MDM tools and APIs that interact with the Network Model 44
  - Network Viewer in edit mode 45
  - MCDF/SNMPMCDF/Auto-MCDF 46
  - Network Model macros 47
  - Shared Memory Utilization tools 48
  - GMDR 48
  - Fault management tools 48
  - Configuration management tools 48

---

## **Chapter 2**

### **Network Model Administration**

**49**

- Network Model administrator's role 50
- Network Model maintenance tasks 50
  - Network Model creation 52
  - Network Model maintenance 54
  - Network Model deployment 57
  - Stand Alone deployment 57
  - Multiple deployment 57
  - Service Selected deployment 58
  - Deploying Regionalized Models 58

---

<b>Chapter 3</b>	
<b>Using Network Viewer to build a Network Model</b>	<b>59</b>
Introduction	60
Operations Performed by Network Model Editing	60
Entering edit mode to enable editing	63
Creating a new Network Model	64
Loading a Network Model	65
Creating an organization	66
Attaching a bitmap to an organization	68
Collecting Network Model data	69
Collecting data for module groups	69
Managing collections and applying collected data to the Network model	72
Managing Collections	72
Applying Collection results to the Network Model	73
Creating and editing components (nodes) manually	74
Manually creating components and subcomponents	75
Editing components and subcomponents	77
Using drag and drop techniques to edit nodes	80
Creating and editing links	82
Using menu commands to create and edit links	82
Using drag and drop to edit links	86
Deleting elements from the Network Model	87
Assigning modules to sites and sites to regions	88
Excluding components from an organization	92
Moving new components into sites	93
Positioning nodes	94
Positioning nodes	94
Positioning multiple nodes	95
Using the cluster and pin options	95
Positioning a node to its shared position	95
Reverting nodes to their current Network Model location	96
Adding and removing link bendpoints	96
Accessing the bendpoint menu	97
Automatically spreading overlapping links	97

- Saving components in collection files 97
  - Saving and distributing Network Model files 99
  - Quitting edit mode 100
    - Quitting Network Viewer while in edit mode 100
  - Using drag-and-drop features 101
  - Using the Network Viewer Edit icon bar 101
- 

## **Chapter 4**

### **Using the MCDF and SNMPMCDF utilities to build a Network Model 103**

- Collecting component data from the network 104
    - Collecting network component information 105
    - Collecting Passport 4400 component information 106
    - Limitations of MCDF, SNMPMCDF, and Auto-MCDF 106
    - Using the MCDUTIL utility for DPN networks 107
    - Creating configuration data files 110
    - Using the auto MCDF utility for DPN networks 117
- 

## **Chapter 5**

### **Network model macros and utilities 121**

- Available macros and utilities 122
  - Activating a network model 122
    - Activating a network model using the Network View model editor 123
    - Activating a Network Model using the makecurrent macro 123
  - Committing a network model 124
    - Committing a network model using the Network View edit mode 124
    - Committing a network model using the commitmodel macro 124
  - Displaying the committed and active model 125
  - Displaying available network models 125
  - Saving a network model in portable format 125
  - Saving a network model in quick format 126
  - Saving selected components and subcomponents in portable format 126
  - Converting between quick and portable formats 128
-

Purging unknown components from the network model	129
Putting components into maintenance state	131
Putting components into acknowledged state	132
Copying a network model fileset to other workstations	132
Setting up the remote workstations	133
Using tapes to transfer files between workstations	134

---

## **Appendix A**

### **Customizing the Network Model Schema 135**

Network model Types files	135
Customizing the network model schema	136
The Attribute Definition File	139
The Module/Subcomponent Definition File	141
The Link Definition File	146
The Organization Definition File	147
Customizing the Network Viewer	149
Creating custom node icon pixmaps	150

---

### **Index 153**



## About this document

---

The following topics are discussed in this section:

- “Who should read this document and why” (page 13)
- “What you need to know” (page 13)
- “How this document is organized” (page 13)
- “What’s new in this document” (page 14)
- “Text conventions” (page 14)
- “Related documents” (page 15)

### Who should read this document and why

This document is for system administrators and network operators who are responsible for building and managing network models. This document describes a network model, provides information about network model administration tasks, and details how to build a network model.

### What you need to know

This document assumes that you know the requirements and services of your own networks, and possess a working knowledge of Nortel Networks products, deployment, and network management software.

### How this document is organized

241-6001-015 *Preside MDM Network Model Administrator Guide* contains the following sections:

- “Introducing the Network Model” (page 17) explains what the Network Model is and what applications use it.

- “Network Model Administration” (page 49) provides information about Network Model administration tasks.
- “Using Network Viewer to build a Network Model” (page 59) shows how to build and edit a Network Model using Network Viewer edit capabilities.
- “Using the MCDF and SNMPMCDF utilities to build a Network Model” (page 103) explains how Nortel Networks products that support these utilities use the Make Configuration Data File (MCDF/Auto-MCDF) utilities.
- “Network model macros and utilities” (page 121) tells you how to use other Network Model macros and utilities.
- “Customizing the Network Model Schema” (page 135) defines what component types are allowed to be modelled.

## What’s new in this document

Support for MPE 9500 has been added to this document in this release.

## Text conventions

The following text conventions may appear in this document:

- `nonproportional spaced plain type`  
Nonproportional spaced plain type represents system generated text or text that appears on your screen.
- **nonproportional spaced bold type**  
Nonproportional spaced bold type represents words that you should type or that you should select on the screen.
- *italics*  
Statements that appear in italics in a procedure explain the results of a particular step and appear immediately following the step.

Words that appear in italics in text are for naming.

- `[optional_parameter]`  
Words in square brackets represent optional parameters. The command can be entered with or without the words in the square brackets.
- `<general_term>`  
Words in angle brackets represent variables which are to be replaced with specific values.
- UPPERCASE, lowercase  
Uppercase and lowercase letters that appear in UNIX commands and parameters must be matched exactly. The system matches upper and lowercase characters differently.
- |  
This symbol separates items from which you may select one; for example, ON|OFF indicates that you may specify ON or OFF. If you do not make a choice, a default ON is assumed.
- ...  
Three dots in a command indicate that the parameter may be repeated more than once in succession.

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

## Related documents

See the following documents for related information:

- 241-6001-011 *Preside MDM Fault Management User Guide*
- 241-6001-100 *Preside MDM Installation*
- 241-6001-101 *Preside MDM Engineering Guide*
- 241-6001-200 *Preside MDM Application Programming Interface Primer*

- 241-6001-201 *Preside MDM Network Model API Reference Guide*
- 241-6001-301 *Preside MDM Customization Administrator Guide*
- 241-6001-303 *Preside MDM Administrator Guide*

# Chapter 1

## Introducing the Network Model

---

The Network Model stores a modeled view of the managed elements in a network and makes it available to Preside Multiservice Data Manager (MDM) applications and to Application Programming Interfaces (API). The Network Model provides applications with access to network topology, as well as state and configuration data. Element states calculated from network information and propagated throughout the network play a crucial role in MDM surveillance.

The Network Model is a database of your network's components, and will need to be created, deployed and maintained. You can create a Network Model to use surveillance tools which show the states and arrangements of network components. The Network Model is used to prepare accurate and efficient representations of the network, to distribute workloads among network operators and for planning purposes.

### Purpose of the Network Model

The Network Model is used for the following purposes:

- state-based surveillance
- diagnostic information
- API access

#### **State-based surveillance**

Network operators use Preside Multiservice Data Manager monitoring tools to determine the state of network elements to manage faults in the network. Network Model topology information is used to add information about component states and their possible impacts through state propagation

(Surveillance Network Updater). The Network Model helps operators evaluate the extent and impact of faults by actually showing their effect on other components related to the faulty one. For example, if a link is brought down due to a failure in one of its endpoint components, the link is shown as down.

### **Diagnostic information**

Network Model components hold selected information describing various characteristics of each component, for example, shelf or PI types. Operators use this data to interpret and isolate faults and to display performance information.

### **API access**

The topology, diagnostic and state information is available to custom applications through the Network Model API Provider. For more information refer to 241-6001-201 *Preside MDM Network Model API Reference Guide*.

## **Network Model elements**

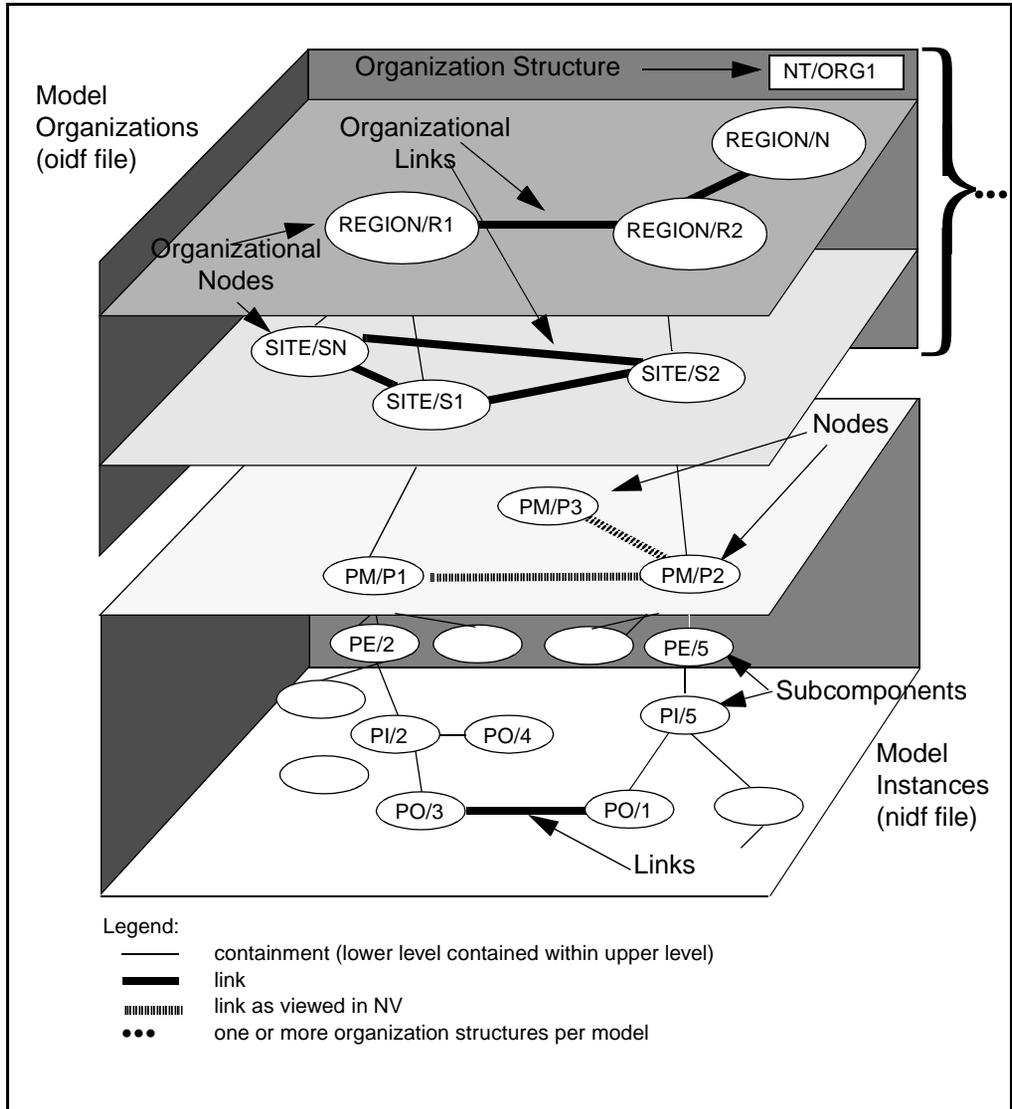
The network model elements are described in the following sections.

### **Network Model schema**

The Network Model is a specialized database designed to store and manage network components. Only certain types of components are admitted to the model. The Network Model schema, a set of ASCII files collectively called *types* files, determine which components can be modeled. The schema define the types of nodes, subcomponents and links that can be created; what attributes they bear (and their legal values); and how they can be arranged into structured organizations. The Network Model is built to store and manipulate components that are hierarchically structured (for example, sites contain nodes, nodes contain subcomponents), interconnected in a network topology (through links or trunks) and logically associated (through Uses/Used\_by attributes). The Network Model schema can be customized. Refer to “Customizing the Network Model Schema” (page 135) for details.

The figure “Typical Network Model component structure” (page 19) shows an organization structure for a typical network consisting of regions, sites, nodes, components and subcomponents. The following sections describe these elements in detail.

**Figure 1**  
**Typical Network Model component structure**



## Components and attributes

The Network Model is a collection of module, link, and subcomponent elements assembled into a hierarchical network topology. Actual components are organized into a number of multi-level organizational structures for the Network Model. Organizing components reduces the number of elements operators have to deal with at one time. All components in the Network Model have a name, a state and a criticality. Some may also hold additional informational attributes. The sections below describe Network Model elements in detail.

## Component naming

You need to know the naming system the Network Model uses, since a component's name determines whether it can be admitted to the Network Model (component names also indicate component types). *Internal* Component IDs in the Network Model are ASCII sequences of category-value pairs separated by spaces (for example, *<category> <value> <category> <value>...* ). The category element defines the type or class of component (for example, PE for a processing element). The value element names the actual instance of this type (for example, PE 2 is the second PE in a module).

The first pair in a component ID identifies the module or organization node type and name. *PM TOTO* indicates that the component is a DPN-100 packet module named *TOTO*. The subsequent pairs in the component ID identify the subcomponents. *PM TOTO PE 1* identifies *PE 1* in the DPN-100 module named *TOTO*.

Component IDs are usually displayed with a slash ('/') between category and value. This is the *display* name. Contrary to the *display* name, the component ID always has paired elements. If a component does not have an instance value (such as a SHELF component), the special value \$ is used. The *display* name *EM/PASSP SHELF CARD/2* actually maps to the component ID *EM PASSP SHELF \$ CARD 2*.

Component IDs for links identify the link type and both endpoints (themselves component IDs as just described). The link *display* name *<-NL-> PM/E1 PE/2 PI/2 PO/1 <-> PM/E2 PE/5 PI/5 PO/2* maps to the component ID *NL:PM E1 PE 2 PI 2 PO 1:PM E2 PE 5 PI 5 PO 2*.

**Note:** Link endpoints are ordered as follows: the lowest endpoint component ID in alphanumeric order is first, *unless* one of the endpoints is a module in which case the module endpoint is always second.

A component ID must be unique throughout the Network Model, whatever and however many organization structures are used.

## Component classes

The classes of components modeled by the Network Model include:

- *Modules or nodes* which usually correspond to the largest physical unit of the component.
- *Subcomponents* which are the individual sub-elements of the module. They can be physical (hardware) or logical, for example, software processes.
- *Links* which represent physical (and sometimes logical) links or trunks between network components.
- *Organization Structures* which represent specific organizational arrangements of modules.
- *Organizational Nodes* which group a set of modules or other organizational nodes in a fixed location of the Network Model hierarchy.
- *Organizational Links* which are used to group and represent links between components belonging to different organizational nodes.

All components are identified by their component ID and each component holds a set of core attributes, for example, its state and informational attributes. The Network Model Schema determine which informational attributes are present on the basis of a component's category type.

### Subcomponents

The sub-classes of subcomponents include:

- *Ordinary* subcomponents are the more important (crucial or backbone) components that make up the module. They support informational attributes to provide context on their use. These subcomponents can be created manually or auto-populated and are always present in the active Network Model.

- *Endpoint* subcomponents are also Ordinary but with the added feature of being legal termination points for links.
- *Dynamic* subcomponents are either less important from a state surveillance standpoint, or potentially too numerous for efficient resource usage. Nevertheless, they must be modeled to indicate faults which may affect them. Dynamic subcomponents are added to the model only when there is notice of a troubled state for them (out- of-service or troubled). As soon as their states revert to in-service or unknown, they are removed from the model. Since this state triggered population can only identify the name and state of the subcomponent, dynamic subcomponents do not support informational attributes. Further, Dynamic subcomponents cannot be created manually in the Network Model.
- *Optional* subcomponents are similar to Ordinary subcomponents in that they can be configured manually. If they are configured manually, they remain in the model whatever their state. However, if an Optional subcomponent is auto-populated by a state change notification (it was not previously in the Network Model), the subcomponent behaves as a Dynamic component and is automatically removed from the Network Model when its state reverts to In-Service or Unknown.
- *Optional Endpoint* subcomponents are similar to Optional subcomponents, but they can also terminate links like Endpoint subcomponents.

### **Links**

Links are used in the Network Model to represent trunk or access links between modules, for example, to a computing host. Though they are displayed between nodes in the network viewers, links are usually defined between module sub-components, for example, between POs in DPN-100 or TRK subcomponents of a Passport. Links come in different types which can be used to better identify their purpose and for filtering. A module component can terminate multiple links, for example, many links can connect the network to a host, but subcomponent endpoints can only support a single link at a time. For nodes and sub-components, links support a set of core attributes for their states and criticality and may hold a set of Informational Attributes (defined in the Network Model schema) such as the Gateway ID (GW) for Gateway links. Links and nodes support attributes (position for nodes and up

to ten bendpoints for links) describing how they will be displayed in Network Viewer. These positions are organizational structure specific. A node can appear in different places depending on which organization is being viewed.

### **Informational attributes**

Informational Attributes attach additional information to a component to help operators diagnose the nature and importance of a fault or simply know more about a component. Many components also support the INFO attribute, a free-form text used to add specific information of your own for that component. A special set of Informational Attributes are tagged as being *Uses/Used\_by* Attributes, for example, a FRAMER subcomponent's LINK\_TO\_FRAMER attribute. This value is a component ID naming another component in the model which uses or is used by that component. Knowledge of which component uses or is being used by another helps to determine the cause or impact of a troubled state.

### **Organizational structures, nodes and links**

Networks typically contain a large number of nodes and links. The Network Model lets you organize them into organizational structures to help you manage them. By grouping nodes and links that have certain common characteristics (for example, location, role, and customers) organizational structures let operators reduce the number of network elements being actively monitored and permit more focussed and effective surveillance. In fact, nodes can be left out of an Organization Structure if they are not relevant to the operator using that organization.

A Network Model can have multiple organizational structures. Its nodes and links may be found in some or all of these structures. Organizational structures are thus like overlays that can be applied to the Network Model to group or hide its nodes.

### **Organization levels, nodes and links**

The standard organization structure has three levels: *module*, *site* and *region*. The *actual* (also called physical) nodes and links of the Network Model together with the subcomponents are found at the *module* level. The *site* level contains site level Organizational Nodes which are groupings of module level nodes and links. Site level nodes and links can be grouped into *region* nodes in the same manner. In a Network Viewer session, a site or region node can be expanded to show the module level nodes and links it contains. The site or

region node then disappears and the *child* components are displayed instead. When a child node is *compressed*, it and all other nodes and links which are the children of the same parent site or region node are hidden and the parent is displayed.

If a link connects two modules within the same site, the link is part of that site. If, on the other hand, a link connects modules from two different sites, a site level organizational link is automatically created between these two Sites (if it does not already exist) and the module level link is assigned to it. The same rule is applied for site level links between sites of different regions.

Effectively, all organizations contain three levels and require organization nodes named REGION or SITE. The component IDs of the site and region nodes have a type and name just like *physical* nodes, for example, SITE/THE\_CITY or REGION/THE\_COUNTRY.

**Note:** The component ID of a site node does not contain that of its parent region, therefore, the name used to name a site or region must be unique across the entire model (that is, unique across all organizational structures).

## Organization structure types

Organization structures are represented as nodes in the model. Organization structures come in a variety of types. These types include DEFAULT, NT, DPN, and ILOOK. For example, the DEFAULT type organization can contain all legal nodes and links in the Network Model.

**Note:** In a previous version, the Network Model used to restrict which type of node and link types could be organized in a given type of organization structure. This has been removed for simplification. Now, all organization structure types can organize any node and link types.

### DEFAULT/ALL organization

One organization structure, DEFAULT/ALL, is automatically created and always exists in the Model. DEFAULT/ALL contains a single region (DEFAULT\_REGION/ALL) and this region a single site (DEFAULT\_SITE/ALL). DEFAULT\_SITE/ALL contains all *physical* nodes and links in the Model. DEFAULT/ALL is not suitable to be used for surveillance purposes; DEFAULT/ALL should only be considered as the buffer of all nodes in the Model.

## New Components organizational nodes

There are two specialized elements in an organization structure: the *NEW\_REGION/<orgname>* and *NEW\_SITE/<orgname>* organizational nodes. These nodes are present in every organization structure other than *DEFAULT/ALL*. They contain nodes dynamically created by SURNUP's auto-population mechanism, and they contain all valid modules and links when a new organization is created.

The *NEW\_SITE/<orgname>* acts as the default parent for manually created modules. Nodes found in *NEW\_SITE/<orgname>* should be properly organized as soon as possible. Each organization contains its own set of new components in the *NEW\_SITE* organizational nodes; you can include specific nodes in one organization without affecting other organizations.

*Note:* Auto-populated components are lost if not saved before a new model is loaded. See “Saved model” (page 36) for more information.

## Excluded organizational nodes

These nodes are automatically created to contain all nodes excluded from an organization structure. Excluded organizational nodes are called *EXCL\_REGION/<orgname>* and *EXCL\_SITE/<orgname>*. By default, these nodes are hidden from Network Viewer. Each organization contains its own set of excluded nodes; you can exclude specific nodes from one organization without affecting other organizations.

## Dial Backup, Dialup Network and BWOD links

The Network Model and Preside Multiservice Data Manager fault management tools support Dial Backup Network Links (DBNL), Dialup Network Links (DNL) and Bandwidth on Demand (BWOD). DBNLs, DNLs and BWODs are not populated up front in the network model. Instead, when a DBNL, DNL or BWOD is up, typically due to the failure or congestion of the main link or trunk, the network emits alarms of the event. These alarms are interpreted by the fault management tools as an indication that a new link should be created in the model and displayed. If the proper conditions apply, the tools will display the new link calling the attention of the operators. While the DBNLs, DNLs and BWODs are up, they are subject to the same state propagation rules as all other links. For more information, refer to “State propagation rules” (page 31). Once the DBNL/DNL/BWOD is disabled, the link is automatically removed from the model and from all displays.

**DPN exception to Dial Backup, Dial Network and BWOD links**

DPN-100 status records are ignored for DBNL/DNL/BWOD link ports, since they always indicate that the port is down when the link is not active.

## Network Model states

Every component in the Network Model contains a set of core attributes to model its state information. Some of these state values are provided by the Preside Multiservice Data Manager Surveillance Data Servers (GMDR, DMDR and FMDR), others are computed by the Surveillance Network Updater (SURNUP) through its State Propagation mechanisms and, finally, some are determined by operator actions. The state information of components is shown in the table “Component state information” (page 26).

**Table 1**  
**Component state information**

<b>Attribute</b>	<b>Purpose</b>
Criticality	Relative importance of the component
Raw or External State	State received from the network through the Surveillance Data Servers
Propagated State (and Criticality)	State computed by SURNUP
Acknowledgment Level	Level of State Acknowledgment or Maintenance

### Criticality

A component’s *criticality* is a number between 1 and 5 defining the importance of the component for network surveillance. The more outages the component’s fault would cause, the higher its criticality value. Criticalities are assigned statically to components. A default criticality is defined by the component type (for example, a PM’s criticality is 4, its PEs are 3, and their PIs and POs are 2). These defaults can be overridden in a Network Model edit session. For example, it is a good idea to assign Criticality 4 instead of 3 to DPN-100 Office PEs to make their faults more visible. The criticality of a component must be greater or equal to the highest criticality of its children

and lower or equal to that of its parent. In other words, the outage a parent can cause is greater than or at least equal to the disruption caused by its children. *Criticality 5* is usually used for Organizational Nodes and Links and *Criticality 4* for physical nodes and links. It is possible to assign *Criticality 5* to important physical nodes and links.

## Raw or external state

The *raw state* (or *external state* or state from the network) is the state computed by the Surveillance Data Servers (GMDR, DMDR and FMDR) for the component on the basis of network management information received for the component. Possible *raw state* values are shown in the table “Raw state values” (page 27).

**Table 2**  
**Raw state values**

Raw state	Meaning
Unknown	The component is out of contact and its state is unknown. See “Unknown state” (page 29).
In-Service (INSV)	The component is known to be In-Service.
Out-Of-Service (OOS)	The component is known to be Out-Of-Service.
In-Service-Troubled (ISTB)	The component is known to be In-Service but experiencing some difficulties, for example, overloaded.

The *raw state* is received for and applies to only a single component. For example, if a PE goes down, the network generates only one alarm for that PE component, though this failure means that of all that PE’s subcomponents are also down.

## Propagation

To make these dependencies visible and to permit surveillance of lower level components at a higher level (that is, monitoring a network at SITE level instead of looking at every single module and link), the Surveillance Network Updater (SURNUP) computes another state for components on the basis of the received raw states. This is the *propagated state*, and is the state displayed when using the fault management tools such as Network Viewer, Component Status Display or Component Information Viewer. The *propagated state* takes into consideration the possible faults (raw states) of related components and shows their impact. The *propagated state* consists of a pair of elements: a state value, for example, Unknown, In-Service or Out-Of-Service, and an optional state criticality or severity for In-Service-Troubled and Out-Of-Service, to identify the importance of the fault as a *Criticality* value.

**Note:** The propagated state of a component is not necessarily an indication of the true state of the real network element, but an indication of how much attention an operator should give to a troubled component. The worse the state, the quicker the operator should respond.

State Propagation is a complex mechanism that takes into consideration the raw state of the components, their criticalities, parentage and other information such as the Acknowledgment Level. Generally, when a Raw State is received for a component, the component's propagated state is re-evaluated. If it changes, the new state is used to determine the propagated state of its children (downwards) and parents (upwards) as appropriate.

A component's propagated state is usually determined by two factors: the worst state of its children, that is the most negative state (OOS or ISTB), and the highest criticality to determine the propagated state criticality. For example, a module that has no troubled raw state of its own but has troubled subcomponents displays a propagated state of In-Service-Troubled, as it has lost part of its services and displays a state severity whose value is the criticality of the most important subcomponent affected. The propagated state values are described in the table "Propagated state values" (page 29).

**Table 3**  
**Propagated state values**

Propagated State	Meaning
Unknown	The component state and that of its children is not known.
In-Service (INSV)	The component, its parents and all its children are In-Service.
Out-Of-Service-N (OOS-N)	The component is known to be out-of-service where N is its criticality.
Hierarchical-Out-Of-Service-N (OOS-N)	The component's parents or one of its grandparents is out-of-service where N is the component's criticality.
In-Service-Troubled-N (ISTB-N)	The component is troubled or one of its Sub-components is troubled or out-of-service where N is the criticality of the most important subcomponent affected, if not that of the component itself.
Maintenance (MTCE)	The component is in Maintenance State
Hierarchical-Maintenance (HIER_MTCE)	One of the component's parents is in Maintenance State.
Acknowledged (ACKED)	The component is in Acknowledged State.

### Unknown state

A Propagated State of *Unknown* indicates that no network management data has been received for the component, even though it is defined in the model.

At the subcomponent level (below modules) this is usually not a problem. State walks and active alarm lists ensure that if the component is really troubled, the Preside Multiservice Data Manager (MDM) is notified. At the subcomponent level, *Unknown* can safely be interpreted as indicating the component is trouble-free.

At the module level an propagated *Unknown* state indicates that the MDM has no current connection to the module and cannot report on its state. If the component is to be managed by this NMS, remedial actions must be taken to re-establish management access.

### **DPN *Unknown* state exceptions**

Due to OA device re-homing in a DPN-100 network, an *Unknown* state may not necessarily mean that a subcomponent is trouble-free.

## **Constant state components**

The Network Model supports constant state components, which are module components that do not receive any state information from the network.

Constant state components do not generate real states or alarms and always have a propagated state of in-service. Examples of constant state components are

- FNMODs for foreign network modules
- GW for gateways
- HOSTs for computing hosts

Constant state components can be used to make certain important trunk, link, or gateway connections visible at module level. For example, if the network supports a connection to a Mainframe host and that connection is important for network operations (for example, an NDI host), we recommended you add a host constant state component to the Network Model with a link (of type XL) to the proper connection endpoints on the network elements. The state of the port (and therefore that of the connection to the host) will be as visible as any other link or trunk in the Network Model.

## State propagation rules

*State Propagation* is a complex mechanism, and is crucial to state-based surveillance. Here are some of the most important rules for state propagation:

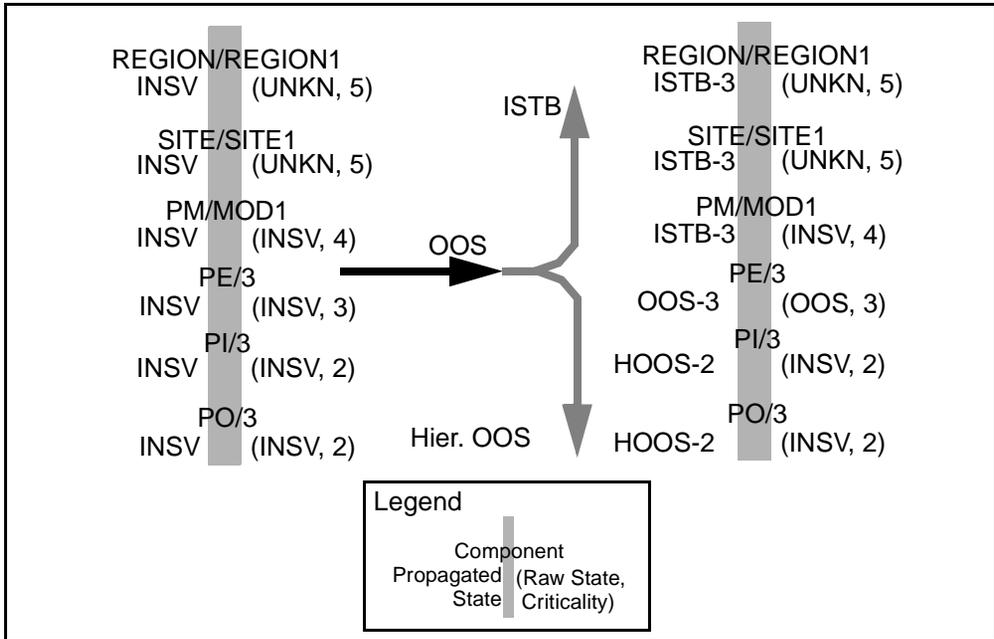
- The propagated state of a component reflects that of its parents and subcomponents.
- A state of *out-of-service* is propagated to subcomponents as *Hierarchical-out-of-service* and to the parents as *in-service-troubled-N* (where N is the criticality of the out-of-service component).
- A state of *in-service-troubled* is only propagated to the parents as *in-service-troubled*.
- A state of *Unknown* forces the raw and propagated states of the subcomponent to unknown.
- The state of a link is deduced from that of its endpoints. If at least one is *out-of-service*, the link is considered out-of-service.

### Example

#### Propagating an Out-Of-Service state from the Network

Suppose an Out-Of-Service raw state is received from PE 3. This OOS is propagated downwards to the PE subcomponents as Hierarchical Out-Of-Service (or OOS From Parent) - if a parent is down, its subcomponents are also down. The OOS is propagated upwards as In-Service-Troubled - if one of the module's subcomponents is down, the module is somewhat impaired. Note that the state criticality for the module, SITE and REGION (ISTB-3) reflects the component criticality of the affected PE (the worst state here since the PE is the only child of the module). See the figure "Example of State Propagation" (page 32).

**Figure 2**  
**Example of State Propagation**



### Acknowledgment level

The acknowledgment level state is used when a component's state has been acknowledged and should be hidden. Two forms of acknowledgments are supported by the Network Model: *Acknowledgment* and *Maintenance*. The Acknowledgment Level values are shown in the table "Acknowledgment Values" (page 33).

**Table 4**  
**Acknowledgment Values**

Acknowledgment Value	Meaning
ACK_NONE	Component state has not been acknowledged.
ACK_TEMP	The component has been put in Acknowledged state. This state will be kept until a new Raw State is received for the component.
ACK_PERM	The component and its subcomponents have been put in Maintenance state. These states will be kept until the Model is reloaded or the Maintenance state is removed.

## Acknowledgment versus maintenance

In a live network there are usually a certain number of components that are troubled with no effect on service or network performance. For example, unconfigured ports or components in the process of being fixed may not affect network performance.

If these components are left as is, most organization nodes in the Network Model eventually display In-Service-Troubled, thus making state propagation ineffective, that is, unable to identify new faults. This is widely known as the *yellow syndrome* due to the color associated with the In-Service-Troubled state in an old Network Display tool. To control this, the Network Model lets operators acknowledge a component fault in one of two ways: permanently, until a new model is loaded through the Maintenance State, and temporarily through the Acknowledged State.

In Maintenance State, the component's real Raw State and those of all its subcomponents are masked (not taken into account in state propagation). The component looks as though it is In-Service. All further raw states for the component and its children are ignored, but stored until the component is explicitly pulled out of Maintenance State by the operator or if the model is

reloaded. A component in Maintenance State exhibits a specific propagated state (Maintenance or Maintenance from Parent) to distinguish it from trouble-free components.

In Acknowledged state, the component's real raw state is masked from state propagation, but the raw state of its subcomponents is not masked. In state propagation, the component looks as though it has received an In-Service state from the network. However, since its children are unaffected by the acknowledgment, any fault they may have can be propagated and shown. This makes their hidden faults visible.

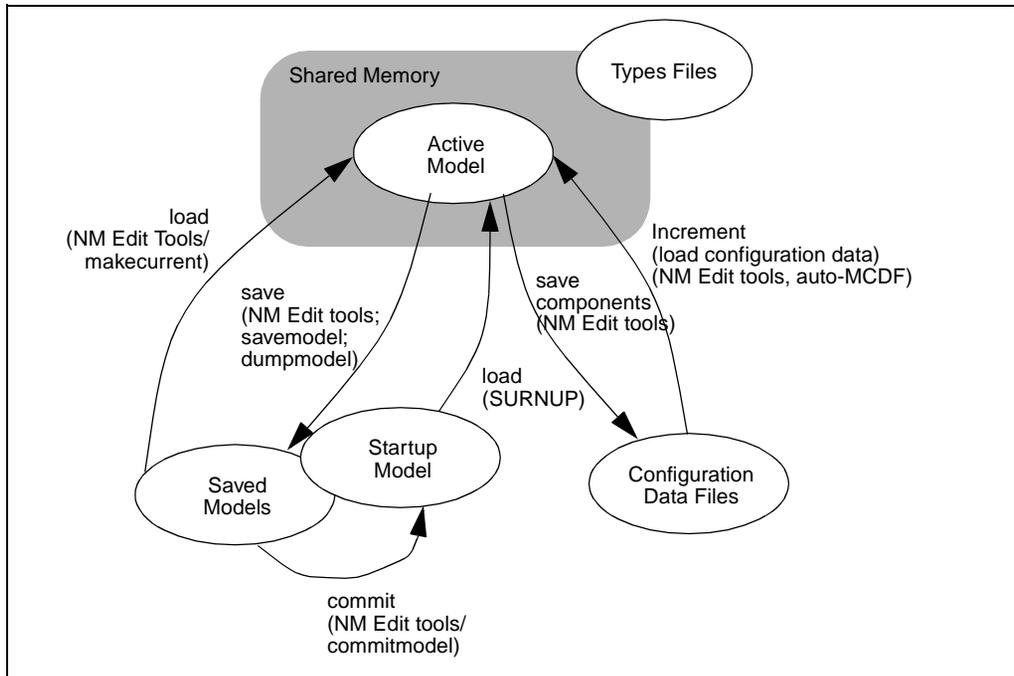
Contrary to Maintenance State, any further raw state change from the network for an acknowledged component causes the automatic removal of the acknowledgment and the propagation of the newly computed state. Since the real raw state is stored (masked), the acknowledgment can be removed by the operator. A component in Acknowledged State with no other apparent problems (for example, parents are not Out-Of-Service and children are not troubled), displays the Acknowledged state to distinguish it from truly trouble free components.

When all outstanding alarms of a component are acknowledged, the component is automatically put in acknowledged state. If an alarm is unacknowledged or a non-active alarm is resolved, the component's acknowledged state is automatically removed to indicate this change. For information on Alarm Acknowledgment, see 241-6001-011 *Preside MDM Fault Management User Guide*

## Network Model storage and life cycle

The Network Model database can exist in a number of different stages: active in shared memory, saved to disk files, configured to be the startup Network Model, and even broken into parts as Configuration Data Files used to update the contents of the Network Model. See the figure "Network Model life cycle" (page 35).

**Figure 3**  
**Network Model life cycle**



## Active model

The active model is the model currently loaded in shared memory. An active model is used for surveillance and by other tools including the NM API Provider. There can only be one active model at a time on a workstation. The active model is usually loaded from a saved model unless it is being initially created in a Network Model edit session. To edit a model, it must first be made active by loading it in shared memory (See “Activating a Network Model using the makecurrent macro” (page 123)). The name of the current active model can be found by invoking the *whichcurrent* macro (See “Displaying the committed and active model” (page 125)).

**Note:** To activate a Network Model, a shared memory segment of sufficient size must be configured in the workstation kernel. Refer to 241-6001-101 *Preside MDM Engineering Guide* for memory

requirements. The shared memory tool in the *Utilities* toolset and the */opt/MagellanNMS/bin/surmemusg* tool can be used to examine the amount of shared memory segment being used by the model.

## Saved model

A model is saved when it has been written out to disk after a Network Model edit session or by using the *savemodel* macro. Models are saved as a fileset directory containing the appropriate Model instance description files in */opt/MagellanNMS/data/model/nmf/<modelname>*. Three files are created there: *instances.nidf* contains the description of the model's nodes and subcomponents (including their attributes); *instances.lidf* contains the description of the 'physical' links and their attributes; and *instances.oidf* contains the description of its organization structures, organizational nodes and links, and their node assignments. There can be multiple saved model filesets in the */opt/MagellanNMS/data/model/nmf* directory. The names of the saved models available to you can be listed by invoking the *whichmodels* macro (see "Displaying available network models" (page 125)).

The model can also be saved in a non-portable binary format using the *dumpmodel* macro and as an *instances.image* file. Models in binary format are saved and loaded quickly (in seconds) which results in shorter surveillance outages. This feature makes quick format the best choice for temporary saves while editing the model and for the committed model. However, this format uses a larger amount of disk space (equivalent in size to the model in memory) and the binary files are not portable across workstations or Preside Multiservice Data Manager releases (distribution). Over time with multiple quick format saves, memory fragmentation will degrade performance.

It is recommended that you always keep both quick and portable format versions of your most important models. Always use portable format when you are integrating new devices into MDM. If you save in binary mode, the new network model files will not take effect.

The *popconv* program can be used to convert a binary format model to portable format (or vice versa) without affecting the active model or surveillance. See "Converting between quick and portable formats" (page 128).

To copy a saved model from one workstation to another, just copy its files. For more information refer to the remote workstation's */opt/MagellanNMS/data/model/nmf* directory.

**Note:** Do not copy the *instances.image* file if it is present. This is the fast load format file and it is not portable.

## Startup (committed) model

The startup model is the saved model that will automatically be loaded by *SURNUP* for surveillance purposes the next time the workstation reboots. Models can be made to be the startup model when they are saved in a Network Model edit session or by using the *commitmodel* macro. The startup model is usually the active model unless another one has been loaded since the workstation rebooted. The startup model is always accessed as */opt/MagellanNMS/data/model/commitmodel* which is a symbolic link usually to */opt/MagellanNMS/data/model/nmf/<modelname>*. The name of the current startup model can be found by invoking the *whichcurrent* macro.

We recommend you maintain the committed model in binary format, so that surveillance will resume faster after a reboot.

## Configuration Data files

One method for creating Configuration Data files (CDF) is to use one of the MCDF tools to describe specific components that should be populated in the Network Model. These files do not constitute a full model, but parts (typically a single module, its subcomponents and links) that should be updated. CDFs are loaded into an active model explicitly (load CDF) or implicitly (collect configuration data) in a Network Model edit session or through auto-MCDF.

**Note:** Make Configuration Data files (MCDF) is supported on Passport and DPN switches.

Another method for creating CDFs is to use the Save Components dialog of the Network Viewer Editor. By using this method, you can specify a list of modules, organizational nodes, or organizational structures to save. The resulting CDFs are then available for loading into another Network Model, allowing you to save and distribute sub-sets of the model.

CDFs, which are fileset directories, are found in */opt/MagellanNMS/data/model/cdf/<cdf name>*. They are stored in this file after a collect configuration data request during a Network Viewer edit session, or when created by MCDF, Auto-MCDF, or SNMPMCDF.

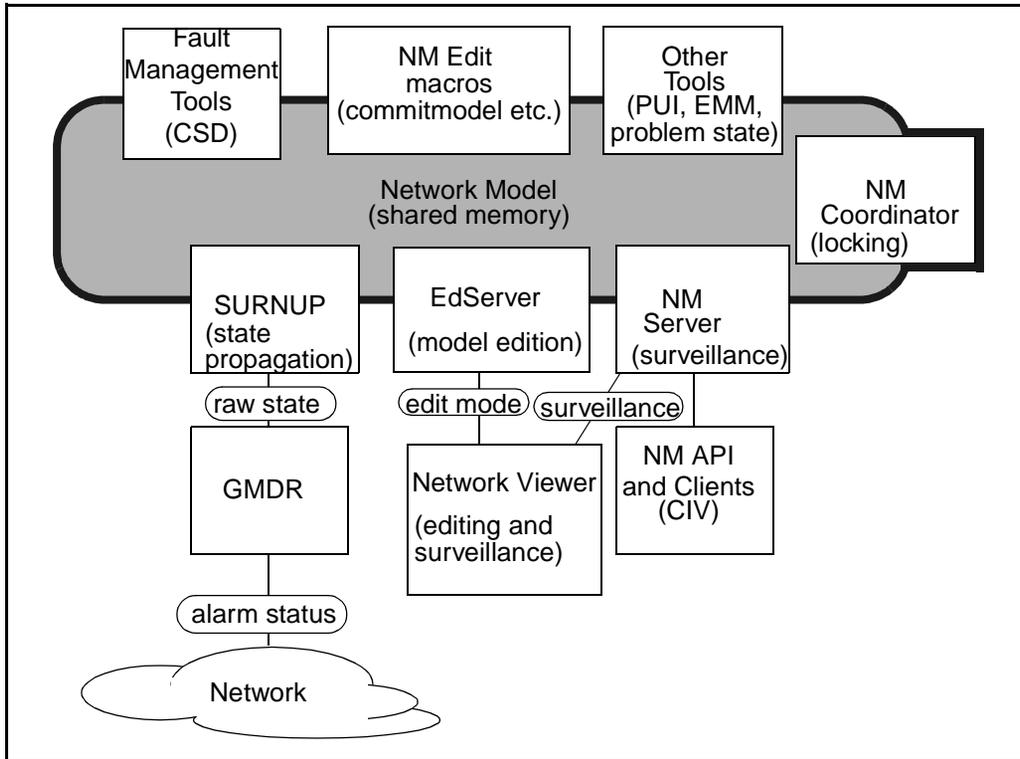
CDFs created by MCDF and SNMPMCDF must be loaded by hand in a Network Model edit session. CDFs created by Auto-MCDF are automatically loaded and destroyed once loaded in the active model. Other CDFs should be deleted using Network Model editing functions when no longer needed.

When CDFs are loaded in the model, the component information they contain is added (increment) to existing information (except for links on subcomponents which override existing links and attributes whose value override the existing one. Optionally, the CDF can be loaded in a ‘destructive’ manner such that, for every module described by the CDF, all subcomponents not defined in the CDF but found in the model are destroyed.

## Network Model tools and servers

For the Network Model tools and servers, see the figure “Network Model tools and servers” (page 39).

**Figure 4**  
**Network Model tools and servers**



### Network Model coordinator (NMC)

The first server needed to support the Network Model is the Network Model Coordinator (NMC). This process creates and owns the shared memory segment used by the Network Model. It provides locking capabilities so only one process can be used to edit the model. Finally, it provides for internal Network Model event broadcasting, for example, new components, so all Network Model clients are informed of its state. The NMC is automatically started and maintained by the Server Manager (SVM). Refer to *Network Model Coordinator (DNMNMNMC)* section in 241-6001-303 *Preside MDM Administrator Guide* and to *Appendix A* in 241-6001-100 *Preside MDM Installation*. If the NMC terminates, all other Network Model processes stop, since the loss of the shared memory segment is fatal.

## Surveillance Network Updater (SURNUP)

The Surveillance Network Updater (SURNUP) server implements state-based surveillance in the Network Model through state propagation.

- 1 The SURNUP server ensures that there is an active model, loading the startup model if necessary.
- 2 The SURNUP server registers to the General Management Data Router (GMDR) server and is informed of all raw state changes which it propagates as appropriate to the related components. The GMDR server informs SURNUP of all component deletions, which are processed according to the SURNUP option selected (see "SURNUP options" on page 41). The GMDR server also informs the SURNUP server whenever all active alarms of a component are acknowledged.
- 3 The SURNUP server notifies the surveillance tools of the raw and propagated state changes in the network, as well as other house-keeping events.
- 4 The SURNUP server implements *Maintenance* and *Acknowledge* states for components.
- 5 The SURNUP server performs auto-population of components, if they are not already present in the model.
- 6 The SURNUP server receives new *ackStateChange* notifications from the GMDR server when components become acknowledged or unacknowledged. For information on acknowledging/unacknowledging components, refer to 241-6001-011 *Preside MDM Fault Management User Guide*.
- 7 The SURNUP server performs auto-deletion that removes obsolete components from the network model (see "SURNUP options" on page 41).
- 8 The SURNUP server automatically deletes components from the network model when a disconnected GMDR is reconnected - this is referred to as resynchronization-based automatic component deletion.

The SURNUP server supports a number of operating options which are described in the 241-6001-310 *Preside MDM Server Reference Guide*. The SURNUP server is automatically started and maintained by SVM and it is through the SVM Administration tool that any operating options should be

changed. SURNUP's startup can be quite lengthy as it might have to load a model and then synchronize its states to the Surveillance Data Server's database.

### **SURNUP initialization**

The General Management Data Router (GMDR) server is used by the SURNUP server to extract raw state information for managed components. Since the SURNUP server obeys workstation wide service selection, the GMDR server used may be running in a remote server set Preside Multiservice Data Manager and need not be local to the Network Model (the `-H` option can be used to force the use of a specified host).

Upon initialization, the SURNUP server extracts a complete state picture from the GMDR server to synchronize the Model to the network reality. This initial query may be long enough to block other GMDR operations and can disrupt tools such as Component Information Viewer and alarm status API clients. This initial query is also performed every time the SURNUP server loses connection or intentionally disconnects from the GMDR server through LAN selection or Model loading.

### **SURNUP options**

The Surveillance Network Updater State Propagation can be controlled through command line arguments. The first of these arguments is the `-C` option. If specified, it directs SURNUP to automatically propagate in-service states downwards. The `-N` option is used to cause out-of-service states to be propagated to parent components as out-of-service instead of in-service-troubled. This helps to enhance fault visibility.

**Note:** The `-C` option automatically turns auto-population off. To avoid this, add `-A all` after `-C` or use the `-V` option instead of both options on the command line.

The SURNUP server is also responsible for the auto-population of the model. When raw states are received for any components not included in the model, the SURNUP server can create them automatically. Auto-populated components are partial, since only the name and raw state is known to the

SURNUP server and no informational attribute can be filled in. You can control the SURNUP server's auto-population through the *-A* command line argument whose values are:

- *-A none* indicates that no auto-population is to occur.
- *-A all* indicates that full Auto-Population is to occur (default)
- *-A backbone* indicates that only the most important module components, for example, PE, PI, LP, SHELF and CARD, are to be auto-populated. See "Customizing the Network Model Schema" (page 135) to see which components are considered backbone components.
- *-A subcomp* indicates that only subcomponents of existing modules are to be auto-populated
- *-A none* is useful when the Model is a relatively complete representation of a part of the real network that should be surveilled by preventing other components from intruding in the model (assuming they are being managed by another network management system). *-A subcomp* lets you examine only a subset of the real network, but it permits the Auto-Population of subcomponents. Using this option means the initial Model need not be so accurate or complete.

If a troubled subcomponent of an existing module does not exist in the model, it is forcibly added to the Model to make its faults visible, whatever the current Auto-Population mode.

**Note:** Auto-Population and the creation of troubled components are automatically turned off during a Network Model editing session or operation.

The SURNUP server also has an option to automatically delete obsolete components from the network model. When the SURNUP server receives component delete notifications from the GMDR, actions are taken based on the option selected. You can control the SURNUP server's automatic deletion through the *-D* command line argument whose values are:

- *-D n* indicates that no automatic deletion is to occur.
- *-D all* indicates that full deletion of obsolete components is to occur.

- *-D all, nl* indicates that full deletion of obsolete components is to occur. Links and their end-points are not deleted. The extent of deletion depends on the level of component deletion notification.
- *-D s* indicates that only subcomponents of existing modules are to be deleted. This is the default.
- *-D s, nl* indicates that only subcomponents of existing modules are to be deleted. Links and their end-points are not deleted. The extent of deletion depends on the level of component deletion notification.

Another way to control what information is manipulated by SURNUP is to use the Criticality Threshold filtering capability of GMDR. For details, see the section on General Management Data Router (GMDR) in 241-6001-310 *Preside MDM Server Reference Guide*. By specifying this threshold to SURNUP using the *-T* command line option, SURNUP is notified by GMDR of only those components whose GMDR Criticality is equal to or higher than the specified one. The GMDR Criticality Threshold is not to be mistaken with the Network Model Criticality; the GMDR Criticality range is between 0 and 255. This allows SURNUP to collect and propagate only state information for the main hardware and trunking components of the network (for example, using *-T 80* on the default GMDR Criticality Schema). You can, therefore, control the width of SURNUP's reach by connecting it to a specific GMDR and using auto-population restrictions (mainly *-A* subcomps). You can also control the depth of SURNUP's reach through auto-population restrictions (*-A* backbone) or, more powerfully, through the Criticality Threshold (*-T*).

## Network Model Server (NMServer)

The Network Model Server lets various Preside Multiservice Data Manager (MDM) tools (for example, Network Viewer and Component Information Viewer) and customer written tools access the Network Model and retrieve information through the Network Model API Provider. It is controlled by SVM. Refer to the Appendix in 241-6001-201 *Preside MDM Network Model API Reference Guide*. See also references to this server in 241-6001-303 *Preside MDM Administrator Guide* and 241-6001-100 *Preside MDM Installation*. The Network Model Server can be service selected by remote MDM tools, which permit centralized model deployment. The server is multi-threaded to better support multiple users.

**Note:** Threading enables the concurrent execution of multiple requests.

## Network Model Editing Server (EdServer)

The EdServer lets users edit Network Models by using the Network Viewer. This server must be running to use the Network Viewer in edit mode. Security must also be configured and there are multiple levels of security possibilities. Command line options for the EdServer are:

`-L (local access)`

Only users running Network Viewer on the same workstation as the EdServer can edit Network Models.

`-P <password>`

If set, one must provide the specified password to be allowed to edit the model.

The default values for the EdServer are remote access permission without a password.

See also references to this server in 241-6001-303 *Preside MDM Administrator Guide* and 241-6001-100 *Preside MDM Installation*.

## MDM tools and APIs that interact with the Network Model

The Network Viewer, Component Information Viewer, and Component Status Display are Preside Multiservice Data Manager (MDM) tools that interact with the Network Model to display topology and network state information. The Network Viewer interacts with the model through the Network Model Server. Network Viewer must maintain its own version of the model specialized for its purposes. This is why there is a delay when Network Viewer starts while it loads the node (module) and link components into its own display model. Subcomponents are loaded only on an as-needed basis when a module's subcomponent or shelf dialog is opened. Refer to the Network Viewer tool in 241-6001-011 *Preside MDM Fault Management User Guide* for more information.

The Component Information Viewer also interacts with the Network Model Server to display state information or attributes of a given target component. The Component Information Viewer does not load the model in its own memory as Network Viewer does, but probes for it providing a *snap-shot* of

the component's state at a point in time. Refer to the Component Information Viewer tool in 241-6001-011 *Preside MDM Fault Management User Guide* for more information.

Component Status Display is a text-based version of Network Viewer with an important distinction: it does not use the Network Model server and connects directly to the Network Model shared memory. This gives it faster access to the entire model. Component Status Display supports filtering to display only specific types of components with specific states. Component Status Display's use of shared memory also implies that Component Status Display cannot support service selection of the model. However, it is possible to use remote sessions for the same effect. Refer to the Component Status Display tool in 241-6001-011 *Preside MDM Fault Management User Guide* for more information.

The Network Model Application Programming Interface (API) is an ASCII interface used to communicate with the Network Model and to write custom applications. The Network Model API is message based and can be remotely accessed from an MDM workstation. The behavior and rules for all MDM APIs are described in 241-6001-200 *Preside MDM Application Programming Interface Primer*.

## Network Viewer in edit mode

The Network Viewer edit mode allows you to create and make changes to a network model. Some of the edit tasks include populating the model with component information to create configuration data files (CDF); and creating or editing nodes, subcomponents, and links. From the edit mode graphical view of the network, you can position nodes and link bendpoints and assign components to sites and regions. You can also save the changes to the network model and, if needed, distribute it to surveillance applications.

You can perform two types of editing:

- online
- offline

### Online editing

Online editing occurs when you are editing a model while Preside Multiservice Data Manager (MDM) is performing active surveillance using the SURNUP server. During online editing of a model, surveillance (state

propagation) continues but the SURNUP server does not add new components to the model through Auto-Population. DBNLs, however, do get added. The NV buffers any new nodes during an edit session. Upon exit of the edit session, the nodes are integrated into the model.

When a new model is loading, the SURNUP server interrupts surveillance. Once the model is loaded, the SURNUP server reconnects and resynchronizes.

Since online editing prevents new components from being added to the model during an edit session and temporarily suspends surveillance during the loading of a model, use online editing for brief editing tasks.

### **Offline editing**

Offline editing occurs when you are editing a model and MDM is not performing active surveillance. Large editing jobs are best performed by offline editing since new components can be added to the model and there are no disruptions to active surveillance.

Since other NV sessions are affected when you edit a network model, it is best to time your editing when no one else is using it as a monitoring tool. In addition, you can inform other users of your upcoming model edit session.

## **MCDF/SNMPMCDF/Auto-MCDF**

The Make Configuration Data File (MCDF) tool surveys network elements and extracts their configuration information to populate the Network Model. SNMPMCDF performs the same service for multiservice access devices. MCDF extracts all the information needed to create the components, fill in the appropriate informational attributes and create the links that link them. The tool can be used to collect information from one or more modules contained in a node managed by GMDR. MCDF can be instructed to collect only backbone component information.

*Note:* Make Configuration Data files (MCDF) is supported only on Passport and DPN switches.

MCDF and SNMPMCDF can be used directly as a Command Console macro or during a Network Model edit session. The Configuration Data Files created by these utilities are stored in `/opt/MagellanNMS/data/model/cdf` directory. See “Using the MCDF and SNMPMCDF utilities to build a Network Model” (page 103).

There are two tools related to MCDF for DPN-100 and LDM (Legacy Data Module). The first tool is MCDFUTIL which helps you create a list of nodes subject to collection. The other one is auto-MCDF that lets you automatically invoke MCDF when a module is reconfigured. Auto-MCDF inspects the alarm flow from the network and automatically invokes MCDF when a node emits an activated alarm. Auto-MCDF also automatically applies the new data to the active model in shared memory.

Configuration Data Files collected by auto-MCDF are also stored in directory `/opt/MagellanNMS/data/model/cdf` and are destroyed as soon as they are loaded into the active model.

**Note:** MCDFUTIL and Auto-MCDF are currently supported for DPN devices only.

## Network Model macros

The Network Model system supports the following macros.

- *whichcurrent* - enables you to verify the state of the active and startup model
- *whichmodels* - indicates which Models are available to be loaded
- *makecurrent* - loads the model
- *commitmodel* - makes the current model the startup model
- *savemodel* - saves the current active model to file in portable format
- *dumpmodel* - saves the current active model to file in binary format.
- *acknowledge* - puts components in acknowledge state
- *maintenance* - puts components in maintenance state.

See “Network model macros and utilities” (page 119).

## Shared Memory Utilization tools

The Network Model administrator has to closely track the amount of shared memory needed to contain a network model. There are two tools available for this purpose. The Shared Memory Utilization tool is a graphical tool found in the Utility toolset. It monitors the current Network Model shared memory usage and displays it in both megabytes and as a percentage graph. It is a good idea to start the Shared Memory Utilization tool before loading a new Network Model if its fit in the current shared memory segment size is not assured. The tool can monitor the shared memory usage as the model is loaded.

A textual version of this tool is also available as `/opt/MagellanNMS/bin/surmemusg`. This text version displays the current memory segment size, the number of bytes used and the resulting utilization percentage.

## GMDR

The GMDR server manages Alarm Acknowledgment. For additional information on the role of the GMDR server in Alarm Acknowledgment, refer to the 241-6001-310 *Preside MDM Server Reference Guide*

## Fault management tools

All fault management tools related to state-based surveillance currently access and use the Network Model, either directly attaching to the shared memory (Component Status Display) which means it must run on the same workstation as the model or through the Network Model API Server (Network Viewer, Component Information Viewer, Network Model API Provider).

## Configuration management tools

The Component Provisioning tool optionally connects to the Network Model to determine certain conditions, such as the existence of a component. This is exclusively an advisory aspect of the tool; the Component Provisioning tool makes no attempt to modify the Network Model due to provisioning activities.

## Chapter 2

# Network Model Administration

---

The building and use of a Network Model is customer specific. Individual customers must decide what and how much component information to maintain in their model. A complete model would contain all components in the network and all information about each component.

Applications that use the network model can operate effectively without a complete and accurate model; however, the more complete the model, the more information about the network can be displayed to operators.

Organizing a Network Model is an activity specific to your company. The creation of organizational structures is a major task for the Network Model Administrator. Which type of Organization Structures to use, what and how many SITES and REGIONS to create and what nodes and links to assign to them are choices that should reflect how you will be using, managing and surveilling your network.

The network model has to be updated periodically and redeployed. This timing is customer specific.

You can also use the network model for future planning. If you want to see how your network might look if you added new components, you can build a proposed model to a specific view and run applications to ensure its feasibility.

This section contains the following:

- “Network Model administrator’s role” (page 50)

- “Network Model maintenance” (page 54)
- “Network Model deployment” (page 57)

## Network Model administrator’s role

The Network Model administrator ensures the accuracy of component information and topology as the network itself changes. Administrators also deploy updated issues of the model to Preside Multiservice Data Manager workstations.

Basic steps in Network Model administration include:

- Collect network configuration data files (CDFs)
- Load CDFs into a Network Model
- Define Regions and Sites
- Position the network elements
- Define the organization
- Save and distribute the Network Model

## Network Model maintenance tasks

Because the Network Model is a database for components, a certain administrative effort is to be expected. This effort is compounded by the fact that the network being modeled is probably in constant change, whether from adding new components and services or reconfiguring existing ones.

Network Model Administration is subdivided into three major sets of activities:

- *model creation* whereby a new Model is created manually
- *model maintenance* whereby an existing model is modified to reflect changes in the network or to restructure its information
- *model deployment* whereby a created or modified model is distributed in full or in part to other Preside Multiservice Data Manager (MDM) workstations

For an illustration of the maintenance tasks, see the figure “Network Model maintenance tasks” (page 51).



## Network Model creation

A new Network Model can be created and initially populated in a number of ways. For details, see “Using Network Viewer to build a Network Model” (page 59).

**Note:** This method applies only to Passport and DPN switches.

To create a new Network Model run Network Viewer and enter edit mode. Select the *Create New Network Model* from the main menu. Next, use the *Collect Data for Module Group* option from the main menu. This process creates Configuration Data Files (CDFs) containing descriptions of network modules, subcomponents, component names, attributes, links and customer IDs. These CDF files are loaded into the new model by selecting the *Apply/View Network Model collection*. Once completed, the new model now has information on all collected nodes, subcomponents and links. Use Network Model edit capabilities to manually fill out informational attributes for the components or to add components. Components and links may be missing from this initial model if the *Collect Data for Module Group* session was not able to query them (they were unreachable or, for links, they are currently down), if they do not yet exist in the network or if they actually represent components that are not queryable by *Collect Data for Module Group*. If the network elements do not exist, the Network Model can be created by building a new model and manually creating all the necessary nodes, links and subcomponents.

If the network already exists and network command access through macros and the *Command Console* has already been configured, you can collect the necessary configuration data, as Configuration Data Files (CDFs), from all access devices.

From the *Command Console*, invoke MCDF/SNMPMCDF for each top level or all-encompassing group, so that each module is queried. MCDF/SNMPMCDF can also be invoked from a terminal window by using its full path name `/opt/MagellanNMS/bin/snmpmcd`. See “Using the MCDF and SNMPMCDF utilities to build a Network Model” (page 103). The result of this operation is one or multiple CDFs.

**Note:** Make Configuration Data files (MCDF) is supported only on Passport and DPN switches.

You can apply partial Network Models, previously saved as CDFs, to a new model. Using this method, you can build a new model from parts of other models. For example, you can create a central model built from various regional parts. In addition, you can also restore previously saved nodes that are not discovered using an MCDF tool.

### **Organizing and positioning a Network Model**

Once the Network Model components are created, they must be organized and positioned.

- 1 Plan your Network Model organizational structure. This includes taking the following factors into account:
  - As sites are created, they need to be assigned to the proper region nodes.
  - Modules need to be assigned to the proper sites.
  - As physical links are created and as modules are assigned to sites, links are automatically assigned to the proper site or organizational link.
  - The creation and destruction of organizational links is automatically performed to maintain the integrity of the organization structure; in practice, you never have to create or destroy organizational links.
  - As components are being assigned and organized, they should be positioned on the display. If a background map is going to be used in Network Viewer to display the Network Model, it should be assigned to the current organization structure with the *Use Background Map* menu command. The map serves as a guide to properly position the nodes. See “Creating an organization” (page 66).
- 2 Using the active model, create an appropriate organization structure.

**Note:** The model being edited is always the active model.

- 3 Inside the active model, create as many organizational nodes at region and site level as you find necessary for effective grouping and manipulation of your Network Model.

Once you are finished, the organized and positioned model should be legible, mapping to your geographical or organizational view of the real network, permitting the easy localization of nodes and links and minimizing the number of visible nodes and links when a SITE is expanded. This provides for faster fault detection.

- 4 Save and distribute the completed model.

## Network Model maintenance

Throughout its lifetime, the Network Model will have to be modified, corrected, augmented and re-organized to follow the evolution of the real network. The Network Model system is capable of auto-updating itself in part (for example, auto-population, auto-deletion, and auto MCDF), however, some elements of the model, such as organization structure, need manual intervention.

Network Model maintenance is performed using a variety of tools such as Surveillance Network Updater (SURNUP), the MCDF and SNMPMCDF tools, the Network Model editing tools, and other macros. Automatic maintenance is done by SURNUP; manual intervention to delete/add components and change organizations can be performed using the NV edit mode options, *Collect Configuration Data Files* and the MCDF/SNMPMCDF utilities.

### Auto-population and auto-collection

The Surveillance Network Updater (SURNUP) server can auto-populate components not yet present in the model that it is informed of through surveillance. SURNUP can populate the component names and states for modules, links, and subcomponents. For more information see “Surveillance Network Updater (SURNUP)” (page 40).

With auto-collection, if auto-MCDF is used, DPN-100 or LDM modules whose service data is changed and activated are subject to an automatic collection of their new configuration data. This information is automatically integrated into the Model.

If the Network Model has been accumulating such changes over time, it is a good idea to save these changes. Otherwise the new information is lost when a new model is reloaded or the workstation rebooted. To do so, complete the information manually, or invoke a Configuration Data Collection on the incomplete auto-populated components. After collection, you can organize and position any new modules that were created. The resulting model should be saved and re-deployed.

### **Updating existing components**

Periodically, you should perform a Configuration Data Collection on new or auto-populated components and correct the Network Model. If a more global update is needed, invoke MCDF and update the configuration data from the network. The resulting Configuration Data Files can then be loaded into the Network Model. If new nodes are created, they should be positioned and organized.

*Note:* Auto-population, auto-collection, MCDF, SNMPMCDF and *Collect Configuration Data* cannot remove components that are in the model but are no longer in the network; they cannot distinguish between not being able to access a component because it is down or because it does not exist. Obsolete components must be removed manually from the model. For more information, see “Deleting obsolete components” (page 56).

### **Adding non-existent (future) components**

It is possible to proactively add components to the Model that do not yet exist in the network. After the components are commissioned, the Model will be ready for them. Use the Network Model’s editing capabilities to add the new modules, links, and subcomponents and organize and position any new modules. Because these components do not yet exist and are not reported by the network, their states are displayed as Unknown.

### **Adding existing components**

The fastest way to add an existing component to the model is to use MCDF/SNMPMCDF to collect its configuration data. Use the *Create/Edit Component* dialog to create the new node and select Collect Configuration Data from the *Collection* dialog for that node. The module’s subcomponents and links are filled in from the network data. An alternative is to create a

Configuration Data File for the affected module by calling MCDF or SNMPMCDf as a macro in the Command Console or a terminal window and loading the resulting file into the model at a later time.

### **Organizing and positioning components**

As your Network Model grows or simply to get a more legible model, you may want to re-organize or re-position your network elements in the Model. Do this by editing the model, creating or deleting Organizational Nodes as appropriate or re-organizing modules from one site to another and re-positioning node and link endpoints. The modified Network Model should then be saved and deployed.

### **Deleting obsolete components**

The Surveillance Network Updater (SURNUP) server can auto-delete obsolete components once informed by surveillance. SURNUP options determine if a component is deleted from the network model. If the -D option (obsolete component deletion) in SURNUP is ALL, SURNUP can delete the component names and states for modules, links, and subcomponents. For more information see “Surveillance Network Updater (SURNUP)” (page 40).

Some obsolete components may not be deleted automatically:

- 1 Components low in the component hierarchy that have not undergone a state walk or are not polled by the fault servers.
- 2 Modules that are removed from the network that have not been manually removed from the fault servers.
- 3 If the -D option (obsolete component deletion) is restricted for SURNUP.
- 4 If a GMDR server has been disconnected from the SURNUP server.

In these cases, follow the steps below:

- 1 Ensure that the GMDR is reconnected to the SURNUP server. When the GMDR is reconnected, obsolete components are automatically deleted.

- 2 Ensure that the component is removed from the Generic Management Data Server (GMDR), otherwise auto-population recreates it when surveillance resumes. Refer to the GMDR chapter in 241-6001-303 *Preside MDM Administrator Guide*.

If multiple GMDRs are deployed in a hierarchical structure, ensure that the component is removed from all of the GMDRs in the structure starting with the GMDR at the bottom of the structure.

- 3 Remove the component from the Network Model. Removing an existing component from the model cannot succeed if the component still exists in the network. This is because the model recreates the component the next time that the Preside Multiservice Data Manager receives an event from the component.

*Note:* Step 2 is not necessary for most links and constant state components HOST and FNMOD, since they are not present in the GMDR database.

## Network Model deployment

The next sections consider how the Network Model is distributed across Preside Multiservice Data Manager, what kind of Network Model is used and whether the Network Model is the startup surveillance model or not.

Some servers and applications, for example, SURNUP and CSD, need direct access to the Network Model shared memory and, therefore, must have their own copy of the model. Other tools such as Network Viewer, Component Information Viewer, and API can use local or remote models.

## Stand Alone deployment

If the Preside Multiservice Data Manager (MDM) workstation where the Network Model is edited is also a State Surveillance workstation, the only deployment decision to make is whether the edited model should be committed as the default startup model, when MDM reboots. If so, you can use the *NV Save Model Dialog* toggle button.

## Multiple deployment

If there are multiple State Surveillance Preside Multiservice Data Manager workstations and they are to display the same Network Model, edit the model on one of them or on an off-line machine. Then copy the model using FTP or

RCP to each workstation. After the model has been copied to the other workstations it should be loaded and made the startup model, using *makecurrent* and *commitmodel*. For more information refer to “Network model macros and utilities” (page 121).

## Service Selected deployment

The Network Model and its servers can be deployed on a limited number of server set workstations. Other Preside Multiservice Data Manager workstations providing surveillance tools can access these models through service selection. Network Viewer can also edit remote Network Models in the usual manner.

## Deploying Regionalized Models

If your network is fairly big and/or you wish to regionalize its management, you can create models for only the part of the network that is to be managed by a given site. These *regionalized* models must be created, maintained and deployed as though they are complete models. Each regionalized network management system will receive its own partial Network Model and collect data from the Preside Multiservice Data Manager (MDM) surveillance servers (GMDR, DMDR and FMDR) for only the appropriate part of the network.

If overlapping network connections cannot be avoided, SURNUP's subcomponent Auto-Population mode (*-A subcomp*) can be used to ensure that no modules other than those already in the partial Model will be populated. Service selection and remote MDM sessions can be used to maintain centralized access to these regionalized models.

You can use the Save Components capability of the Network Model to construct a central model and then distribute, as CDFs, some of the nodes or organizations to regional models. Alternatively, you can build a central model by grouping partial models, built and saved as CDFs, from regional models.

## Chapter 3

# Using Network Viewer to build a Network Model

---

This section shows you how to:

- create and edit a Network Model
- create and edit components in the network
- link network components
- save and distribute network models to client applications and surveillance operators

This section contains the following topics:

- “Introduction” (page 60)
- “Operations Performed by Network Model Editing” (page 60)
- “Entering edit mode to enable editing” (page 63)
- “Creating a new Network Model” (page 64)
- “Loading a Network Model” (page 65)
- “Creating an organization” (page 66)
- “Attaching a bitmap to an organization” (page 68)
- “Collecting Network Model data” (page 69)
- “Managing collections and applying collected data to the Network model” (page 72)
- “Creating and editing components (nodes) manually” (page 74)

- “Creating and editing links” (page 82)
- “Deleting elements from the Network Model” (page 87)
- “Assigning modules to sites and sites to regions” (page 88)
- “Excluding components from an organization” (page 92)
- “Moving new components into sites” (page 93)
- “Positioning nodes” (page 94)
- “Adding and removing link bendpoints” (page 96)
- “Saving components in collection files” (page 97)
- “Saving and distributing Network Model files” (page 99)
- “Using drag-and-drop features” (page 101)
- “Using the Network Viewer Edit icon bar” (page 101)

## Introduction

Network Models for networks are built using Network Viewer in edit mode. There is a set of edit tools in Network Viewer edit mode provided under the *Network Model Edit* main pulldown menu. Some of these functions are also available using the icon bar.

*Note:* *MCDF* and *SNMPMCDF*, Passport- and DPN-supported commands, can be run independently as a macro from the *Command Console*.

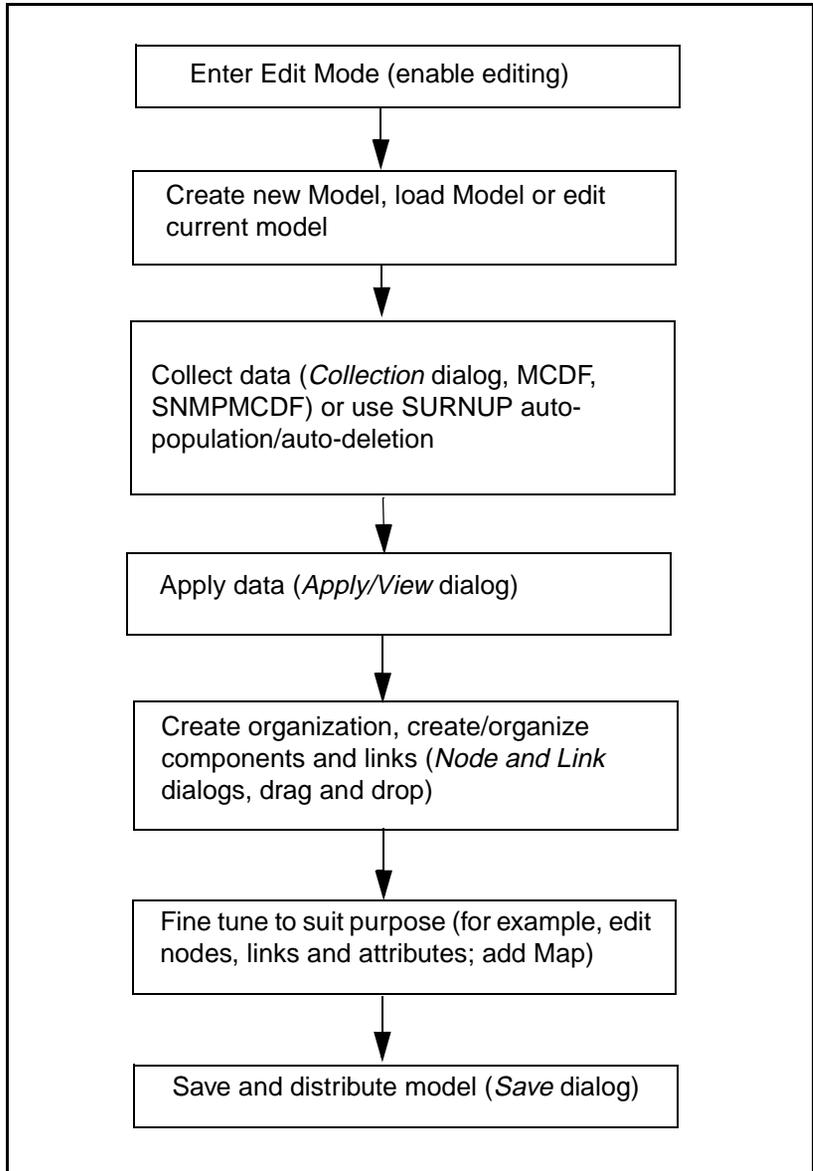
## Operations Performed by Network Model Editing

The following operations can be performed by Network Model editing, which are shown in the figure “Typical steps in using Network Viewer in edit mode” (page 62):

- “Entering edit mode to enable editing” (page 63)
- “Creating a new Network Model” (page 64)
- “Loading a Network Model” (page 65)
- “Creating an organization” (page 66)
- “Attaching a bitmap to an organization” (page 68)

- “Collecting Network Model data” (page 69).
- “Managing collections and applying collected data to the Network model” (page 72)
- “Creating and editing components (nodes) manually” (page 74)
- “Using drag and drop techniques to edit nodes” (page 80)
- “Creating and editing links” (page 82)
- “Using drag and drop techniques to edit nodes” (page 80)
- “Deleting elements from the Network Model” (page 87)
- “Assigning modules to sites and sites to regions” (page 88)
- “Excluding components from an organization” (page 92)
- “Moving new components into sites” (page 93)
- “Positioning nodes” (page 94)
- “Adding and removing link bendpoints” (page 96)
- “Saving and distributing Network Model files” (page 99).
- “Quitting edit mode” (page 100).

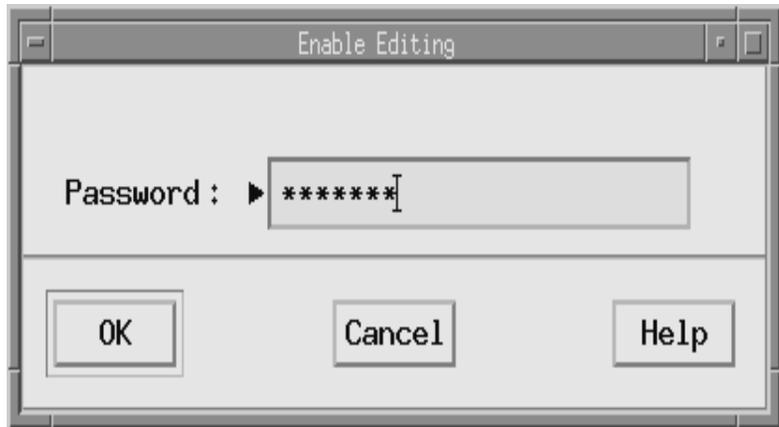
**Figure 6**  
**Typical steps in using Network Viewer in edit mode**



## Entering edit mode to enable editing

Access to Network Viewer's Network Model editing capabilities is controlled by password. You need the proper permissions to access the selections under the Network Model Edit menu and icon bar. See "Network Model Editing Server (EdServer)" (page 44). For information about how to configure the Network Model Edit server, see 241-6001-303 *Preside MDM Administrator Guide* and 241-6001-100 *Preside MDM Installation*.

- 1 On the main window, select Fault Management -> Network Viewer.  
The Network Viewer main window opens.
- 2 From the *Network Model Edit* menu, select *Enable Network Model Editing*.



Depending on your setup, an Enable Editing dialog may open.

- 3 If the Enable Editing dialog opens, enter the appropriate password.  
The password you enter in the dialog must match the password defined in the EdServer. For information about how to set the password in the EDSERVER, see 241-6001-303 *Preside MDM Administrator Guide*.  
If no other user is currently editing the model, the network model editor opens. The Network Model Edit menu becomes available and the Network Viewer icon bar changes to the edit mode icon bar. You can begin editing the current network model.

**Note:** If the model you want to edit is small, there is a momentary wait for the network model edit mode. If the model is large, you may need to wait several seconds.



**CAUTION**

**Risk of unrecoverable error**

Changing node positions is an unrecoverable action.

Ensure that the node position changes have been saved to a file.

- 4 If you change any node positions or endpoints in your edit session, you need to specify whether you want to use and save the node positions in the current Network Viewer display, or use the original positions stored in the shared Network Model.

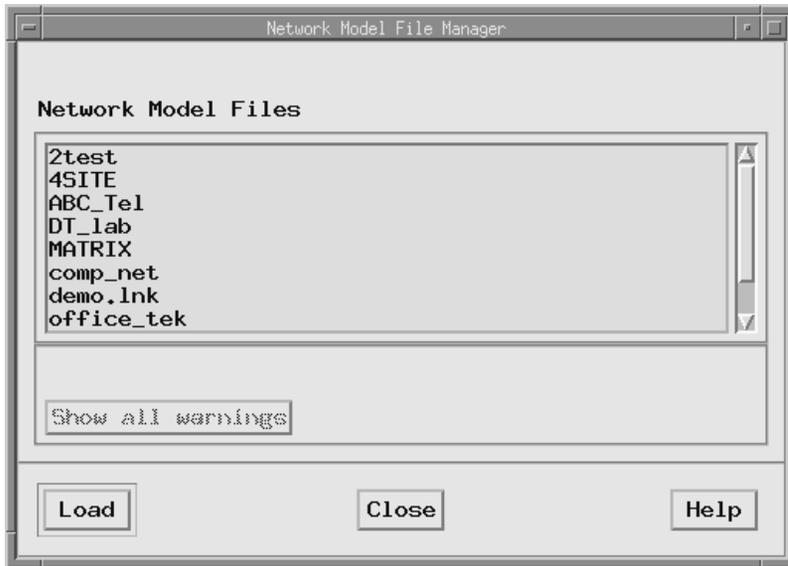
## Creating a new Network Model

- 1 From the *Network Model Edit* menu, select *Create New Network Model*.
- 2 Ensure that your changes have been saved, and select *Yes* to reply to the confirmation.

## Loading a Network Model

Network Models are stored in named files. You can load and save these files to reflect changes in your actual network or to prepare Network Model plans. Use the following procedure for loading an existing Network Model file.

- 1 From the *Network Model Edit* menu, select *Load/View Network Model File*.



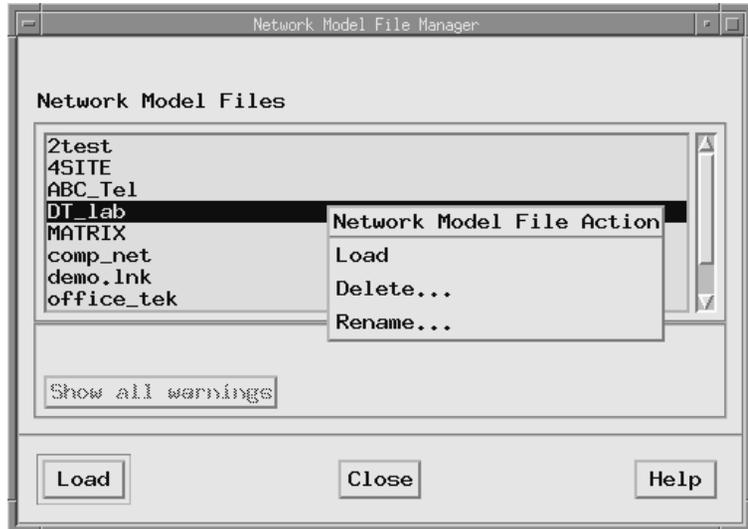
The Network Model File Manager dialog opens.

- 2 From the list of files in the dialog, select a network model file.
- 3 Click the *Load* button.

The Network Model is displayed in Network Viewer and can be edited using Network Viewer edit tools. There may be a delay while the model is loading to shared memory and then to Network Viewer.

**or**

Press and hold the right mouse button to see the *Network Model File Action* pop-up menu.



- 4 Use the right mouse button and select *Load* to load the Network Model file; *Delete...* to delete a Network Model file from the *NMF Files* list; or *Rename...* to change the name of a Network Model file.

If you selected *Load*, the Network Model is displayed in Network Viewer and can be edited using Network Viewer edit tools. There may be a delay while the model is loading to shared memory and then to Network Viewer.

**Note:** NMFs take up significant disk space; we recommend you delete unnecessary NMFs.

## Creating an organization

A complete Network Model contains all the important physical components in your network. You can organize these components into orderly structures to serve various purposes and can create any number of Network Model organizations. For planning purposes, you can see how your Network Model will look when you add new components. You can create a different Network Model organization to divide your network into easily managed segments and to assign responsibility for surveillance to several network operators. Use the following procedure for creating a new network organization.

- 1 From the View menu, select *Select a New Organization to View*.

**Note:** You must be at root level to create a new organization. If you already have an organization open, the *Create Organization* option is disabled.

- 2 From the *Network Model Edit* menu, select *Create Organization*.



The Create Organization dialog opens.

- 3 Press the *Organization Type* menu button and make a selection.
  - 4 Enter an *Organization Name*.
- Note:** Each *Organization Name* must be unique.
- 5 Press *OK*.

The dialog window closes and the icon for a new organization appears in the Network Viewer display. Icons for the organizational nodes `NEW_REGION/ <Orgname>` and `NEW_SITE/ <Orgname>` are automatically created in the Network Viewer display. These icons represent the parents of components in your network. Components valid for the selected organization type are assigned to the Organizational node called `NEW_SITE/ <Orgname>`. Icons for Excluded Components are also created and remain empty until you fill them with components you want to exclude from a network organization.

- 6 To view components in the new organization, select *Open Organization* and use the *Expand* functions on the `NEW_REGION/<Orgname> Node`

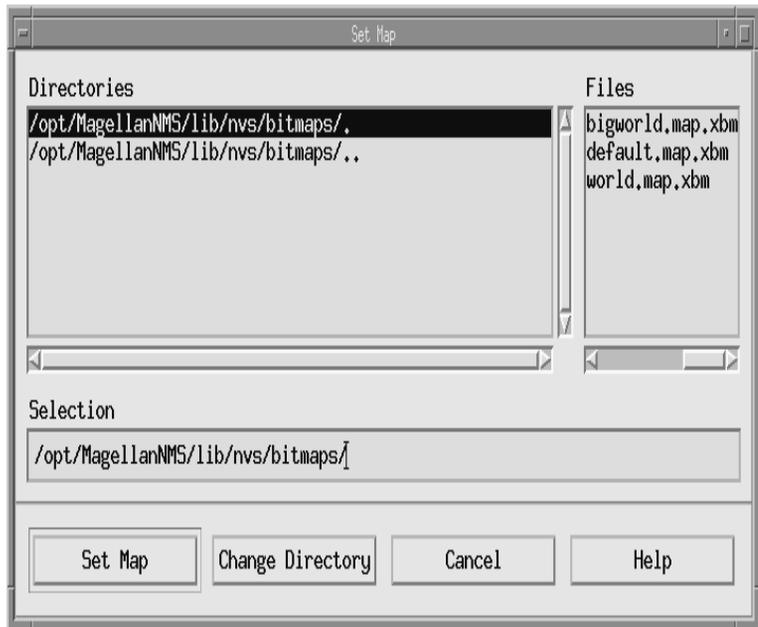
popup menu. Alternatively, you can use the *Show New Modules* option from the Network Model Edit main menu. You can proceed to organize and position components in the new organization.

## Attaching a bitmap to an organization

You can display your Network Model against any background map available to Network Viewer.

**Note:** Map files must be accessible by the machine running Network Viewer. Background maps must be in X11 bitmap format. If you do not have the specified bitmap, the display is presented against a neutral background.

- 1 From the Network Model Edit menu, select Set Organization Map. A file selection dialog displays the available map files.



- 2 Select a map file.
- 3 Press OK.

Your Network Model is displayed against the selected background map.

- 4 To detach a map from an organization, select the *Detach Organization Map* from the *Network Model Edit* main menu.

The background map disappears after you press *OK*.

Additional geographical area pixmaps are available on the Preside Multiservice Data Manager CD-ROM in the directory */cdrom/cdrom0/NVMmaps*. Use the *installmap* tool on the CD-ROM to install these pixmaps.

## Collecting Network Model data

Collected information about your network components is stored in Configuration Data Files (CDF). CDF files contain ASCII descriptions of network modules and subcomponents which includes component names, attributes, links and Customer ID's. You can collect data from all the modules in a network, or you can collect data from selected modules only. After collection, apply these files to your Network Model.

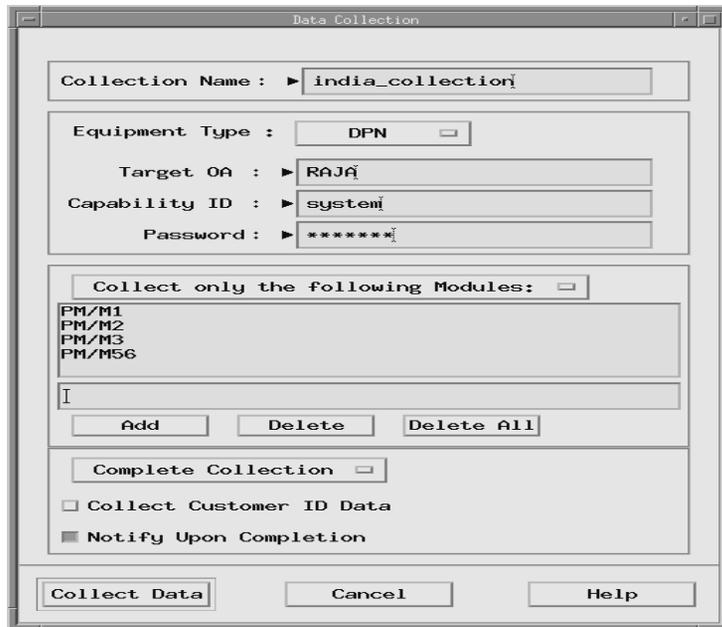
**Note 1:** The procedures in this section for collecting configuration data apply only to Passport and DPN switches.

**Note 2:** Starting the SURNUP server with an option to auto-populate or auto-delete creates or updates the network model data for Nortel Networks Multiservice Provider Edge 9500 (MPE 9500). For information on starting the SURNUP server and about start options, see "Surveillance Network Updater (SURNUP)" (page 40).

### Collecting data for module groups

- 1 From the Network Model Edit menu, select *Collect Data for Module Group*. Alternatively, click the Collect Configuration Data icon in the icon bar.

The Data Collection dialog opens.



- 2 In the Equipment Type field, open the menu and select the appropriate equipment.
- 3 Enter the target OA or Group, your on-switch capability or user ID and your password.

**Note:** The password is your network element authentication, not the password for Network Model Editing permission.

An option button above the module list allows you to define the scope of your collection.

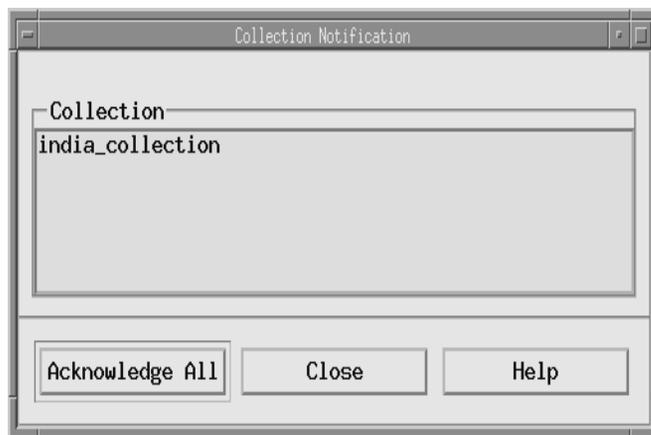
- 4 Click on the option button to display a list of options. To obtain a complete collection of all modules associated with the target, choose *Collect all Modules under <destination>*. Or, to collect only modules listed in the module list area, choose *Collect only the following modules option*.
- 5 To add a module to your collection list, click the Add button.  
To delete a module from your collection, click the Delete button.  
To delete all modules from your collection, click the Delete All button.

- 6 Click to select the Collect Customer ID Data button if you want Customer ID information collected.
- 7 Select the Complete Collection option for a complete collection, or select the Backbone Collection option for backbone-specific data.

If you select a backbone collection, only information for the most important components is collected, for example, module, network link and trunk. A complete collection collects all data.

- 8 Enter a collection name in the Collection Name field or accept the default collection name. The default is provided if no collection name is entered.
- 9 Select the Notify Upon completion button if you want to be notified when the collection is finished.

The following Collection Notification dialog appears when the collection is complete.



- 10 Click the OK button.

Data collection begins. You can close the dialog and continue other edit operations by pressing the Close button. The CDF collection name appears in the Apply/View Network Model Collections dialog with the current status of the collection.

## Managing collections and applying collected data to the Network model

Once you have collected network data in CDF files, you can view the results of your collection and apply the CDFs to your Network Model.

**Note 1:** This procedure applies only to Passport and DPN switches.

During application, no other edit operations can be run. Collection and application processes continue from edit session to edit session as long as the NM server is not restarted. If you attempt to leave edit mode, collection and application processes continue to run.

### Managing Collections

- 1 Choose *Apply/Change Network Model Collections* from the *Network Model Edit* main menu or use its icon bar icon.



The Collection File Manager dialog opens and shows the results of your collection under the Data Collection status column. CDF filenames appear under the Collection Name column. Possible data collection results are as follows:

In Progress	The information is in the process of being collected.
Done	The collection is complete.

Warning	Warnings were generated during the collection process.
Error	Errors were found during the collection process.
Stopped	The collection process was stopped.

- 2 To stop the collection of CDF files, select any number of collections and select *Stop Data Collections* from the Collection popup menu.
- 3 To see a dialog showing collection errors and warnings, select a file from the list and choose *Show Collection Errors and Warnings* from the CDF popup menu. To see errors and warnings from the most recent application, choose the *Show Model Increment Errors and Warnings* option.
- 4 To see a dialog showing the collection parameters for a CDF file, select it from the list and choose *Show Data Collection Parameters* from the CDF popup menu.
- 5 To open a dialog to rename a CDF file, select a file from the list and choose *Rename Collection* from the CDF popup menu.
- 6 To open a confirmation dialog to delete a CDF file, select a file from the list and choose *Delete Collection* from the CDF popup menu.

## Applying Collection results to the Network Model

- 1 To apply your collections to the Network Model, select any number of collections and press the *Apply Collected Data to Network Model* button. You can also use the *Apply Collected Data* selection from the *Collection* popup menu.

The selected CDF files are applied to the Network Model. The Data Application status column gives the results of the application. Possible results are as follows:

Applied	The CDF was successfully applied to the Network Model with no errors or warnings.
In Progress	The CDF is being applied.
Warning	Warnings were generated during a successful application of the CDF.

Error	Errors were found during the application process.
Not Applied	The CDF has not been applied to the Network model.

- 2 To see a dialog showing errors and warnings resulting from the last application of a collection to the network model, select it from the list and choose *Show Model Increment Errors and Warnings*. This option is only available for files that were applied in the current session.
- 3 To automatically delete a collection after its successful application, select the *Delete Collection after Successful Application* button before applying the collection.
- 4 To automatically destroy the old subcomponents (refresh a module collection) before applying a collection, select the *Destroy old subcomponents Upon application* button.

All the current subcomponents for each module contained in the collection file are first destroyed and the collection results applied to the model. The current organizational context (parentage and positions) of the modules are left unchanged.

- 5 To automatically restore the organizational structure upon application, select the *Load Organizational Structure Upon Application* button. This option is used when components were saved using the options explained in “Saving components in collection files” (page 97)

The organizational context (parentage and positions) of the components that you saved in the applied collection is automatically restored upon application.

## Creating and editing components (nodes) manually

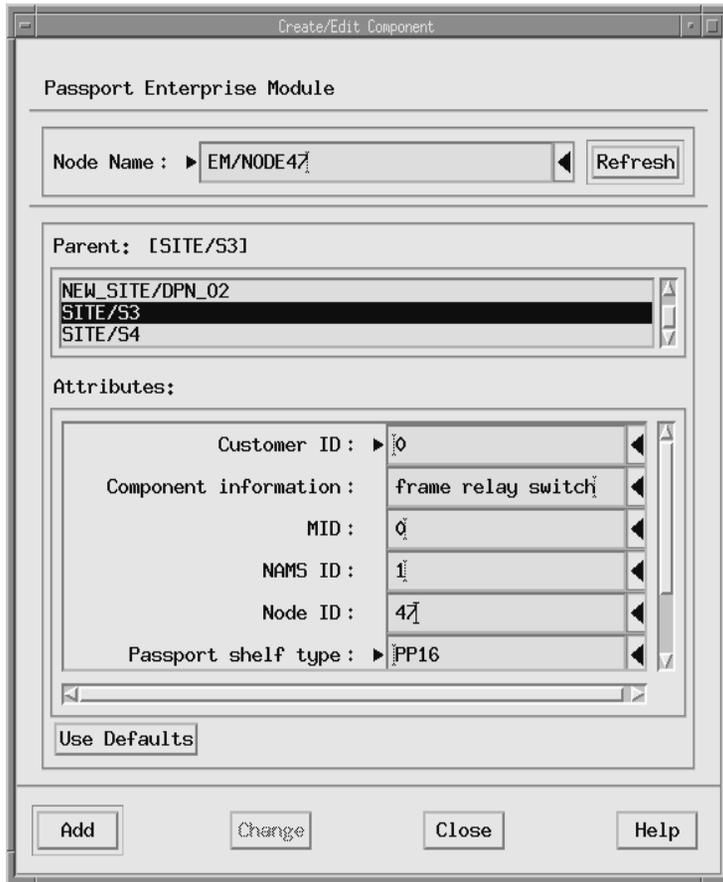
You may manually create and edit Network Model components and subcomponents. For information about collecting component information, see “Collecting Network Model data” (page 69). For more information about editing nodes, see “Using drag and drop techniques to edit nodes” (page 80).

The following procedures describe how to create and edit model components using dialogs and menu commands:

- “Manually creating components and subcomponents” (page 75)
- “Editing components and subcomponents” (page 77)

## Manually creating components and subcomponents

- 1 Choose *Create/Edit Node* under the *Network Model Edit* main menu or use its icon bar icon.



The *Create/Edit Component* dialog appears.

- 2 Enter the component name in the *Node name* field.

**Note:** Another way to edit a node is to simply drag the node from the Network Viewer display and drop it in the *Node Name* field. You can also use the *Node Name* selector at the right of the *Node Name* field. Use the *Get Context* menu item to paste the component name in context to the *Node Name* field.

**Note:** You can create subcomponents by entering the full component name in the *Node Name* field. If, for example, you want to create a port under the parent module *PM/R66* and no *PE* or *PI* exists, type *PM/R66 PE/1 PI/1 PO/1*. The necessary intermediate components (the *PE* and *PI*) are created with default attributes. Alternatively, you can use the menu selector button at the right of the *Node Name* field to build the subcomponent by increments. Click the right mouse button on the selector and choose *Show Legal Types* from the menu. Select the next level subcomponent from the list and it will be appended to the component name. Repeat this procedure until you have created the necessary intermediate components. To see an information dialog about valid names for a component, choose the *Show Legal Names* from the selector.

- 3 Click the *Refresh* button or press the *Enter* key to validate changes.

The editor validates the component ID, fills the *Attributes* area with default values and enables the *Add* or *Change* buttons, as appropriate.

- 4 Assign a parent by choosing the component's parent from the *Parent* list or by creating the component and then dragging its name to a parent in the Network Viewer main display and dropping it.

**Note:** When this dialog is used for subcomponents, the *Parent* area is inactive since the subcomponent name is part of its ID. If you do not select a parent for a node, the special site level buffer *NEW\_SITE/ <orgname>* is shown as its parent. You can assign the node to a different parent later.

- 5 Define the component attributes in the *Attributes* window. The menu button on the right of each *Attribute* fields can be used to display information about that attribute, the valid range of values for numeric values or a list of possible values for enumerated type values. When you select a value from the list, the selected value is inserted in the *Attributes* field.

**Note:** You can use the *Use Defaults* button to select default values for component attributes.

- 6 Click the *Refresh* button or press the *Enter* key.

If the component is a new component, the *Add* button is activated. If you have edited an existing component, the *Change* button is enabled.

- 7 Click the *Change* button to change the attributes of an existing component.

- 8 Click the *Add* button to add a new component to your Network Model.

A new node icon appears in the upper left corner of the Network Viewer display and the *Create/Edit* window remains open.

- 9 Repeat this procedure to create/edit another component.

**Note:** You cannot edit the criticality value of an organization node.

You can also use the *Node* popup menu to edit nodes.

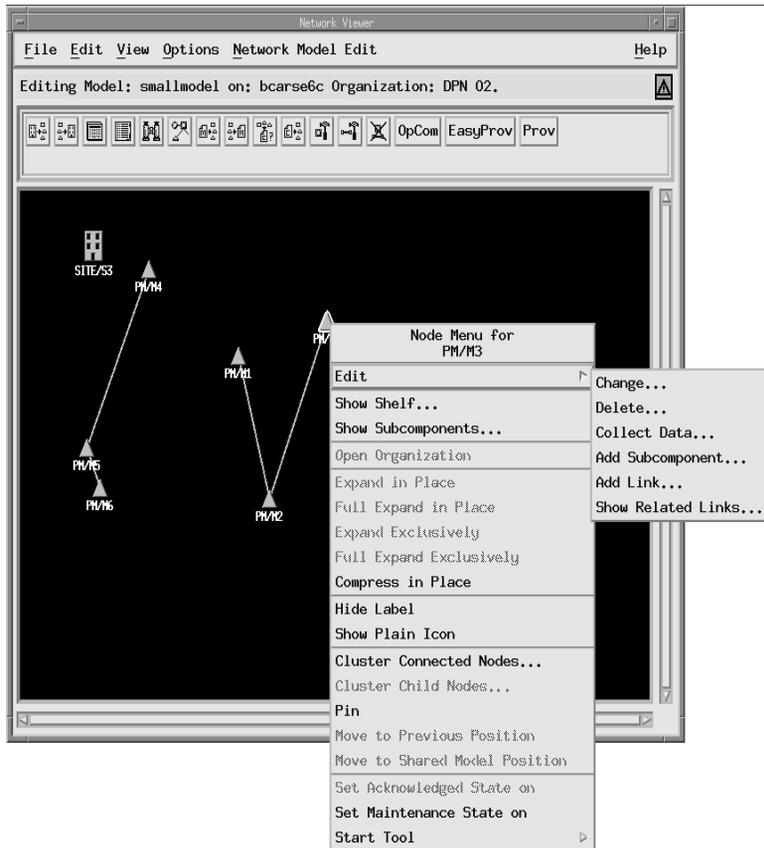
## Editing components and subcomponents

There are several ways to edit an existing component: repeat the above procedure and click the *Change* button; select the component from the Network Viewer main display, bring up the *Node* popup menu using the right mouse button and choose *Change* from the *Edit* cascade menu; or simply drag a component to the dialog and repeat the above procedure from step 4.

**Note:** You may want to use an existing component's set of attributes when you create components of the same type. One way to do this is to clone component attributes. To do this, drag an existing component into the dialog, change its ID, click *Refresh* and use the *Add* button.

### Using the Node popup menu to edit component information

- 1 Select a node from the Network Viewer display and click the right mouse button to bring up the *Node* popup menu.

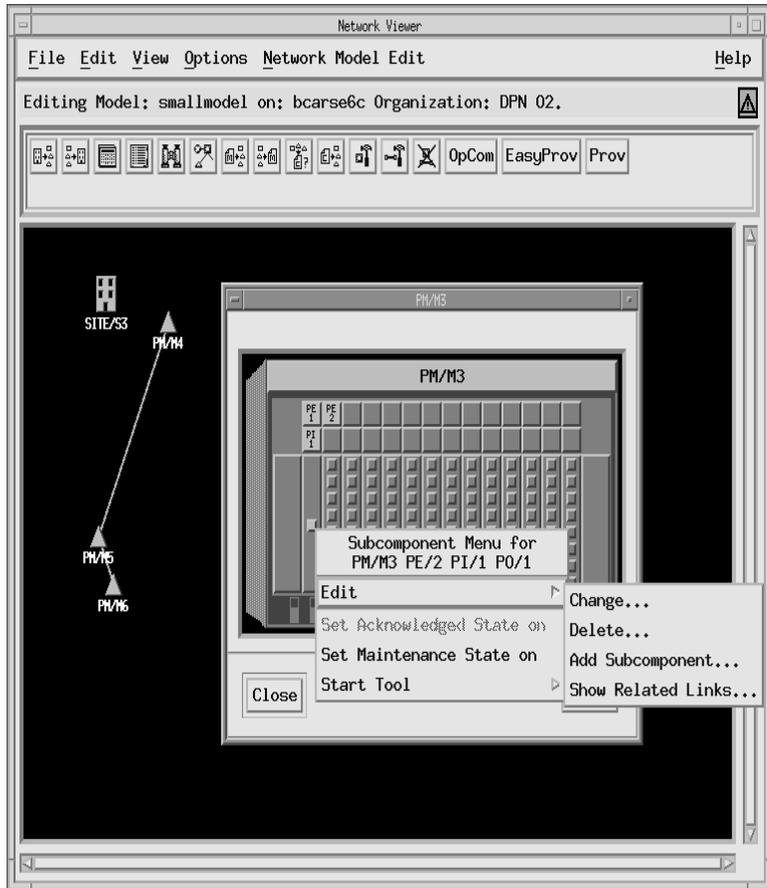


- 2 Choose an option from the cascade menu available beside the *Edit* option.
  - Change lets you alter the node's characteristics and attributes.
  - Delete brings up a confirmation dialog.
  - Collect Data opens the Collect Data dialog with the selected node id pasted in the module list.

- Add Subcomponent and Add Link options let you add children to the node or create links with other nodes.

### Using the Subcomponent popup menu

- 1 Select a subcomponent from the Network Viewer *Shelf* or *Subcomponent* dialogs and click the right mouse button to bring up the *Subcomponent* popup menu



- 2 Choose an option from the option menu beside the *Edit* option.
  - Change lets you alter the node's characteristics and attributes.
  - Delete brings up a confirmation dialog.

- Add Subcomponent lets you add another subcomponent.
- *Show Related Links* brings up a dialog showing links associated with the selected node/subcomponent.

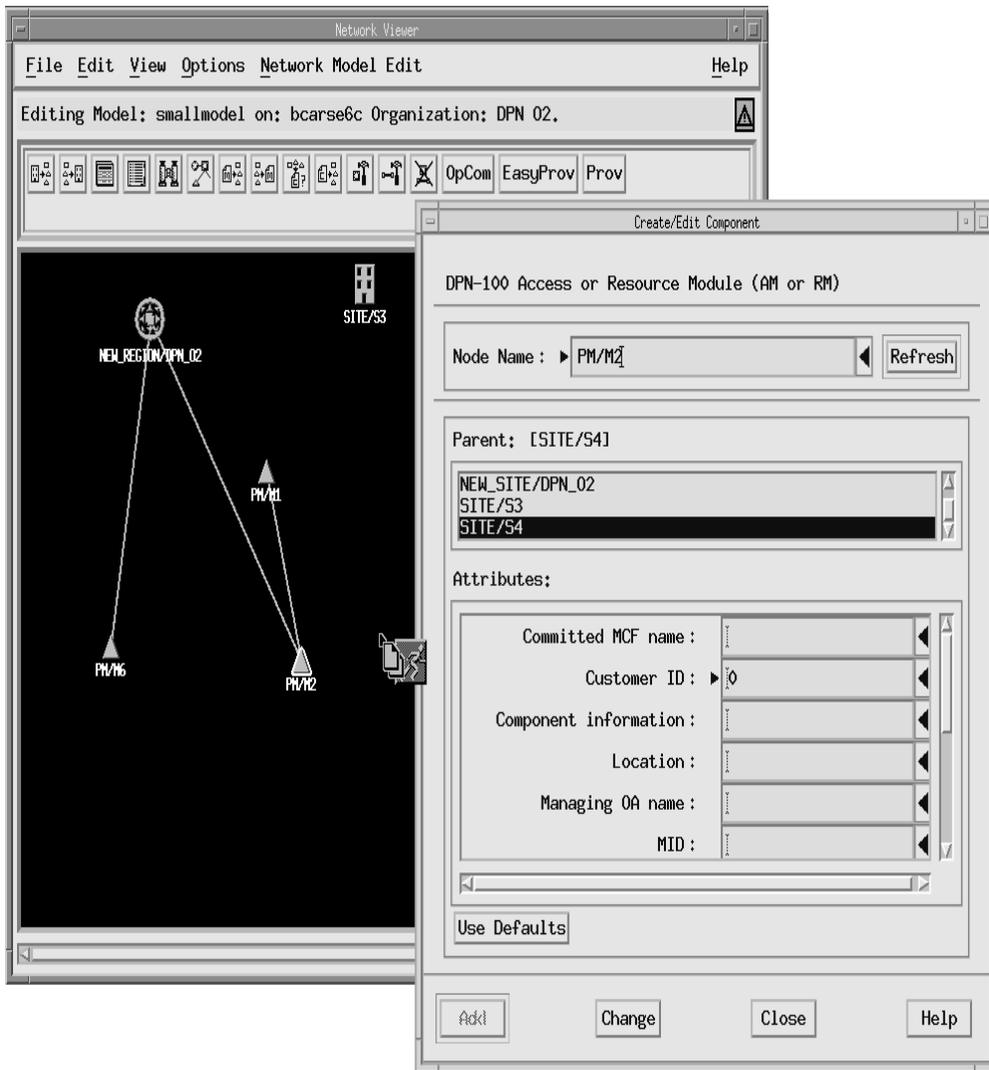
## Using drag and drop techniques to edit nodes

To edit a node, you simply have to drop elements representing a component into the *Create/Edit Node* dialog. The Edit dialog is automatically refreshed to show you the attributes of the dropped elements.

Drag Sources for nodes is any element which represents a component ID or which contains a component ID, for example, a Node Icon in Network Viewer, an alarm in Component Information Viewer or Alarm Display, a component in Network Viewer Subcomponent dialog. The drop sites to Edit Nodes is the Create/Edit Node dialog.

See the figure “Using drag and drop to edit nodes” (page 81) for an example of using drag and drop to edit node attributes. The figure shows a node icon dragged from the Network Viewer display to the *Create/Edit Node* dialog. The dialog was updated with the selected node’s attributes.

**Figure 7**  
**Using drag and drop to edit nodes**



## Creating and editing links

Network Model subcomponents and modules are joined by physical links. You can create various kinds of links between these components, edit these links and control Network Viewer's display of linked components. The *Create/Edit Link* dialog is used to create links and edit link information.

**Note:** Link endpoints are created automatically when links are created between endpoints with existing module parents. The necessary intermediate components are created with default attributes.

There are several ways to create/edit links between two components. You can select both module icons from the Network Viewer display and bring up the *Create/Edit Link* dialog. You can also drag a link from the Network Viewer display to the *Create/Edit Link* dialog. Finally, you can drag a link from the Related Links dialog to the *Create/Edit Link* dialog.

**Note:** You cannot create or edit organizational links; they are automatically managed by the Network Model.

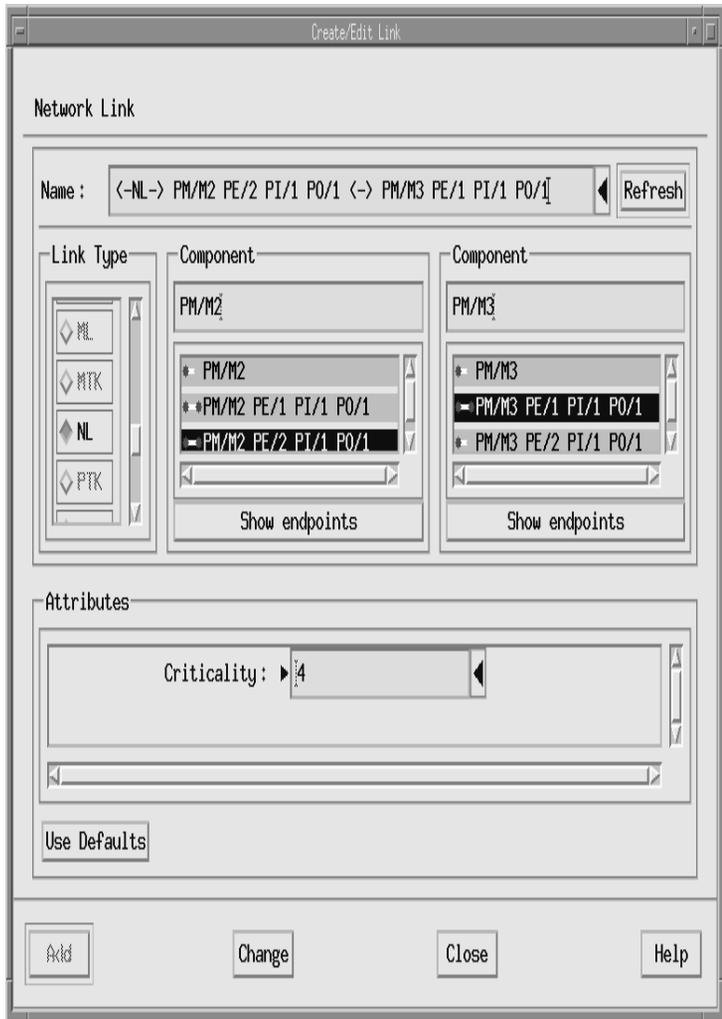
## Using menu commands to create and edit links

The following procedures describe how to create and edit links using dialogs and menu commands.

### Creating links

- 1 Choose *Create/Edit Link* under the *Network Model Edit* main menu or use its icon bar icon.

The *Create/Edit Link* dialog is displayed.



- 2 Enter the component endpoint names in the *Component* fields, drag and drop a component from the main display to the component field or select an existing link from the Network Viewer main display.

**Note:** The menu button on the right of the *Link Name* field has two menu options. *Get Context* pastes the link name in context to the *Link Name* field. *Display Information* brings up an information dialog about the current link type.

If a link was already selected in the Network Viewer display, the link dialog is updated to show the link endpoints and attributes. If no link was selected but a module was, the component field is updated to show possible endpoints related to that module. Connected endpoints are displayed as a barbell icon; possible legal endpoints are displayed as a half barbell.

- The component is an endpoint in an existing link.



- The component is not an endpoint in a link but could be.



- Notification symbol: If you create a link between this subcomponent and another one, this subcomponent will also be created.



**Note:** Link Types are restricted to valid link types for the endpoint selections. Each *Link Type* button is enabled/disabled according to the endpoint selection.

- 3 For links between subcomponents, press the *Show endpoints* button to see existing endpoints in the component text field.
- 4 Choose an endpoint from each component text field.  
The Link Name field is filled.
- 5 Define the link attributes in the *Attributes* window. The menu button on the right of each Attribute fields can be used to display information about that attribute, the valid range of values for numeric values that can be selected for that attribute and a list of possible values for enumerated type values. When you select a value from the list, the selected value is inserted in the *Attributes* field.

- 6 Click the *Refresh* button.

*For subcomponents the Attributes field is filled with values from a previous Create Link session appearing in the display. If the link does not exist in the Network Model, the Add button is enabled. If the link already exists, the Change button is enabled.*

- 7 Click the *Add* or *Change* button.

*If a new link has an endpoint that terminates another link, you are warned that creating the link causes the destruction of another link. A dialog asks you to confirm your choice. The Change button is enabled and the Network Viewer display is updated with the new link. The link dialog remains open.*

If both module endpoints exist, any subcomponent needed to create the link is automatically created.

### **Example**

The following procedure creates a link between two endpoints that do not exist in the Network Model. In this example the link is between two ports; the components being linked are: PM/M1 PE/1 PI/1 and PM/M2 PE/1 PI/1.

- 1 Enter the component names in one or both *Components* fields.
- 2 Click the *Show Endpoints* button for each component.  
A half barbell icon shows the potential legal endpoints for the link.
- 3 Select a *Link Type* from the list.
- 4 Click the *Add* button.

The ports and all intermediate components (if necessary) are created and so is the link. In this example ports PO/1 are created for each of the two components.

### **Editing link Information**

- 1 Add or edit another link by changing at least one of the endpoints; selecting a new link type from the *Link Type* menu; or dragging another link from the Network Viewer display to the *Create/Edit Link* dialog.

**Note:** Another way to edit a link is to simply type the link name in the *Link Name* field. You can also use the *Link Name* selector *Get Context* menu item and click *Refresh*.

- 2 The *Node* and *Subcomponent* menus include a *Show Related Links* option. Selecting this option brings up a dialog showing links associated with the selected node/subcomponent.

**Note:** You can also use the *Link* popup menu to edit existing links.

### Using the Link popup menu

- 1 Select a link in the Network Viewer display and click the right mouse button to bring up the *Link* popup menu.
- 2 Choose *Delete* or *Change* from the *Edit* cascade menu to delete or alter a link to the selected node.

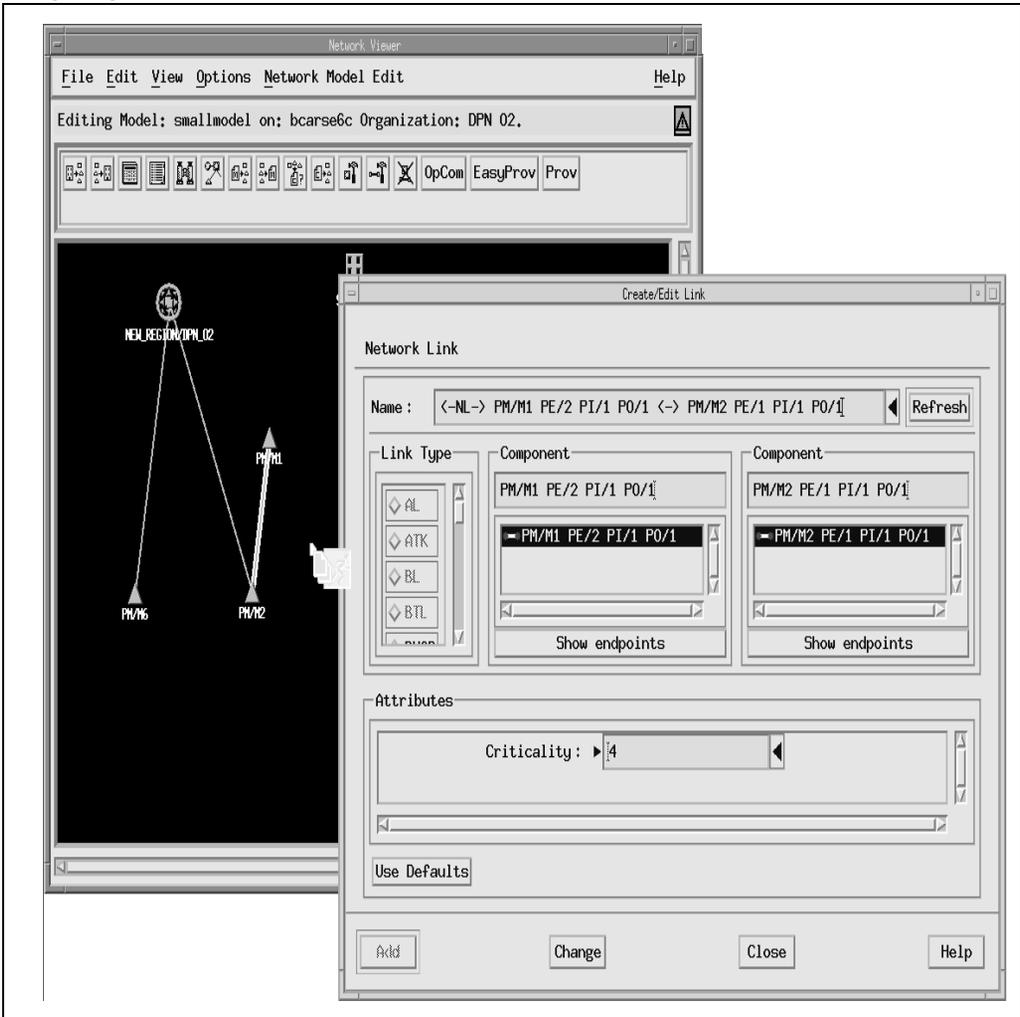
### Using drag and drop to edit links

To edit a link, you simply have to drop elements representing a component into the *Create/Edit Link* dialog. The Edit dialog is automatically refreshed to show you the attributes of the dropped elements.

Drag Sources for links is any element which represents a component ID or which contains a component ID, for example, a Node Icon in Network Viewer, an alarm in Component Information Viewer or Alarm Display, a component in Network Viewer Subcomponent dialog. The drop sites to Edit Links is the Create/edit link dialog.

See the figure “Using drag and drop to update links” (page 87) for an example that illustrates a drag and drop of a link icon selected from the Network Viewer display to the Create/Edit Link dialog. The first screen shows the selected link. After the middle mouse button is pressed, a drag icon is displayed and follows the cursor. The second screen shows what happens after the middle button is released over the drop site. The link dialog screen has been updated with the link name and its attributes. You could proceed to edit this link by changing the link type, endpoints or attributes using the link dialog.

**Figure 8**  
Using drag and drop to update links



## Deleting elements from the Network Model

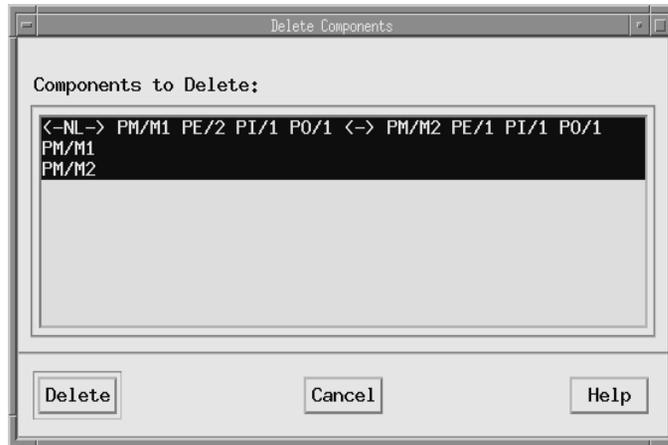
Several Network Model edit menus include delete options. Selecting *Delete* from any menu always brings up a dialog to confirm your deletion. Deleting an organization removes regions, sites and virtual links but does not remove modules, subcomponents and physical links from the Network Model. Note that you cannot delete the *DEFAULT/ALL* organization, its organizational

nodes or the *NEW\_SITE*, *NEW\_REGION*, *EXCLUDED\_SITE* and *EXCLUDED\_REGION* nodes. If you delete a site, the underlying modules are excluded from the organization and appear in the Excluded Modules Dialog, *EXCL\_SITE/<orgname>*.

Deleting modules and subcomponents removes them from the Network Model, that is, from all of the organizations in that Network Model. Deleted links are also removed from the model. Each time you use a delete option, the confirmation dialog appears.

The figure “Delete Confirmation dialog” (page 88) shows a sample dialog.

**Figure 9**  
**Delete Confirmation dialog**



## Assigning modules to sites and sites to regions

To create a Network Model, you need to assign modules to sites, and sites to regions. You can use drag and drop to simplify assigning components.

To assign module(s) to a site, pick modules from one of the drag sources representing modules (Module Node Icons, NV Finder Dialog Components List, Excluded Module Dialog Modules List, and New Module Dialog modules list) and drop them in one of the drop sites representing a site (Site Node Icon, Excluded Module Dialog and New Module Dialog).

To assign site(s) to a region, pick sites from one of the drag sources representing sites (Site Node Icons and NV Finder Dialog Components List) and drop them into a drop site representing a Region (Region Node Icon).

The Drag Sources are the following:

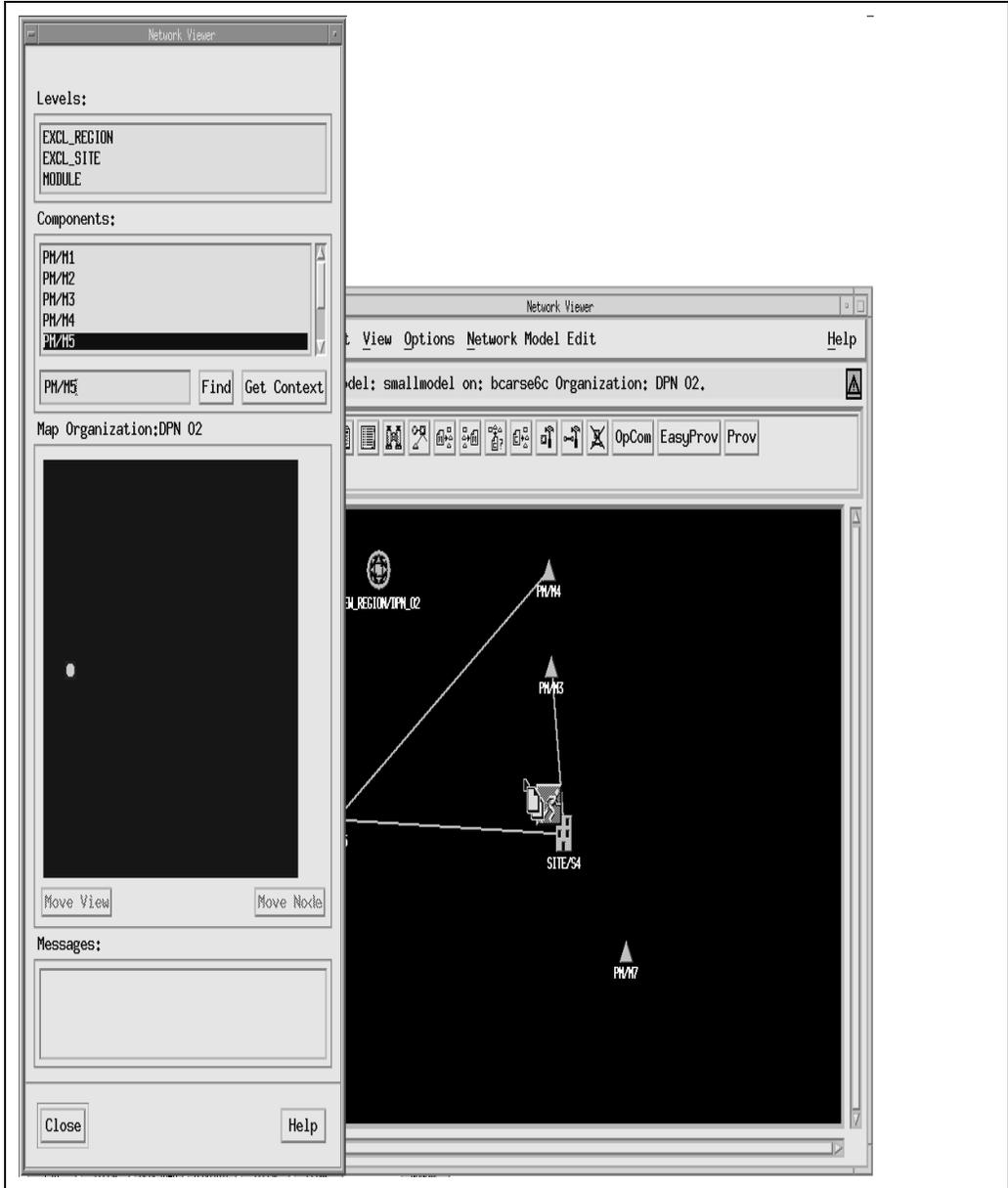
- NV Finder Dialog Components: Modules or Sites from the Components list may be dropped on valid drop sites or regions.
- Excluded module dialog modules list: Modules from this list may be dropped on valid drop sites thus including the excluded modules in the current organization.
- New modules Dialog Modules list: Modules from that list may be dropped on valid drop sites thus assigning the new modules to proper sites in the current organization.
- Node icon: A node icon representing a module or a site may be dropped on valid drop sites or regions.

The Drop Sites include the following:

- REGION component dialog: A drop site for sites. The whole dialog acts as a drop site representing the Region node icon. Dropping modules on the dialog assigns the dragged sites to the region represented by this dialog.
- SITE component dialog: A drop site for modules. The whole dialog acts as a drop site representing the Site node icon. Dropping modules on that dialog assigns the dragged modules to the site represented by this dialog.
- Excluded modules Dialog: A drop site for modules. The whole dialog acts as a drop site representing the excluded site node icon. Dropping modules on that dialog exclude them from, the current organization.
- New module Dialog: A drop site for modules. The whole dialog acts as a drop site representing the new site node icon. Dropping modules on that dialog will make them new modules for the current organization.
- Node icon: Site node Icons are drop sites for modules and Region Node Icons are drop sites for sites.

See the figure “Using drag and drop to assign nodes to sites” (page 91) for an example of dragging from the NV Finder module list to a site node in the Network Viewer display. This assigns the module to the selected site. To reassign this module to another site, drag the icon to that site.

**Figure 10**  
Using drag and drop to assign nodes to sites



## Excluding components from an organization

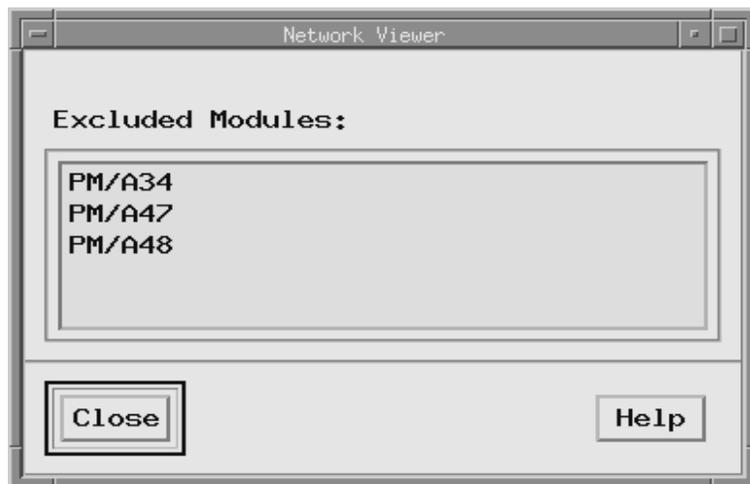
This dialog lists all modules which you do not want to be part of the currently displayed organization. The dialog is invoked from the Network Model Edit main menu using the *Show Excluded Modules* menu option.

The *Excluded Modules* dialog is a drop site for modules. The whole dialog acts as a drop site representing the excluded site node icon. Dropping modules on that dialog exclude them from the current organization.

Another way to see unassigned modules is to expand the organizational nodes EXCL\_REGION and/or EXCL\_SITE appearing in the Network Viewer display. These nodes are only visible if the *Show Excluded* option is chosen from the *View* menu. To assign the new modules to another site, drag them and drop them onto a new site. This is equivalent to dragging an excluded module icon in the Network Viewer display and dropping it on a site icon. You can also drag and drop excluded modules from this list to other Network Model Edit dialogs.

For an illustration of the *Excluded Modules* dialog, see “Excluded Modules dialog” (page 92).

**Figure 11**  
**Excluded Modules dialog**



## Moving new components into sites

The *New Modules* dialog lists all modules which have not been assigned to the currently displayed organization.

The *New Modules* dialog is a drop site for modules. The whole dialog acts as a drop site representing the new site node icon. Dropping modules on that dialog will make them new modules for the current organization.

New modules will appear in your organization for the following reasons:

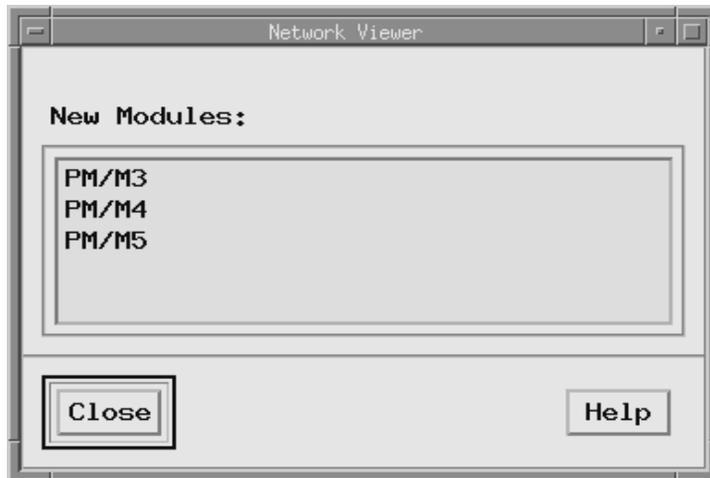
- When you create an organization all the existing modules in the network are filtered for their type and if appropriate appear as new modules.
- The SURNUP server autopopulated the model with new modules. See “Surveillance Network Updater (SURNUP)” (page 40).
- Collection Data Files (CDF) containing new modules were applied to the model. The new modules filtered for their types appear in each organization as new modules. CDF may be applied manually or automatically. See “Collecting Network Model data” (page 69).

The dialog is invoked from the Network Model Edit main menu using the *Show New Modules* menu option. Another way to see new modules is to expand the *NEW\_SITE/<orgname>* appearing in the Network Viewer display.

To assign the new modules to another site, drag them and drop them on that site. This is equivalent to dragging the corresponding module icon in the Network Viewer display and dropping it on a site icon. You can also drag and drop new modules from this list to other Network Model Edit dialogs.

The figure “Using drag-and-drop features” (page 101) shows a sample dialog.

**Figure 12**  
**New Modules dialog**



## Positioning nodes

You can use the mouse to position nodes in the Network Viewer display, if the Network Viewer is currently in Model Editing mode. These position changes are made permanent into the Network Model and therefore available to all NMFs monitoring this network. Otherwise, these changes are temporary and are known only to the current Network Viewer session.

There are cluster and pin options available to help you make legible and efficient positioning. Nodes are recorded in the Network Model where you position them in the display.

## Positioning nodes

- 1 To move a single node, press and hold the left mouse button and move the node to the position in the display you want it to occupy.

The cursor changes to a crosshair and the node follows the cursor.

- 2 Release the left mouse button to position the node in the display.

The new node location appears in the Network Viewer display and is recorded in the Network Model.

## Positioning multiple nodes

- 1 To select several nodes draw a rubber band around them or press and hold the *Shift* key and use the left mouse button to select nodes and links.
- 2 Position the cursor on one of the selected nodes, press and hold the *Shift* key and click the left mouse button.

When you start the move, a rectangle outlines the group.

- 3 Move the group to a chosen position and release the left mouse button.

Nodes are repositioned in the Network Viewer display and are recorded in the Network Model. Selected links with endpoints are also moved.

## Using the cluster and pin options

- 1 If there are nodes you do not want moved, invoke the *Node* popup menu and select the *Pin* option from the *Edit* cascade menu.

Selected nodes are pinned to their positions in the display and are not affected by cluster commands.

- 2 Select a node with the left mouse button and invoke the *Node* popup menu using the right mouse button.
- 3 Select *Cluster Connected Nodes* or *Cluster Child Nodes* from the *Edit* cascade menu.

*Cluster Connected Nodes* positions all nodes which are connected to the selected node in a circle around the selected node; *Cluster Child Nodes* expands the selected node and positions all subordinate nodes in a grid around the expanded node's original position.

*Note:* Nodes are never allowed to be positioned outside the map or viewspace boundaries and will be automatically forced on if needed.

## Positioning a node to its shared position

- 1 Select the node with the left mouse button and invoke the *Node* popup menu using the right mouse button.
- 2 To position the node to its previous position, select *Move to Previous Position* from the *Node* popup menu.

The node is repositioned to its previous position.

- 3 To position the node to its shared position, select *Move to Shared Model Position* from the *Node* popup menu.

The node is repositioned to its original position.

## Reverting nodes to their current Network Model location

- 1 To revert all nodes to their current Network Model location, select *Set Shared Model Positions for all Nodes and Links* from the *View* menu.

OR

To revert selected nodes to their current Network Model location:

- 2 Select *Set Shared Model Positions for Selected Nodes and Links* from the *View* menu.

The nodes will revert to their current Network Model location.

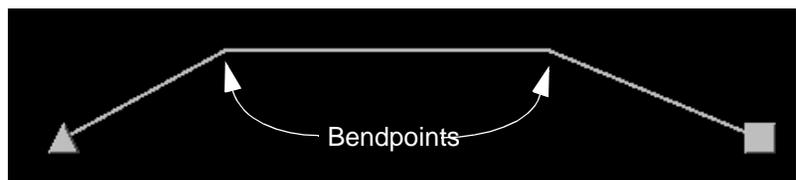
## Adding and removing link bendpoints

You may add bendpoints to a link to separate it from other links which may be superimposed over one another. Bendpoints are the points where the lines meet. Link menus contain commands to show and hide bendpoint *handles*. Bendpoint handles must be showing in order to move or delete single bendpoints.

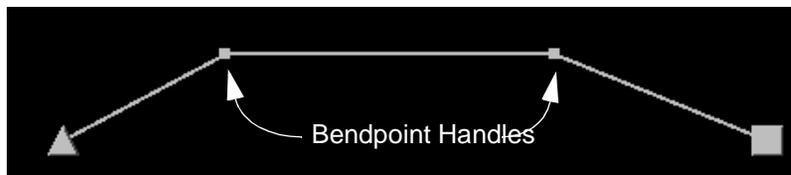
Overlapping links between two nodes can also be spread out by the automatic insertion of bendpoints.

The figure “Link containing two bendpoints” (page 96) shows a link containing two bendpoints. The figure “Link containing two bendpoints with handles showing” (page 97) shows the same link with bendpoint handles showing. You can access a bendpoint menu to perform bendpoint related activities.

**Figure 13**  
**Link containing two bendpoints**



**Figure 14**  
**Link containing two endpoints with handles showing**



## Accessing the endpoint menu

- 1 Move the cursor over the link whose endpoints you want to work with, press *menu*, and choose *Show Endpoint Handles*.  
**Note:** If the endpoint handles are already showing, skip this step and proceed to step 2.
- 2 Move the cursor over the endpoint handle you want to work with and press *menu* to display the endpoint menu selections.
- 3 Choose an action from the cascade menu.

## Automatically spreading overlapping links

**Note:** This is only available if there is more than one link between two given nodes.

- 1 Spread the links using the *Link* popup menu. Use the Spread Links command.

OR

- 2 Double-click on the link.

All previous endpoints are removed. New endpoints are inserted at mid section to spread the links evenly.'

## Saving components in collection files

After you finish making changes to your model, you can save a selected set of current network model modules (and their subcomponents) to disk in portable collection files. Select the modules to save by specifying module naming patterns (for example, \* and ?). Organizational nodes and organizational root names are also specified in this manner. When nodes are saved, their corresponding children and parent nodes (up to the organizational root) are also saved. You can save the current organizational context (parentage and positions) of the selected modules, or you can save all the links

related to the selected modules. The standard load collection capability of Network Viewer may be used to increment a current model with the saved components, links and organizational structures.

- 1 Choose *Save Components* from the *Network Model Edit* main menu, or use its icon bar icon.



- 2 Enter a collection name in the *Collection Name* field or accept the default collection name. The default is provided if no collection name is entered.
- 3 Click the *Add* button to add a module, organizational node or root, or a regular expression to your collection list; Click the *Delete* button to delete the module from your collection; or click the *Delete All* button to delete all modules from your collection. You can specify a complete module name like PM R66 or a regular expression like FNMOD \* or EM A\*. By example suppose the components list contains the following expressions: PM R66, FNMOD \* and EM A\* then click the Save Components button to save the PM R66 module, all the FNMOD modules and all the EM modules with a name starting with the letter A.
- 4 You may optionally choose to save all links related to the selected modules by clicking the *Save Related Links* button.
- 5 You may optionally choose to save all the organizational structures related to the selected modules by clicking the *Save Organizations*

button. Saving the organizational structures will also save the positions of the modules and of their related links.

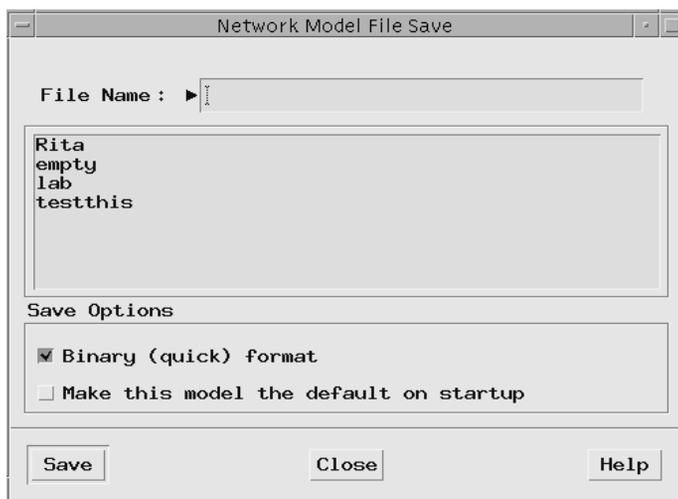
- 6 Click the *Save Components* button. The standard load collection capability of Network Viewer may be used to increment a current model with the saved components, links and organizational structures. See “Managing collections and applying collected data to the Network model” (page 72).

## Saving and distributing Network Model files

After you finish making changes to your model, save your Network Model to a central location. You can choose which format your model is saved in and you can commit a specified model to be the common startup model for surveillance. The default save option is ASCII (portable) format.

- 1 Choose *Save Network Model* from the *Network Model Edit* main menu, or use its icon bar icon.

The Network Model File Save dialog is displayed.



- 2 Enter a Network Model name in the *File Name* field or select an existing Network Model from the list.
- 3 Select the format to save the network model file:

- a. If desired, select the Binary (quick) format option. This saves the model in a non-portable, but faster-loading format.
- b. Leave the Binary (quick) option unselected if you wish to save the model in a format that is portable.

For additional information, refer to “Saved model” (page 36).

- 4 Click the *Make this model the default on start-up* button if you want this model to be the common startup model. For more information, see “Startup (committed) model” (page 37).
- 5 Click the *Save* button.

## Quitting edit mode

- 1 Choose *Leave Network Model Editing* from the *Network Model Edit* main menu.

A dialog appears asking if you want to save any changes since the last save.

- 2 Make a selection. Choose *Save* to save changes made during the current edit session; *Leave without saving* to simply quit edit mode; and *Cancel* to dismiss the dialog and remain in edit mode.

**Note:** If the “Resynch the Network Model states with GMDR” checkbox is checked, all states in the network will be recalculated based on the information in GMDR.

## Quitting Network Viewer while in edit mode

- 1 If you attempt to quit Network Viewer while in edit mode, a dialog is presented, stating that you have made changes to the Network Model, and that you can exit with or without saving them.
- 2 Make a selection. Choose *Exit and Save* to save changes made during the current edit session; *Exit without saving* to quit Network Viewer; and *Cancel Exit* to dismiss the dialog and remain in edit mode.

Once you confirm your choice to quit, Network Model Edit selections are disabled and you exit Network Viewer.

**Note:** If the “Resynch the Network Model states with GMDR” checkbox is checked, all states in the network will be recalculated based on the information in GMDR.

## Using drag-and-drop features

Network Viewer in edit mode provides drag-and-drop functionality which can be used instead of manually typing component IDs. You can select modules, nodes or links from the Network Viewer display and drag-and-drop them into a dialog. This pastes the component's name into the dialog's *Name* field and, if appropriate, inserts attribute values in the appropriate fields. Alternatively, you can select a node or module icon from the edit dialog you are using and drag-and-drop it into an organizational node (for example, region or site) in the Network Viewer display. This action assigns dragged components to the chosen organizational node in the Network Model.

To drag a single element, position the cursor over the icon in the display or a component name in a dialog list. Press the middle mouse button to begin the drag and hold it while you drag the cursor to the drop site. To drop the element, release the middle mouse button.

For multiple component drag-and-drop operations, select a group of components in the Network Viewer display. Begin the drag by moving the cursor over one of the selected elements in the group, press and hold the middle mouse button and move the drag icon to a location in the display or to another dialog. This action moves the group to the drop zone. Release the middle button to drop the group. This is useful for assigning multiple nodes to a site and for moving multiple nodes into the collection list.

**Note:** The cursor icon changes when you drag selections over legal drop sites. Drop attempts fail if the moves you make are not legal for the selected items or locations.

## Using the Network Viewer Edit icon bar

Network Viewer in edit mode includes an icon bar immediately above the main display. The icons offer a quick way to invoke commonly used commands found in Network Viewer's pulldown menus. Position the cursor over an icon and press the left mouse button to activate its command. For additional information on the Network Viewer icon bar, see 241-6001-011 *Preside MDM Fault Management User Guide*.



---

## Chapter 4

# Using the MCDF and SNMPMCDF utilities to build a Network Model

---

This section describes the programs and tools used to create Configuration Data Files (CDF) from an existing network, procedures for maintaining the Network Model, and information on “Collecting component data from the network” (page 104).

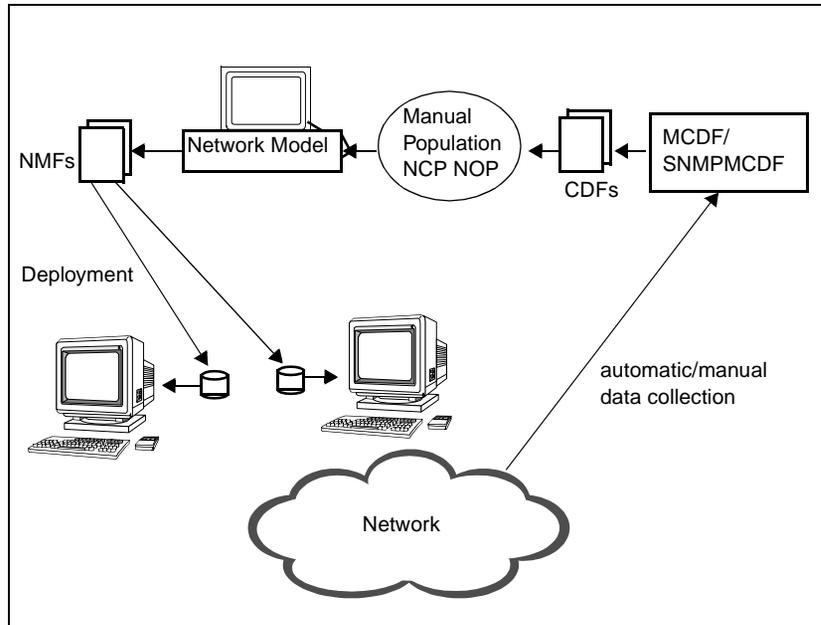
*Note 1:* This method applies only to Passport and DPN switches.

*Note 2:* Make Configuration Data files (MCDF) is supported only on Passport and DPN switches.

Building a model for an existing network involves collecting data from the network using a program called Make Configuration Data File (MCDF) or its Passport 4400 series variant, SNMP Make Configuration Data File (SNMPMCDF). This program creates Configuration Data Files (CDF) which define the topology of the actual network. After you build the physical level of the model, you build organizational structures for the network model. MCDF/SNMPMCDF and Network Viewer editing tools are subsequently used to maintain the model as the network changes.

The figure “Information collection for Network Model” (page 104) shows how data is collected from the network using MCDF/SNMPMCDF and edited to create a Network Model.

**Figure 15**  
**Information collection for Network Model**



## Collecting component data from the network

Configuration Data Files (CDF) are the most efficient way to create a network model. These files are created using the Make Configuration Data File (MCDF or SNMPCMDF) program to collect configuration data from the network. MCDF/SNMPCMDF can be invoked as a macro from the Command Console (CC), an RNCS VT100 access terminal, or a Preside Multiservice Data Manager Xterm window. You can also use a *cron* process to create these files on a regular basis.

**Note:** An alternative is to use the *Collection* dialog available from Network Viewer in edit mode. See “Collecting Network Model data” (page 69) for details.

The workstation running MCDF/SNMPCMDF must be connected to the network, either directly by X.25 if Frame Relay, or by Network Access Service selection to another workstation which is connected to the network.

## Collecting network component information

The process for collecting network component information with MCDF is as follows:

- 1 Log on to the NCS Operations Agent (OA) or Passport (Passport) group that manages the portion of the network you want to model.
- 2 Specify modules to collect data from by running MCDFUTIL, if required. Refer to “Using the MCDFUTIL utility for DPN networks” (page 107) for more information.
- 3 Run MCDF.

MCDF queries the network modules either through operator commands or internal queries and extracts topology and attributes information from these responses.

**Note:** Before collecting network information, ensure you are using an `rncs cron` to connect to a specified OA.

For

a connection to the top level OA lets you create a Network Model which represents the entire DPN network. Data can be collected incrementally by sequentially accessing lower level NCS OAs. Or, MCDF can be run in parallel on any of the lower level OAs from the top level OA.

To access a module, a minimum NCS capability level of *PASSIVE* is sufficient to collect data from DPN-100 modules. For network models, a capability level of *PRIVILEGED* is required. To access a Passport module you need a scope of *NETWORK* and an impact of *PASSIVE*.

For DPN, data is only collected for active network components that are reporting to NCS. Components which have not yet been commissioned are not reflected in the CDFs. Refer to “Creating configuration data files” (page 110) for more information on running the MCDF macro.

For Passport, information is gathered for the entire network depending on how you have partitioned the modules. If you have many groups, you will have to run MCDF on each group and apply the CDFs to the Network Model

one at a time. However, if you create a group for MCDF purposes only, you can include all modules in the group and gather all the information through a single invocation of MCDF.

### **Collecting Passport 4400 component information**

The process for collecting component information for Passport 4400 series of devices with SNMPCDF is as follows:

- 1 Run SNMPCDF.

SNMPCDF queries the data collection daemon (DCD) collected data through GMDR to build up the topology and attribute information. If you do not specify a module in the collection request, SNMPCDF collects information for all Passport 4400 nodes known to GMDR. No on-switch commands are actually issued by SNMPCDF itself.

### **Limitations of MCDF, SNMPCDF, and Auto-MCDF**

If a Passport node is out-of-service during a collection, data cannot be collected from any components under it. You can re-execute MCDF as soon as the OA, module or multiservice switch returns to service.

If a module, component, or OA fails during collection, the data may be incorrect or incomplete. MCDF may have to be re-executed.

Automatic checking for disk space availability is not performed by MCDF, SNMPCDF, or Auto-MCDF. If disk space is insufficient, these utilities terminate.

MCDF sends a significant number of queries to the network to collect data. This may impose extra overhead on network components. MCDF should be run during non-peak hours and on components that are either fully commissioned or are not changing frequently.

Similarly, SNMPCDF will perform a large number of GMDR “getProperty” queries. As these queries are in fact passed on to the DCD process through the SNMP Management Data Router (SMDR), SNMPCDF may therefore cause a sensible degradation of Surveillance capability for the duration of its execution.

## Using the MCDFUTIL utility for DPN networks

This utility is used for DPN networks only.

The MCDFUTIL utility is a macro that lets you specify which modules to collect data from. This permits customization of network models for regional sites. This macro is recommended if you repeatedly collect configuration data from specific modules or OAs which are not all located under one managing OA.

*MCDFUTIL* creates an include list or an exclude list. The include list specifies the mnemonics to be included in an MCDF collection; the exclude list specifies the mnemonics to be excluded from an MCDF collection. You can name the include and exclude lists. These names let you create customized views of the network which are used to maintain different network models. If no names are specified, *mcd.include* and *mcd.exclude* are used by default.

- 1 In the application main window, select System -> Utilities -> Command Console.
- 2 Log on to an NCS operations agent.  
Refer to “How to log on” in 241-6001-011 *Preside MDM Fault Management User Guide*.
- 3 In the Command Input area at the bottom of the Command Console window, enter:

```
mcdfutil
```

The syntax of the *mcdfutil* command is as follows:

```
$mcdfutil [-dexclude [<mnemonic...> | everything]]  
[-dininclude [<mnemonic...> | everything]]  
[-exclude [<mnemonic...>]]  
[-include [<mnemonic...>]]  
[-file [<filename>]]  
[-help]
```

**Note:** The command abbreviations *-de*, *-di*, *-e*, *-i*, *-f* and *-h* can be used instead of typing the full name.

where:

`-dexclude [<mnemonic...> | everything]` deletes one or more mnemonics from the users exclude list. If *everything* is specified, all entries in the exclude list are deleted.

`-dinclude [<mnemonic...> | everything]` deletes one or more mnemonics from the users include list. If *everything* is specified, all entries in the include list are deleted.

`-exclude [<mnemonic...>]` adds one or more mnemonics to the users exclude list. If no exclude list exists, one is created. If no mnemonic is provided, the exclude list (if any) is displayed.

`-include [<mnemonic...>]` adds one or more mnemonics to the users include list. If no include list exists, one is created. If no mnemonic is provided the include list contents (if any) is displayed.

`-file [<filename>]` saves the user's include or exclude entries in the list named by the user. If this option is not used, the default file names are *mcdf.include* and *mcdf.exclude*.

`-help` provides a brief description of MCDFUL and its syntax.

The following examples give you commands for using MCDFUL in specific situations. OA1, OTW8 and REGINA-A1 are used as example mnemonics. OA1 represents an NCS OA; OTW8 is a DPN-100 device reporting to a regular OA; and REGINA-A1 is a DPNNS OA *REGINA*.

### **Example of adding one or more mnemonics to the include lists**

This command is used to add a mnemonic to the default include list:

```
mcdfutil -i OA1 OTW8 REGINA-A1
```

The following output is received:

```
The mnemonic "OA1" has been added to include file  
"mcdf.include"  
The mnemonic "OTW8" has been added to include file
```

```
"mcdf.include"
```

```
The mnemonic "REGINA-A1" has been added to include file  
"mcdf.include"
```

### **Example of deleting one or more mnemonics from the include list**

This command is used to delete a mnemonic from the default include list:

```
mcdful -di OA1 OTW8 REGINA-A1
```

The following output is received:

```
The mnemonic "OA1" has been deleted from include file  
"mcdf.include"  
The mnemonic "OTW8" has been deleted from include file  
"mcdf.include"  
The mnemonic "REGINA-A1" has been deleted from include  
file "mcdf.include"
```

### **Example of displaying the entire include list**

The following command displays the entire include list:

```
mcdful -i
```

The following output will be received:

```
The mnemonics in the include file "mcdf.include" are:  
OA1  
OTW8  
REGINA-A1
```

### **Example of adding a mnemonic to the exclude list**

Enter this command to add a mnemonic to the exclude list:

```
mcdful -e OA1 OTW8 REGINA-A1
```

The following output is received:

```
The mnemonic "OA1" has been added to exclude file  
"mcdf.exclude"  
The mnemonic "OTW8" has been added to exclude file  
"mcdf.exclude"  
The mnemonic "REGINA-A1" has been added to exclude file  
"mcdf.exclude"
```

### **Example of deleting a mnemonic from the exclude list**

Enter this command to delete a mnemonic from the exclude list:

```
mcdful -de OA1 OTW8 REGINA-A1
```

The following output is received:

```
The mnemonic "OA1" has been deleted from exclude file
"mcdf.exclude"
The mnemonic "OTW8" has been deleted from exclude file
"mcdf.exclude"
The mnemonic "REGINA-A1" has been deleted from exclude
file "mcdf.exclude"
```

## Creating configuration data files

The type of Network Model you are currently creating or maintaining determines the type of data you will want to collect with the MCDF utilities. Their command options let you specify the modules and the type of data that is to be collected.

With MCDF, you can collect just backbone information or, by default, collections down to the port level. For collections containing link type information for port components, the *-ql* option must be specified. During the default level collection, you can also request MCDF to collect all Customer Network Management identifications (CNM IDs or CIDs) related to customer access ports in your network.

With SNMPCDF, these options are not currently supported as they do not apply to Passport 4400 series devices.

You can create CDFs from the Command Console window.

**Note:** You must be familiar with the Command Console tool. For more information, see the Command Console section in 241-6001-804 *Preside MDM Workstation Utilities User Guide*.

### Creating CDFs for DPN-100 nodes from the Command Console

- 1 In the application main window, select System -> Utilities -> Command Console.

The Command Console main window opens.

- 2 From the *File* menu choose *Connect*.

The Connection Manager is displayed.

- 3 Select an OA destination and enter the capability ID and password.
- 4 Click *Connect*.

- 5 Run MCDF. In the Command Input area at the bottom of the window, enter:

```
mcdf
```

The syntax of the MCDF for DPN-100 command is as follows (options can be entered by themselves or in any combination):

```
mcdf [-c [<OA_name>]]  
[-b]  
[-cid [-dspcid]]  
[-d <delay>]  
[-o <directory>]  
[-h]  
[-im <mnemonic>...]  
[-em <mnemonic>...]  
[-i [<file_name>]]  
[-e [<file_name>]]  
[-ql]
```

where:

-b collects only backbone information from the network.

-c [<OA\_name>] collects configuration information through a specified Operations Agent (OA).

- If no OA name is specified, the utility tries to retrieve an OA name from a context variable set by the Command Console (CC); if the context variable is not set, that is, no route has been selected in the CC, an error occurs and an appropriate message is printed. If no error occurs, the configuration information is gathered for all modules beneath the OA unless the -e, -i, -im, or -em options are specified.

-cid collects Customer Network Management IDs (CIDs) down to the port level for DPN.

-dspcid collects CID information from 3270 DSP TPAD ports and sets the CID of the port according to the following rules:

- If all TPADs under the port have the same CID, then that CID is assigned to the port as well.

- If one or more of the TPADs has a CID that differs from the rest of the TPAD CIDs, then the port CID is set to zero.

The *-cid* option must also be specified. Using this *-dspcid* option significantly increases the time required to collect data from the targeted module.

*-d <delay>* defines the number of seconds delay between successive commands sent to an OA. The allowable range is 0-9 seconds. The default is 0.

*-e [<file\_name>]* excludes all mnemonics from the collection listed in the exclude file specified. If a file name has not been provided, the default *mcdف.exclude* is used. If a mnemonic provided in the exclude list is an OA mnemonic, all modules reporting to this OA are excluded from the collection.

*-em <mnemonic>* excludes all mnemonics listed in the command line from the collection. If the mnemonic provided is an OA, no data will be collected from any modules under that OA.

*-h* displays help on command syntax and parameter definitions.

*-i [<file\_name>]* collects data from all mnemonics listed in the include file specified. If a file name has not been provided, the default *mcdف.include* is used. If a mnemonic provided in the include list is an OA mnemonic, data from all modules under that OA are collected.

*-im <mnemonic>* includes all mnemonics listed in the command line in the collection. If the mnemonic provided is an OA, all data is collected from any modules under that OA.



**CAUTION**

**Corruption of data**

Before using the *-o* option, ensure that a CDF file does not already exist in the desired output directory since any existing CDF file will be overwritten. Also, ensure that no other user concurrently executes the MCDF command with the same value for the *-o* option to prevent corruption of data.

-o <directory> creates a directory to contain all the collected data. The output from MCDF defaults to */opt/MagellanNMS/bin/data/model/cdf/yymmdd\_hhmmss*. You can specify your own output file name as an absolute path or relative path name. A relative path specification always creates the CDF files in the */opt/MagellanNMS/bin/data/model/cdf* directory.

-ql queries all module ports for link information. If this option is not specified, then the ports are not queried for link information. For example, if -ql is not specified, the link type attribute for port components would be set to dedicated, not dial.

**Note:** To use this option, the Network Model Editing Server (EDSERVER) must be set up with the -q option. For information on setting up the EDSERVER, refer to the 241-6001-310 *Preside MDM Server Reference Guide*.

The following conditions cause MCDF to fail and display a message:

- missing parameters after the -d, -o, -em, or -im options
- specifying mutually exclusive options as noted in the option descriptions

### **Creating CDFs for Passport nodes from the Command Console**

- 1 In the application main window, select System -> Utilities -> Command Console.

The Command Console main window opens.

- 2 From the *File* menu choose *Connect*.

The Connection Manager is displayed.

- 3 Select a group and enter the user ID and password.

- 4 Click *Connect*.

- 5 Run MCDF. In the Command Input area at the bottom of the window, enter:

```
mcdf
```

The syntax of the MCDF command is as follows. The options can be entered by themselves or in any combination.

```
mcdf[-g [<group_name>]]  
[-b]  
[-cid]  
[-d <delay>]  
[-o <directory>]  
[-h]  
[-im <mnemonic>...]  
[-em <mnemonic>...]  
[-u <userid>]  
[-p <password>]  
[-noprogess]
```

where:

`-b` collects only backbone information from the network.

`-cid` collects Customer Network Management IDs (CIDs) down to the port level for eligible components.

`-d <delay>` is the number of seconds delay between successive commands sent to a network element. The allowable range is 0-9 seconds. The default is 1.

`-em <mnemonic>` excludes all mnemonics listed in the command line from the collection.

`-g [<group_name>]` If no group name is specified, the utility tries to retrieve a group name from a context variable set by the Command Console; if the context variable is not set, that is, no route has been selected in the Command Console, an error occurs and an appropriate message is printed. If a group name is specified but the `userid/password` option are not, checks with CM to see if such a connection exists; if not, an error message is displayed. If a name is provided along with a `userid` and `password`, the utility attempts to create its own connection using the specified information; if the logon fails, an error message is displayed. If no error occurs, configuration information is gathered for all modules in the target group unless either the `-em` or `-im` options are specified.

**Note:** To use these options the same `userid` and `password` combination must be valid for all network elements in the specified group.

`-h` displays help on command syntax and parameter definitions.

`-im <mnemonic>` includes all mnemonics listed in the command line in the collection.

`-noprogress` indicates that during a data gathering session, progress messages are not displayed. This option is useful when the utility is run in the background or as a cron job.

**CAUTION****Corruption of data**

Before using the `-o` option, ensure that a CDF file does not already exist in the desired output directory since any existing CDF file will be overwritten. Also, ensure that no other user concurrently executes the MCDF command with the same value for the `-o` option to prevent data corruption.

`-o <directory>` creates a directory to contain all the collected data. The output from MCDF defaults to `/opt/MagellanNMS/bin/data/model/cdf/yymmdd_hhmmss`. You can specify your own output file name as an absolute path or relative path name. A relative path specification always creates the CDF files in the `/opt/MagellanNMS/bin/data/model/cdf` directory. When you run MCDF with your own output directory name, you must ensure this CDF does not already exist.

`-p <password>` is the password for the userid specified by the `-u` option.

`-u <userid>` is the userid for logging on to a group.

The following conditions cause MCDF to fail and display a message:

- specifying the `-u` option without the `-p` option or `-p` without `-u`
- missing parameters after the `-d`, `-o`, `-em` or `-im` options
- specifying mutually exclusive options as noted in the option descriptions

### Creating CDFs for Passport 4400 nodes from the Command Console

- 1 In the application main window, select System -> Utilities -> Command Console.

The Command Console window opens.

- 1 Run SNMPCDF. In the Command Input area at the bottom of the window, enter:

```
snmpcdf
```

The syntax of the SNMPCDF command is as follows. The options can be entered by themselves or in any combination.

```
snmpcdf  
[-h]  
-t <device type>  
[-o <directory>]  
[-im <mnemonic>...]  
[-em <mnemonic>...]
```

Where:

-h

Displays help on command syntax and parameter definitions.

-t <device type>

The type of SNMP devices from which data is collected. Currently, the only accepted value is mpa (Passport 4400).

-o <directory>

**CAUTION****Corruption of data**

Before using the `-o` option, ensure that a CDF file does not already exist in the desired output directory since any existing CDF file will be overwritten. Also, ensure that no other user concurrently executes the MCDF command with the same value for the `-o` option to prevent data corruption.

Creates an output directory for the network model instance file and the log file. You can specify your own output file as an absolute path name or as a relative path name. If you use a relative path name, the output directory is created in your current working directory. The default output directory is based on the time of execution and uses the naming convention `/opt/MagellanNMS/data/model/cdf/yyMMMdd_hh.mm.ss`.

`-im <mnemonic>`

Collects data for only those devices listed on the command line. If this option is not specified, SNMPMCDF collects information on all Passport 4400 devices known to GMDR (excluding those named with the `-em` option, if specified).

`-em <mnemonic>`

Excludes all of the named devices from the collection process.

The following conditions cause SNMP MCDF to fail and display a message:

- missing parameters after the `-t`, `-o`, `-em` or `-im` options
- specifying mutually exclusive options as noted in the option descriptions
- missing `-t` option for SNMPMCDF

## Using the auto MCDF utility for DPN networks

This utility is used for DPN networks only.

The MCDF program can be configured to run automatically on any modules in the network which have had new service data activated.

Hardware and software configuration changes in network modules are implemented through service data activation. The service data activation configuration changes produce alarms with the fault code FFFF 30FF. If your workstation is responsible for monitoring network alarms, you can configure *automcdf* to automatically collect new configuration data from the affected module on reception of the service data activation alarm.

**Note 1:** This data is also applied to the current network model. This new data is resident only in shared memory and must be saved to disk using Network Model Editing tools. See “Saving and distributing Network Model files” (page 99) for details.

**Note 2:** Deletion of network components can only be done manually using Network Model Editing. If the *automcdf* server is used to update the current surveillance model with new configuration data from the network, obsolete components are not removed. CDFs applied by using Network Model Editing are added incrementally to the network model. Network model components which do not exist in the new CDF are not removed from the network model.

**Note 3:** *automcdf* uses the Alarm and Status API Provider to trace the activation alarms. GMDR must be running on the same workstation.

Upon receipt of a service data activation alarm, the server process waits 5.5 minutes for the network module to complete activation. After this the *automcdf* server initiates an *mcdf* process on the affected module. *Automcdf* is configured to connect to only one OA. If a workstation is monitoring alarms from multiple low-level OAs, for *automcdf* to access all modules from these OAs it must have access to a single, common, higher-level OA.

The server process resides in */opt/MagellanNMS/bin/automcdf*. The system administrator must configure this into the *SVM admin tool* using this command and options:

```
/opt/MagellanNMS/bin/automcdf [-b]
-d <ncs mnemonic>
```

where:

-b specifies that only backbone data is to be collected.

-d <ncs mnemonic> specifies an NCS OA mnemonic indicated in the */opt/MagellanNMS/cfg/MCDFauto.cfg* file. You must specify this parameter. The system administrator can modify the file so that it contains any, or all, OAs and the corresponding capability IDs and passwords required by the *automcdf* server.

For information on how to configure an SVM server, refer to the 241-6001-303 *Preside MDM Administrator Guide*.



---

## Chapter 5

# Network model macros and utilities

---

The network model supports several macros and utilities. This section provides a description of these network model macros and utilities.

*Note:* Run the macros and utilities on the same workstation as the network model that supports them.

This section contains the following topics:

- “Available macros and utilities” (page 122)
- “Activating a network model” (page 122)
- “Committing a network model” (page 124)
- “Displaying the committed and active model” (page 125)
- “Displaying available network models” (page 125)
- “Saving a network model in portable format” (page 125)
- “Saving a network model in quick format” (page 126)
- “Saving selected components and subcomponents in portable format” (page 126)
- “Converting between quick and portable formats” (page 128)
- “Purging unknown components from the network model” (page 129)
- “Putting components into maintenance state” (page 131)
- “Putting components into acknowledged state” (page 132)
- “Copying a network model fileset to other workstations” (page 132)

## Available macros and utilities

The following macros and utilities are available:

- `makecurrent` - to activate a network model, see “Activating a network model” (page 122)
- `commitmodel` - to commit a network model, see “Committing a network model” (page 124)
- `whichcurrent` - to display the active and committed network models, see “Displaying the committed and active model” (page 125)
- `whichmodels` - to display the available network models, see “Displaying available network models” (page 125)
- `savemodel` - to save the network model in portable (ASCII) format, see “Saving a network model in portable format” (page 125)
- `dumpmodel` - to save the network model in quick (binary) format, see “Saving a network model in quick format” (page 126)
- `popsave` - to save selected components and subcomponents in portable format, see “Saving selected components and subcomponents in portable format” (page 126)
- `popconv` - to convert between quick and portable network model formats, see “Converting between quick and portable formats” (page 128)
- `poppurge` - to purge from the model components in an unknown state, see “Purging unknown components from the network model” (page 129)
- `maintenance` - to put components into maintenance state, see “Putting components into maintenance state” (page 131)
- `acknowledge` - to put components into acknowledged state, see “Putting components into acknowledged state” (page 132)

## Activating a network model

You can make current, or activate, a network model in various ways. You can activate a model using the Network Viewer model editor. You can also activate a model using the `makecurrent` macro from the Command Console tool or from a UNIX window.

## Activating a network model using the Network View model editor

Use this procedure from the Network Viewer edit mode.

- 1 From the Network Model Edit menu, select Load/Change Network Model File.

The Network Model File Manager dialog opens listing the available network model filesets.

- 2 Select the model fileset that you want to make current from the list and then click the Load button.

The model file you selected is now the current model in the Network Viewer edit mode.

## Activating a Network Model using the makecurrent macro

The makecurrent macro allows you to activate a model without having to be in the Network Viewer model editor. You can execute the makecurrent macro from the Command Console tool or from a UNIX window. The makecurrent macro is in the directory */opt/MagellanNMS/cfg/macros/nms*.

### Activating a Network Model from the Command Console tool

- 1 In the application main window, select System -> Utilities -> Command Console.

The Command Console window opens. Depending on your current connections, you may see the Command Console Connection Management dialog. If so, complete the dialog with the appropriate responses and click the Connect button to open the Command Console window.

- 2 Enter the following command on the command line:

```
$makecurrent <nmffileset>
```

where <nmffileset> is the name of the network model fileset you want to make current.

### Activating a Network Model from a UNIX window

- 1 In the application main window, select System -> Utilities -> UNIX Access.
- 2 Enter the following pathname and macro:

```
/opt/MagellanNMS/cfg/macros/nms/makecurrent  
<nmffileset>
```

where <nmffileset> is the name of the network model fileset you want to make current. The network update process uses the new network model fileset (NMF) as the network model

## Committing a network model

You can commit a model to save it as the start-up model in various ways. You can commit a model using the Network Viewer model editor. You can also commit a network model using the commitmodel macro from the Command Console tool; or, you can use the commitmodel macro from a UNIX window.

### Committing a network model using the Network View edit mode

Use this procedure from the Network View edit mode.

- 1 From the Network Model Edit menu, select Save Network Model.  
The Network Model File Save dialog opens listing the available network model filesets.
- 2 Select the model fileset that you want to commit.
- 3 Select the option Make this model the default on start-up.  
The model file you selected is now the committed model.

### Committing a network model using the commitmodel macro

The commitmodel macro allows you to commit a model without having to be in the network model editor. You can execute the commitmodel macro from the Command Console tool or from a UNIX window. The commitmodel macro is in the directory */opt/MagellanNMS/cfg/macros/nms*.

#### Committing a network model from the Command Console

- 1 In the application main window, select System -> Utilities -> Command Console.  
The Command Console window opens. Depending on your current connections, you may see the Command Console Connection Management dialog. If so, complete the dialog with the appropriate responses and click the Connect button to open the Command Console window.
- 2 Enter the following macro on the command line:  

```
$commitmodel <nmffileset>
```

where <nmffileset> is the name of the network model fileset you want to make current.

### **Committing a network mode from a UNIX window**

- 1 In the application main window, select System -> Utilities -> UNIX Access.
- 2 Enter the following pathname and macro:

```
/opt/MagellanNMS/cfg/macros/nms/commitmodel  
<nmffileset>
```

where <nmffileset> is the name of the network model fileset (NMF) you want to make current.

The next time the SURNUP process is restarted and there is no active model, SURNUP uses the new NMF as the start-up network model (typically at reboot).

## **Displaying the committed and active model**

Use the whichcurrent macro to display the committed model and the active network model. The committed model is used, by default, when SURNUP is restarted. The current network model is the one being used for network surveillance by the Network Viewer, Component Status Display and Component Information Viewer. The whichcurrent macro is in the directory */opt/MagellanNMS/cfg/macros/nms*.

## **Displaying available network models**

Use the whichmodels macro to display all the available network models that are found under */opt/MagellanNMS/data/model/nmf*. The whichmodel macro is in the directory */opt/MagellanNMS/cfg/macros/nms*.

## **Saving a network model in portable format**

Use the savemodel macro to save the current network model to disk in portable (ASCII) format. The savemodel macro is in the directory */opt/MagellanNMS/cfg/macros/nms*.

The format of the savemodel macro is

```
/opt/MagellanNMS/cfg/macros/nms/savemodel  
<nmffileset>
```

where

`nmffileset` is the name of the network model fileset you want to save in portable format.

**Note:** When integrating new devices into Preside Multiservice Data Manager, it is recommended that the network model be saved as ASCII format. Otherwise the new network model files will not take effect.

## Saving a network model in quick format

Use the `dumpmodel` macro to save the model in quick (binary) format. The `dumpmodel` macro is in the directory `/opt/MagellanNMS/cfg/macros/nms/dumpmodel`.

The format of the `dumpmodel` macro is

```
/opt/MagellanNMS/cfg/macros/nms/dumpmodel  
<nmffileset>
```

where

`nmffileset` is the name of the network model fileset you want to save in quick format.

**Note:** When integrating new devices into Preside Multiservice Data Manager, it is recommended that the network model be saved as ASCII format. Otherwise the new network model files will not take effect.

## Saving selected components and subcomponents in portable format

Use the `popsave` utility to save a selected set of current network model components and their subcomponents in portable format. Specify the selected set of components in a file. You can specify a complete name such as `PM R66`, or you can use a regular expression like `FNMOD *` or `EM A*`. The `popsave` utility is in the directory `/opt/MagellanNMS/bin`.

The format of the popsave utility is

```
/opt/MagellanNMS/bin/popsave  
[-f <module expressions file name>]  
[-d <model directory> | -r <model name>]  
[-c <number>] [-sianbloh]
```

where

- f saves only those components that are specified in the filter file (ASCII format only)
- d dumps the current model to the specified directory
- r dumps and renames the current model
- c saves a customer id filtered version (ASCII format only)
- s commits the dumped model
- i dumps the model in quick (binary) format
- a dumps the model in portable (ASCII) format (default)
- n dumps the DPN.proc file
- b does not back up the model if it already exists
- l when used with the -f option, saves the links that are associated with the filtered components
- o when used with the -f option, saves the organizations that are associated with the filtered components
- h displays popsave help information

### Example

A sample file is as follows:

```
PM R66  
FNMOD *  
EM A*
```

If you use the Example 1 file with the `popsave` utility, `popsave` saves the PM R66 module, all the FNMOD modules, and all the EM modules with a name starting with the letter A.

**Example**

To save the modules specified in file `comps` and all their related links, use the following command:

```
/opt/MagellanNMS/bin/popsave -f comps -l
```

**Example**

To save all the organizational structures related to the selected modules, use the `-o` option. Saving the organizational structures will also save the positions of the modules and of their related links. To save the modules specified in file `comps` and all their related organizations and positions, use the following command:

```
/opt/MagellanNMS/bin/popsave -f comps -o
```

**Example**

To save the selected modules, their related links and their organizational structures, use the following command:

```
/opt/MagellanNMS/bin/popsave -f comps -l -o
```

Use the `-d` option to specify where to save the components model information. If you use the `-d /opt/MagellanNMS/data/model/nmf/<your_model_name>` option, then the model represents a regular network model file. The standard load model capability of Network Viewer may be used to load such a model. For additional information see “Loading a Network Model” (page 65).

If you use the `-d /opt/MagellanNMS/data/model/cdf/<your_model_name>` option, then the model represents a collection data file. The standard load collection capability of Network Viewer may be used to increment a current model with the saved components, links and organizational structures. For additional information see “Managing collections and applying collected data to the Network model” (page 72).

## Converting between quick and portable formats

Use the `popconv` utility to convert between quick and portable model formats without affecting surveillance. The `popconv` utility is in the directory `/opt/MagellanNMS/bin`.

The format of the popconv utility is:

```
/opt/MagellanNMS/bin/popconv  
[-r]  
[-n model_name | -d directory]  
[-s shmem_size] [h]
```

where

- r reverses the conversion by converting from quick to portable format
- n specifies the model to restore
- d overrides the /opt/MagellanNMS/data/model/nmf for the model
- s indicates the shared memory size in MB
- h displays popconv help information

Use the *-n <fileset name>* command line option to convert from portable format to quick format; use the *-r -n <fileset name>* command line option to convert from quick format to portable format. If your model is bigger than 24 MB and loaded in memory, you should add a *-s <shmem size in MB>* option similar to that specified for the DNMNMC server.

## Purging unknown components from the network model

Use the poppurge utility to purge (remove) those components in the network model that have an unknown state. The poppurge utility is in the directory */opt/MagellanNMS/cfg/macros/nms/poppurge*.

The format of the poppurge utility is

```
/opt/MagellanNMS/cfg/macros/nms/poppurge  
[-lmnxh]  
[-s <number of days>]  
[-t <fileName>]  
[-d <fileName>]  
[-c <criticality>]
```

where

- l removes links

- m removes modules
- n prints to standard output the names of the modules and sub-components to be deleted without deleting them (includes -x by default)
- x does not save the current time as the last update
- h displays poppurge help information
- s removes components that have been in an unknown state since the specified number of days
- t removes only those components in an unknown state that match the type filterset in the specified file
- d removes only those components in an unknown state that do not match the device filterset in the specified file
- c removes only those unknown components with a criticality lower than that specified

If you do not specify any options with the poppurge utility, the poppurge utility removes those subcomponents that have been in an unknown state since the last time poppurge was run or since the last synchronization with GMDR.

### **Example**

A sample type filterset is as follows:

```
EM
EM-*
PTK;
```

If you use the Example 1 type filterset file with the -t option and you also use the -l option, poppurge removes only those components and PTK links that are in an unknown state.

**Example**

A sample device filterset is as follows:

```
EM MICKEY
```

If you use the Example 2 device filterset file with the `-d` option, `popurge` deletes all modules except `EM/MICKEY` and all of its subcomponents.

**Example**

If you use both the Example 1 type filterset and the Example 2 device filterset, `popurge` deletes all the components in an unknown state but keeps `EM/MICKEY` and all of its subcomponents.

A component might not be removed if it fails on any other test, for example if the component is not in an unknown state or if there is a criticality specified.

## Putting components into maintenance state

Use the maintenance macro to put components into maintenance state. For example, you can put components into maintenance at workstation boot time to hide the state of known components. The maintenance macro is in the directory `/opt/MagellanNMS/cfg/macros/nms`. Use the maintenance macro from a Unix Access window.

The format of the maintenance macro is

```
/opt/MagellanNMS/cfg/macros/nms/maintenance  
[-i|-o] [-eh]
```

where

- i put components into maintenance state (default)
- o takes components out of maintenance state
- e echoes what maintenance state changes occur
- h displays maintenance help information

The maintenance macro reads component IDs (one per line) and puts them into maintenance state (-i) or takes them out of maintenance state (-o). If you specify the `-e` option, the maintenance macro echoes the names of those components successfully handled. To terminate component input, type an

end-of-file character (Control-d). It is useful to put the names of the components in a data file and use that file with the command. For example, here is how the maintenance macro can be used from the command console:  
*\$maintenance -i < /localdisk/<user>/maint.comps.*

## Putting components into acknowledged state

Use the `acknowledge` macro to put components into acknowledged state. The `acknowledge` macro is in the directory */opt/MagellanNMS/cfg/macros/nms*. Use the maintenance macro from a UNIX Access window.

The format of the `acknowledge` macro is

```
/opt/MagellanNMS/cfg/macros/nms/acknowledge  
[-i | -o] [-eh]
```

where

- i put components into acknowledged state (default)
- o takes components out of acknowledged state
- e echoes what acknowledge state changes occur
- h displays acknowledge help information

## Copying a network model fileset to other workstations

You may want to produce a network model fileset (NMF) on one workstation and copy it to other workstations.

The NMF for each network is stored in the directory */opt/MagellanNMS/data/model/nmf*. There is one directory for each NMF and the name of each directory is the name of the NMF. When you copy an NMF to other workstations, you must copy all the files along with the directory.

**CAUTION****Interruption of service**

Binary format network model files, stored in *instances.image* files, are not portable across workstations or Preside Multiservice Data Manager versions. Remove these binary files before the tar command to create, or immediately after the tar command to extract from, a tape archive file. Alternatively, you can execute the tar command with the -X option to exclude the *instances.image* files. If the *instances.image* files are not removed, the model will fail to reload on the destination workstation.

## Setting up the remote workstations

- 1 Create a tape archive (*tar*) file called *nmf.tar* at the central workstation, by entering:

```
cd /opt/MagellanNMS/data/model/nmf
tar cvf nmf.tar <dir_nmf>
```

where:

*/opt/MagellanNMS/data/model/nmf* is the home directory of the user who creates the network model fileset

*<dir\_nmf>* is the name of the directory which contains the NMF being distributed.

**Note:** Several models may be transferred at the same time by typing the following command:

```
tar cvf nmf.tar <dir_nmf1> <dir_nmf2>
```

This puts all files into one tar file.

- 2 Move the file to another workstation using the UNIX file transfer protocol (FTP) or the UNIX tape archive program (tapetool).
- 3 Extract the NMF from *nmf.tar* by entering:

```
cd /opt/MagellanNMS/data/model/nmf
tar xvf /opt/MagellanNMS/data/nmf.tar
```

- 4 Remove the file *nmf.tar* by entering:

```
rm /opt/MagellanNMS/data/nmf.tar
```

- 5 When the file has been extracted, use the macros *makecurrent* and/or *commitmodel* *<nmffileset>*.

**Note:** You can use the *popconv* program to convert the transferred model into binary format before loading it, to minimize surveillance disruptions.

## Using tapes to transfer files between workstations

- 1 Insert a write-enabled tape into the tape drive can access the source files.
- 2 Create a tape archive (tar) file by entering the following commands from a *UNIX Access* window:

```
cd /opt/MagellanNMS/data/model/nmf
tar cvf /dev/rst0 <nmffileset>
```

where

`/opt/MagellanNMS/data/model/nmf` is the source directory containing the files to be transferred.

`<nmffileset>` is the name of the tar file

- 3 Eject the tape from the tape drive and insert the tape into the tape drive that can access the target destination.
- 4 Enter the following commands to extract the tar file from the tape:

```
cd /opt/MagellanNMS/data/model/nmf
tar xvf /dev/rst0
```

where:

`/opt/MagellanNMS/data/model/nmf` is the target directory that is to receive the files.

**Note:** Similar procedures can be used to transfer a model using diskettes.

---

## Appendix A

# Customizing the Network Model Schema

---

The Preside Multiservice Data Manager model schema defines the component types that can be modeled, the informational attributes of these components, how these components can be inter-connected through links, and how these components can be organized into an organization structure.

This section covers the following topics:

- “Network model Types files” (page 135)
- “Customizing the network model schema” (page 136)
- “Customizing the Network Viewer” (page 149)

### Network model Types files

The definitions in the Preside Multiservice Data Manager (MDM) network model schema are stored in a number of ASCII configuration files called network model Types files. These files are specific to, and delivered with, each new MDM release. The following list describes each Types file:

- */opt/MagellanNMS/lib/model/types/types.atdf*

The attribute types definition file defines what types of attributes can be assigned to module, subcomponents and links.

- */opt/MagellanNMS/lib/model/types/types.mtdf*

The module and subcomponent types definition file defines the types of components that can be created in the model, how they are contained in one another and what attributes they support.

- */opt/MagellanNMS/lib/model/types/types.ltdf*

The link types definition file defines the types of links that can be created in the model, what are their legal endpoints, and what attributes they support.

- */opt/MagellanNMS/lib/model/types/types.otdf*

The organization types definition file defines what types of organization structures can be created in the model and what types of modules and links they are allowed to organize.

You can customize the network model schema by using second-party customized files or by creating your own customized files. The second-party customized files are created by other Nortel Networks groups and are available in the */opt/MagellanNMS/ext/lib/model/types/* directory.

If you create your own version of the schema files, use the same extensions as described in the preceding list (*.atdf*, *.mtdf*, *.ltdf*, and *.otdf*) and store the files in the */opt/MagellanNMS/data/model/types/* directory. Your customized *.types* files are incremental and need only contain the description of entities added or modified. The resulting model schema is the union of all three sets of schema files. Your customized *.types* file are loaded last so they override the specification of the other areas.

## Customizing the network model schema

You can customize the Preside Multiservice Data Manager (MDM) network model schema. The allowed changes are as follows:

- adding new attribute types
- adding new module, subcomponent, and link types
- adding new organization structure types
- changing the criticality, explanation and label of existing module types
- adding new legal attributes to existing module types.
- changing the criticality, explanation, sub-type and label of existing subcomponent types.
- adding new legal attributes to existing subcomponent types.

- changing the criticality, explanation and label of existing link types
- adding new legal attributes to existing link types

If you customize the network model schema, the changes are stored in separate Types files from those provided by MDM. These files are as follows:

- `/opt/MagellanNMS/data/model/types/<file name>.atdf`

The customized attribute types definition file.

- `/opt/MagellanNMS/data/model/types/<file name>.mtdf`

The customized module and subcomponent types definition file.

- `/opt/MagellanNMS/data/model/types/<file name>.ltdf`

The customized link types definition file.

- `/opt/MagellanNMS/data/model/types/<file name>.otdf`

The customized organization types definition file.

There can be multiple files of each type. All of these files are loaded as part of the schema. Some schema changes impact multiple files.

- Adding new attribute types in the types.atdf file implies adding their names to the appropriate module, subcomponent and link legal attributes lists in types.mtdf and types.ltdf.

**Note:** In previous versions of the Network Model Schema, you could restrict the model by specifying allowed name patterns for modules and subcomponents, legal endpoint types for links, and the allowed module and link types for an organizational structure type. Although it is still possible to define legal names and link endpoint types, it may not be the case in the future. The current delivered schema does not use these definitions. You can now give modules and subcomponents almost any name. Links of any type can now be created between any ENDPOINT subcomponents. Finally, all organization structure types can contain instances of all module and link types.

After you make any changes, you need to reload the schema changes. To do so, reload the current model or, if appropriate, re-create the model in the usual manner. Models loaded from an image file (Fast Load format) do not use the modified types because the types they were built with are contained in the image. To access the new types, you need to reload the model from its ASCII/portable format. If the type definitions cannot be read, the model does not load and an error message is sent to the Log display. It is recommended that you use the *makecurrent* macro (*/opt/MagellanNMS/cfg/macros/nms/makecurrent <model name>*) when loading a model after changing the schema. More diagnostic information may be displayed.

Alternatively, you can load the schema changes without reloading the model. This method has the advantage of being non-disruptive to surveillance. To load the schema without reloading the model, use the following command:

```
/opt/MagellanNMS/bin/poprest -u
```

where

-u specifies the update option.

If the schema changes cannot be loaded using this method, error messages display and the current model is not validated.

The following sections describe the format and meaning of the individual Types files. Note that these formats are basically the same as the master types files. It is therefore useful to look at them for examples of how to define a Network Model Schema type. The section that follows demonstrates this capability through examples. Finally, the last section describes how the Network Viewer can be customized to display new node types with their own type icons.

## The Attribute Definition File

The Network Model customized attribute types definition file is named /opt/MagellanNMS/data/model/types/<file name>.atdf. If it does not exist already, it should be created and edited to allow the customization of the Network Model informational attributes. The syntax of the atdf file is as follows:

```
Attribute_type: <attribute name>
Value_type: <INTEGER or STRING>
[Subtype: <USES or USED_BY>]
Default_value: <default value>
[Range: <lower> <upper>]
[Legal_values: <legal string value>]*
Label: <label string>
Verbatim_start: Explanation:
<explanation>
Verbatim_end: Explanation:
<blank line>
```

where:

`<attribute name>` is the internal/API name for the attribute (all upper-case)

`Value_type` indicates the syntactic type for the attribute (INTEGER or STRING)

`Subtype` is used to indicate that the attribute is a component name either used by or using the component that hold this attribute (optional).

`Default_value` is the default integer or string value for the attribute (must be specified but may be empty).

`<lower>` `<upper>` indicate the legal value range for INTEGER attributes only (optional).

`Legal_values` can occur multiple time to indicate the possible values for a STRING attribute (must be specified but may be empty).

`<label string>` is the user interface name for the attribute

`<explanation>` is a text explaining the specification of the attribute.

### **Example of New PHONE\_NB attribute type**

```
Attribute_type: PHONE_NB
Value_type: STRING
Default_value:
Legal_values:
Label: Phone Number
Verbatim_start: Explanation:
Contact's phone number (111-nnnn).
verbatim_end: Explanation:
```

The only customization allowed in this file is to create new attributes. Any redefinition of an existing attribute type will be ignored.

## The Module/Subcomponent Definition File

The Network Model customized module and subcomponent types definition file is named `/opt/MagellanNMS/data/model/types/<file name>.mtdf`. If it does not exist already, it should be created and edited to allow the customization of the Network Model components. The syntax of the mtdf file consists of two types of records, module type definitions and subcomponent type definitions. A module type definitions must precede in the file that of its subcomponent types. The syntax of a module type definition is:

```
Module_type: <module type>
Criticality: <criticality>
[State: IN_SERVICE]
[Flag: BACKBONE]
[Legal_name: <legal name>]..*
Label: <label string>
Verbatim_start: Explanation:<explanation>
Verbatim_end: Explanation:
[Info_attribute: <attribute name>]*
<blank line>
```

where:

`<module type>` is the module type name (all in uppercase).

`<criticality>` is a number from 1 to 5 indicating the module's default criticality.

`State: IN_SERVICE` identifies a constant state component (optional).

`Flag: BACKBONE` indicates that this module is a backbone component (modules are always considered backbone) (optional).

`Legal_name` can occur multiple times to indicate the legal values for the module name (in grep regular expression format).

`<label string>` is the user interface name for the module type.

`<explanation>` is a text explaining the specification of the module name.

`<attribute name>` names the legal information attributes on this component (optional). This attribute can occur multiple times. Note the special attribute `NODE_PIXMAP` which is specified as follows:

`Info_attribute: NODE_PIXMAP <path or name of pixmap>`

This attribute specifies the icon pixmap used by the Network Viewer to display instances of that module type. Specify the pixmap as a full path name or use the pixmap file name. If you use the pixmap file name Viewer searches for it first in `/opt/MagellanNMS/data/nds/pixmaps` (the recommended location for your own customized icons), then in `/opt/MagellanNMS/ext/lib/nds/pixmaps` (second-party customizations), and finally in the delivered pixmaps in `/opt/MagellanNMS/lib/nds/pixmaps`.

### Example of New MODEM component type

```
Module_type: MODEM
Criticality: 3
Flag: BACKBONE
Legal_name: ^[A-Z][A-Z0-9_-]*$
```

```
Label: MODEM
Verbatim_start: Explanation:
The legal names for a MODEM are composed of letters and
numbers and the characters '-' and '_'. The name must
start with a letter and be no more than 12 characters
long.
Verbatim_end: Explanation:
Info_attribute: PHONE_NB
```

If an existing module type is being modified, only the *Module\_type*, *Criticality*, *Legal\_name*, *Label*, *Explanation*, and *Info\_attribute* are recognized. If a new module type is added, its type should be added to the appropriate organization types in types.otdf. The *Criticality*, *Label*, and *Explanation* fields need not be specified when modifying an existing type.

The syntax of a subcomponent type definition is:

```
Subcomponent_type: <subcomponent type>
In_module: <module type>
[Parent: <module/subcomponent type>]+
Criticality: <criticality>
Subtype: <ORDINARY, DYNAMIC, OPTIONAL or ENDPOINT>
[Flag: BACKBONE]
[Legal_name: <legal name>]*
Label: <label string>
Verbatim_start: Explanation:
<explanation>
Verbatim_end: Explanation:
[Info_attribute: <attribute name>]*
<blank line>
```

where:

<subcomponent type> is the subcomponent type name (all in uppercase).

<module type> is the parent module type name

Parent must occur at least once to name the direct parent module or subcomponent types. A subcomponent type can occur under different parents of the same module type.

<criticality> is a number from 1 to 5 indicating the subcomponent's default criticality.

Subtype indicates the type of subcomponent (see below)

Flag: BACKBONE is used to indicate if the subcomponent should be considered a backbone type component for the module (optional).

Legal\_name can occur multiple times to indicate the legal values for the subcomponent name (in grep regular expression format) (optional for DYNAMIC subcomponents).

<label string> is the user interface name for the subcomponent type.

<explanation> is a text explaining the specification of the subcomponent name (optional for DYNAMIC subcomponents).

<attribute name> can occur multiple times to name the legal informational attributes on this component (optional).

### **Example of New M\_PO subcomponent on a MODEM**

```
Subcomponent_type: M_PO
In_module: MODEM
Parent: MODEM
Subtype: ENDPOINT
Criticality: 2
Legal_name: ^[0-1]$
Label: MODEM Port
```

```
Verbatim_start: Explanation:  
The legal names for a MODEM port are numbers in the  
ranges 0-1.  
Verbatim_end: Explanation:
```

or (change the default PO criticality to 1):

```
Subcomponent_type: PO  
In_module: PM  
Parent: PI  
Criticality: 1
```

The *Criticality*, *Subtype*, *Label*, and *Explanation* fields need not be specified when modifying an existing type, but the appropriate *In\_module* and *Parent* fields must. It is not possible to define differently the same subcomponent types if found under several different parent subcomponent types of the same module (though it is possible to differentiate the same subcomponent type found in different modules).

Subcomponent types come in four possible sub-types:

- **ORDINARY** indicates a basic subcomponent that can be configured from NCP, and is maintained permanently in the model.
- **ENDPOINT** indicates an **ORDINARY** subcomponent type that can also be used as a link endpoint.
- **DYNAMIC** indicates a subcomponent type whose instances will only appear in the model if their state is troubled or out-of-service (they are automatically removed otherwise).
- **OPTIONAL** indicates a subcomponent type that can be configured through NV editing like an **ORDINARY** subcomponent and therefore be kept permanently. If instances of this type are created through auto-population (their state is troubled or out-of-service) they will behave like **DYNAMIC** subcomponents. **OPTIONAL** subcomponents are usually too numerous to be modeled permanently, however some of them are part of the model because they carry diagnostic information, usually through informational attributes.

## The Link Definition File

The Network Model customized link types definition file is named `/opt/MagellanNMS/data/model/types/<file name>.ltdf`. If it does not exist already, it should be created and edited to allow the customization of the Network Model links. The syntax of the ltdf file is as follows:

```
Link_type: <link type>
Criticality: <criticality>
[Subtype: DYNAMIC]
Label: <label string>
Verbatim_start: Explanation:
<explanation>
Verbatim_end: Explanation:
[Info_attribute: <attribute name>]*
<blank line>
```

where:

`<link type>` is the link type name (all in uppercase).

`<criticality>` is a number from 1 to 5 indicating the link's default criticality.

`Subtype: DYNAMIC` if specified (optional), indicates that the link is automatically removed if its state is set to unknown

`<label string>` is the user interface name for the link type.

`<explanation>` is a text explaining the specification of the link.

`<attribute name>` can occur multiple times to name the legal information attributes on this link (optional).

### Example of A new CBL link for MODEM links

```
Link_type: CBL
Criticality: 3
Label: MODEM Cable
Verbatim_start: Explanation:
MODEM Cables are legal between DPN-100 ports and MODEM
module ports.
Verbatim_end: Explanation:
```

If a new link type is added, its type should be added to the appropriate organization types in types.otdf. The *Criticality*, *Label*, and *Explanation* fields need not be specified when modifying an existing type.

## The Organization Definition File

The Network Model customized organization types definition file is named /opt/MagellanNMS/data/model/types/<file name>.otdf. If it does not exist already, it should be created and edited to allow the customization of the Network Model organizations. The syntax of the otdf file is as follows:

```
Organization_type: <org type>
Organizational_levels: <level label list>
Sub_module_level: <component level label>
```

where:

<org type> is the organization type name (all in uppercase).

<level label list> is a blank separated list of labels for the various (3) levels of the organization structure (unused).

<component level label> is the label for the subcomponent level (unused).

### Example of creating a new organization type:

```
Organization_type: CORENETWORK
```

Organizations can only have two levels which are always called REGION and SITE.

### Example of changing the Criticality of an Existing Component:

To modify the default criticality of the PO subcomponent to 1, create the /opt/MagellanNMS/data/model/types/types.mtdf file with the following contents (do not forget the blank line after the record):

```
Subcomponent_type: PO
In_module: PM
Parent: PI
Criticality: 1
```

**Example of creating a New Attribute Type and Supporting it on a Component**

To add a new PHONE\_NB attribute to the PM components, create the */opt/MagellanNMS/data/model/types/types.atdf* file with the following contents:

```
Attribute_type: PHONE_NB
Value_type: STRING
Default_value:
Legal_values:
Label: Phone Number
Verbatim_start: Explanation:
Contact's phone number (l11-nnnn).
Verbatim_end: Explanation:
```

and create the */opt/MagellanNMS/data/model/types/types.mtdf* file with the following contents:

```
Module_type: PM
Info_attribute: PHONE_NB
```

**Example of creating a New Module, Subcomponent, and Link Type**

To create a new MODEM module type and PORT subcomponent type with use of a new CBL link type to network element ports, create the */opt/MagellanNMS/data/model/types/types.mtdf* file with the following contents for the new module and subcomponent types:

```
Module_type: MODEM
Criticality: 3
Flag: BACKBONE
Legal_name: ^[A-Z][A-Z0-9_-]*$
Label: MODEM
Verbatim_start: Explanation:
The legal names for a MODEM are composed of letters and
numbers and the characters '-' and '_'. The name must
start with a letter and be no more than 12 characters
long.
Verbatim_end: Explanation:
Info_attribute: INFO

Subcomponent_type: PORT
In_module: MODEM
Parent: MODEM
```

```
Subtype: ENDPOINT
Criticality: 2
Legal_name: ^[0-1]$
Label: MODEM Port
Verbatim_start: Explanation:
The legal names for a MODEM port are numbers in the
ranges 0-1.
Verbatim_end: Explanation:
```

and create the file `/opt/MagellanNMS/data/model/types/types.ltdf` with the following contents for the new link type:

```
Link_type: CBL
Legal_endpoints: MODEM-PORT {PM-PO,EM-V35,EM-X21}
Label: MODEM Cable
Verbatim_start: Explanation:
MODEM Cables are legal between DPN-100 and Passport X
series ports and MODEM ports.
Verbatim_end: Explanation:
```

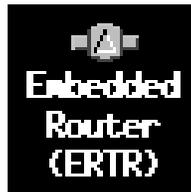
## Customizing the Network Viewer

The Network Viewer displays modules in one of two different ways: a type (or flag) icon, and a Shelf icon used in the Shelf Dialog (as this icon needs extensive code support, it is not possible to create a Shelf Dialog icon for customized module types). Modules of a custom created type are usually displayed with the Unknown Type flag icon.



This section describes how to configure the Network Viewer to display these custom modules with their own distinct type icons.

Node icons in the Network Viewer are defined as XPM3 pixmaps. Each pixmap is identified by a resource line. Another resource line identifies the module type the icon corresponds to. The following example demonstrates this through the Embedded Router type icon (ERTR node type)



This icon, whose shape is reminiscent of a sphere with a pipe going through it, has a triangle inset mark to indicate its relationship with an Access Module (whose general shape is also a triangle). The icon has a 3-D appearance due to the top and bottom drop shadows and is colored according to the current node state. The shape of the icon can also be quite complex (as XPM pixmaps support the so-called transparent color).

## Creating custom node icon pixmaps

An ideal tool for working with XPM pixmaps is the CDE Icon Editor tool (dticon). What makes this tool so suited to the task of creating node icons is the ability to assign symbolic colors (called Dynamic Colors in the Icon Editor color palette) which is crucial to node icons.

Node icons change color according to the component state recorded in the network model. Rather than creating a separate pixmap file for each possible node state color, only one pixmap file needs to be created using symbolic colors. The symbolic colors are then substituted with actual (state-associated) colors at run-time.

Since all pixmaps in the X window system are rectangles, non-rectangular node icon shapes are achieved by using the special symbolic color “Transparent”. Wherever a transparent pixel is defined in the node icon pixmap, nothing is drawn on the screen. This allows whatever graphic exists “behind” the node icon to show through.

When using the Icon Editor to create a new node icon, first use the symbolic “Background” color to fill in a silhouette of the node icon’s shape. Then use “Top Shadow” and “Bottom Shadow” symbolic colors to give the icon a 3-D appearance. This is usually done by replacing the outer-most row of pixels that make up the outer edge of the silhouette. Top shadows and bottom shadows are also useful for enhancing the 3-D appearance of the icon, as has been done with the Embedded Router icon so that the pipe appears to be separate from the sphere rather than appearing as a single integrated piece. The top and bottom shadows are also useful to create the indented appearance of the triangle mark on the icon.

Top Shadows are applied to the top and left sides of an icon. Bottom shadows are applied to the bottom and right sides of an icon. Where these shadows meet, if one pixel could be assigned to either the top or bottom shadow color, the top shadow should be used (e.g., the top-right and bottom-left corners of a rectangle).

After all background, top shadow, and bottom shadow pixels have been assigned, use the Select color to draw a 2-pixel wide border immediately outside the node’s silhouette (no gaps). In the Network Viewer, these pixels will only appear when the node is selected, and will be drawn using the viewer’s Select color.

Next, fill all pixels not already filled with Background, Top Shadow, Bottom Shadow, and Select symbolic colors with the Transparent symbolic color.

At this point, if desired, additional solid colors can be added to the icon to enhance its detailed appearance. When doing this, avoid using colors that are indistinguishable from the state colors. Otherwise carefully crafted details may be “washed out” when the node is set to a state whose color is close to the color used to represent some detail.

Finally, make sure the graphic you have drawn is located as close as possible to the upper left corner of the Icon Editor canvas, and resize the canvas so that the icon just fits without losing or wasting any pixels. Then you can save the final version of your custom node icon.

After you create the icon, you can associate its path with a component type by customizing the corresponding module type in the network model schema (if not already done) and providing a description in a line such as

```
Info_attribute: NODE_PIXMAP <pixmap file path>
```

For a description of how to customize the module types, see “The Module/Subcomponent Definition File” on page 141.

To create a customized icon for your module types, use the following procedure.

### Creating a customized icon

- 1 Decide on a commonly accessible directory to store your icon pixmaps. The `/opt/MagellanNMS/data/nds/pixmaps` directory is available for your use.
- 2 Copy an existing icon pixmap (one that corresponds roughly to the shape you have in mind) from the `/opt/MagellanNMS/lib/nds/pixmaps` directory to your storage directory and change the access modes to make them writable.
- 3 Use a pixmap editor to edit the pixmap.
- 4 Customize the appropriate module type in the Network Model Schema to specify its `NODE_PIXMAP Info_attribute`.

For instructions about customizing module types, see “The Module/Subcomponent Definition File” (page 141).

- 5 Start/Restart the Network Viewer and verify that the new icon is used.

To include the new icon in the Network Viewer Legend, customize the `ND*Legend*iconTypeList` resource entry from `/opt/MagellanNMS/lib/app-defaults/C/ND` to add your new node type to its value. Note that this change should be repeated for every release as NMS may have included new standard types in the legend.

---

# Index

---

## A

- acknowledged state
  - putting components into 132
- Acknowledged states 32
- Administrators role 50
- available models
  - displaying 125

## B

- Building a Network Model 103

## C

- Cluster and pin 95
- Collecting data 69
- Component classes 21
- Components
  - acknowledged and maintenance states 33
  - adding 55
  - auto-population 54
  - criticality 26
  - organizing and positioning 56
  - propagated state 28
  - raw state 27
  - removing 56
  - updating existing components 55
- Configuration data
  - automatically collecting 104
- Configuration data files 37

## D

- DEFAULT/ALL 24
- Deleting nodes 87
- Drag and drop 101

## E

- Enable editing 63

## G

- GMDR 48

## I

- Icon bar 101

## L

- Limitations of Auto-MCDF 106
- Limitations of MCDF 106
- Limitations of SNMPMCDF 106
- Link types 22
- Links, creating and editing 82

## M

- macros 121–134
  - acknowledge 132
  - commitmodel 124–125
  - dumpmodel 126
  - maintenance 131
  - makecurrent 122–124
  - savemodel 125
  - whichcurrent 125

See also utilities  
maintenance state  
  putting components into 131  
MCDF  
  automatically collecting configuration  
    data 104  
  using 103  
  using automcdf utility 117  
MCDF utility  
  limitations 106  
MCDFUTIL utility 107  
  add a module to the exclude list 109  
  add a module to the include list 108  
  delete a module from the exclude list 109  
  delete a module from the include list 109

**N**  
Naming components 20  
Network Model 97, 99  
  active model 35  
  committed (startup) model 37  
  copying a Network Model Fileset to other  
    workstations 132  
  deploying 57  
  EdServer 44  
  Loading 65  
  maintenance 50, 54  
  MCDF  
    collecting configuration data  
      automatically 104  
  remote workstation setup 133  
  saved model 36  
  storage and life cycle 34  
network model  
  activating 122–124  
  committing 124–125  
  copying to other workstations 132  
  purging unknown components 129  
network model formats  
  converting 128  
  portable 125, 126

  quick 126  
Network Model macros 47  
Network Model schema 18  
Network Model tools and servers 38  
  MCDF 46  
  NMC 39  
  NMServer 43  
  SurNUP 40

## O

On-line editing 45  
Organizational levels, nodes and links 23

## Q

Quitting edit mode 100

## R

Remote workstation set up 133

## S

Saving network models 97, 99  
Shared Memory Utilization tools 48  
SNMPMCDF  
  using 103  
State propagation 26  
  rules 31

## U

utilities 121–134  
  popconv 128  
  poppurge 129  
  popsave 126  
  See also macros



Preside Multiservice Data Manager  
**Network Model**  
Administrator Guide

Release: R15.1

Copyright © 2004 Nortel Networks.  
All Rights Reserved.

NORTEL, NORTEL NETWORKS, the globemark design, the NORTEL NETWORKS corporate logo, PRESIDE, DPN, and PASSPORT are trademarks of Nortel Networks. SUN and SUNLINK are trademarks of Sun Microsystems Inc. SPARCSTATION is a trademark of Sparc International Inc. UNIX is a trademark licensed exclusively through X/Open Company Ltd.

Publication: 241-6001-015  
Document status: Standard  
Document version: 15.1RSUP  
Document date: August 2004  
Printed in Canada

